

HANDBOOK OF GRANULAR COMPUTING

Edited by

Witold Pedrycz

University of Alberta, Canada

and Polish Academy of Sciences, Warsaw, Poland

Andrzej Skowron

Warsaw University, Poland

Vladik Kreinovich

University of Texas, USA



A John Wiley & Sons, Ltd, Publication

HANDBOOK OF GRANULAR COMPUTING

HANDBOOK OF GRANULAR COMPUTING

Edited by

Witold Pedrycz

University of Alberta, Canada

and Polish Academy of Sciences, Warsaw, Poland

Andrzej Skowron

Warsaw University, Poland

Vladik Kreinovich

University of Texas, USA



A John Wiley & Sons, Ltd, Publication

Copyright © 2008 John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester,
West Sussex PO19 8SQ, England
Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wileyeurope.com or www.wiley.com

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except under the terms of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London W1T 4LP, UK, without the permission in writing of the Publisher. Requests to the Publisher should be addressed to the Permissions Department, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, or emailed to permreq@wiley.co.uk, or faxed to (+44) 1243 770620.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the Publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 42 McDougall Street, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 6045 Freemont Blvd, Mississauga, ONT, L5R 4J3

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Cataloging-in-Publication Data

Pedrycz, Witold, 1953-

Handbook of granular computing / Witold Pedrycz, Andrzej Skowron, Vladik Kreinovich.

p. cm.

Includes index.

ISBN 978-0-470-03554-2 (cloth)

1. Granular computing—Handbooks, manuals, etc. I. Skowron, Andrzej. II. Kreinovich, Vladik. III. Title.

QA76.9.S63P445 2008

006.3-dc22

2008002695

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN: 978-0-470-03554-2

Typeset in 9/11pt Times by Aptara Inc., New Delhi, India

Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire

Contents

Preface	ix
Foreword	xiii
Biographies	xv
Part One Fundamentals and Methodology of Granular Computing Based on Interval Analysis, Fuzzy Sets and Rough Sets	1
1 Interval Computation as an Important Part of Granular Computing: An Introduction <i>Vladik Kreinovich</i>	3
2 Stochastic Arithmetic as a Model of Granular Computing <i>René Alt and Jean Vignes</i>	33
3 Fundamentals of Interval Analysis and Linkages to Fuzzy Set Theory <i>Weldon A. Lodwick</i>	55
4 Interval Methods for Non-Linear Equation Solving Applications <i>Courtney Ryan Gwaltney, Youdong Lin, Luke David Simoni, and Mark Allen Stadtherr</i>	81
5 Fuzzy Sets as a User-Centric Processing Framework of Granular Computing <i>Witold Pedrycz</i>	97
6 Measurement and Elicitation of Membership Functions <i>Taner Bilgiç and İ. Burhan Türkşen</i>	141
7 Fuzzy Clustering as a Data-Driven Development Environment for Information Granules <i>Paulo Fazendeiro and José Valente de Oliveira</i>	153
8 Encoding and Decoding of Fuzzy Granules <i>Shounak Roychowdhury</i>	171
9 Systems of Information Granules <i>Frank Höeppner and Frank Klawonn</i>	187
10 Logical Connectives for Granular Computing <i>Erich Peter Klement, Radko Mesiar, Andrea Mesiárová-Zemáňková, and Susanne Saminger-Platz</i>	205

11	Calculi of Information Granules. Fuzzy Relational Equations <i>Siegfried Gottwald</i>	225
12	Fuzzy Numbers and Fuzzy Arithmetic <i>Luciano Stefanini, Laerte Sorini, and Maria Letizia Guerra</i>	249
13	Rough-Granular Computing <i>Andrzej Skowron and James F. Peters</i>	285
14	Wisdom Granular Computing <i>Andrzej Jankowski and Andrzej Skowron</i>	329
15	Granular Computing for Reasoning about Ordered Data: The Dominance-Based Rough Set Approach <i>Salvatore Greco, Benedetto Matarazzo, and Roman Slowiński</i>	347
16	A Unified Approach to Granulation of Knowledge and Granular Computing Based on Rough Mereology: A Survey <i>Lech Polkowski</i>	375
17	A Unified Framework of Granular Computing <i>Yiyu Yao</i>	401
18	Quotient Spaces and Granular Computing <i>Ling Zhang and Bo Zhang</i>	411
19	Rough Sets and Granular Computing: Toward Rough-Granular Computing <i>Andrzej Skowron and Jarosław Stepaniuk</i>	425
20	Construction of Rough Information Granules <i>Anna Gomolińska</i>	449
21	Spatiotemporal Reasoning in Rough Sets and Granular Computing <i>Piotr Synak</i>	471
Part Two Hybrid Methods and Models of Granular Computing		489
22	A Survey of Interval-Valued Fuzzy Sets <i>Humberto Bustince, Javier Montero, Miguel Pagola, Edurne Barrenechea, and Daniel Gómez</i>	491
23	Measurement Theory and Uncertainty in Measurements: Application of Interval Analysis and Fuzzy Sets Methods <i>Leon Reznik</i>	517
24	Fuzzy Rough Sets: From Theory into Practice <i>Chris Cornelis, Martine De Cock, and Anna Maria Radzikowska</i>	533
25	On Type 2 Fuzzy Sets as Granular Models for Words <i>Jerry M. Mendel</i>	553
26	Design of Intelligent Systems with Interval Type-2 Fuzzy Logic <i>Oscar Castillo and Patricia Melin</i>	575

27 Theoretical Aspects of Shadowed Sets <i>Gianpiero Cattaneo and Davide Ciucci</i>	603
28 Fuzzy Representations of Spatial Relations for Spatial Reasoning <i>Isabelle Bloch</i>	629
29 Rough–Neural Methodologies in Granular Computing <i>Sushmita Mitra and Mohua Banerjee</i>	657
30 Approximation and Perception in Ethology-Based Reinforcement Learning <i>James F. Peters</i>	671
31 Fuzzy Linear Programming <i>Jaroslav Ramík</i>	689
32 A Fuzzy Regression Approach to Acquisition of Linguistic Rules <i>Junzo Watada and Witold Pedrycz</i>	719
33 Fuzzy Associative Memories and Their Relationship to Mathematical Morphology <i>Peter Sussner and Marcos Eduardo Valle</i>	733
34 Fuzzy Cognitive Maps <i>E.I. Papageorgiou and C.D. Stylios</i>	755
 <hr/>	
Part Three Applications and Case Studies	775
35 Rough Sets and Granular Computing in Behavioral Pattern Identification and Planning <i>Jan G. Bazan</i>	777
36 Rough Sets and Granular Computing in Hierarchical Learning <i>Sinh Hoa Nguyen and Hung Son Nguyen</i>	801
37 Outlier and Exception Analysis in Rough Sets and Granular Computing <i>Tuan Trung Nyuyen</i>	823
38 Information Access and Retrieval <i>Gloria Bordogna, Donald H. Kraft, and Gabriella Pasi</i>	835
39 Granular Computing in Medical Informatics <i>Giovanni Bortolan</i>	847
40 Eigen Fuzzy Sets and Image Information Retrieval <i>Ferdinando Di Martino, Salvatore Sessa, and Hajime Nobuhara</i>	863
41 Rough Sets and Granular Computing in Dealing with Missing Attribute Values <i>Jerzy W. Grzymala-Busse</i>	873
42 Granular Computing in Machine Learning and Data Mining <i>Eyke Hüllermeier</i>	889

43	On Group Decision Making, Consensus Reaching, Voting, and Voting Paradoxes under Fuzzy Preferences and a Fuzzy Majority: A Survey and a Granulation Perspective	907
	<i>Janusz Kacprzyk, Sławomir Zadrożny, Mario Fedrizzi, and Hannu Nurmi</i>	
44	FuzzJADE: A Framework for Agent-Based FLCs	931
	<i>Vincenzo Loia and Mario Veniero</i>	
45	Granular Models for Time-Series Forecasting	949
	<i>Marina Hirota Magalhães, Rosangela Ballini, and Fernando Antonio Campos Gomide</i>	
46	Rough Clustering	969
	<i>Pawan Lingras, S. Ashraf, and Cory Butz</i>	
47	Rough Document Clustering and The Internet	987
	<i>Hung Son Nguyen and Tu Bao Ho</i>	
48	Rough and Granular Case-Based Reasoning	1005
	<i>Simon C.K. Shiu, Sankar K. Pal, and Yan Li</i>	
49	Granulation in Analogy-Based Classification	1037
	<i>Arkadiusz Wojna</i>	
50	Approximation Spaces in Conflict Analysis: A Rough Set Framework	1055
	<i>Sheela Ramanna</i>	
51	Intervals in Finance and Economics: Bridge between Words and Numbers, Language of Strategy	1069
	<i>Manuel Tarrazo</i>	
52	Granular Computing Methods in Bioinformatics	1093
	<i>Julio J. Valdés</i>	
	Index	1113

Preface

In *Dissertio de Arte Combinatoria* by Gottfried Wilhelm Leibniz (1666), one can find the following sentences: ‘If controversies were to arise, there would be no more need of disputation between two philosophers than between two accountants. For it would suffice to take their pencils in their hands, and say to each other: “Let us calculate”’ and in *New Essays on Human Understanding* (1705) [1], ‘Languages are the best mirror of the human mind, and that a precise analysis of the signification of words would tell us more than anything else about the operations of the understanding.’ Much later, methods based on fuzzy sets, rough sets, and other soft computing paradigms allowed us to understand that for calculi of thoughts discussed by Leibniz, it is necessary to develop tools for approximate reasoning about vague, non-crisp concepts. For example, human is expressing higher level perceptions using vague, non-Boolean concepts. Hence, for developing truly intelligent methods for approximate reasoning about such concepts in two-valued accessible for intelligent systems languages should be developed. One can gain in searching for solutions of tasks related to perceptions by using granular computing (GC). This searching in GC becomes feasible because GC-based methods use the fact that the solutions satisfy non-Boolean specifications to a satisfactory degree only. Solutions in GC can often be constructed more efficiently than in the case of methods searching for detailed, purely numeric solutions. Relevant granulation leads to efficient solutions that are represented by granules matching specifications to satisfactory degrees.

In an inductive approach to knowledge discovery, information granules provide a means of encapsulating perceptions about objects of interest [2–7].

No matter what problem is taken into consideration, we usually cast it into frameworks that facilitate observations about clusters of objects with common features and lead to problem formulation and problem solving with considerable acuity. Such frameworks lend themselves to problems of feature selection and feature extraction, pattern recognition, and knowledge discovery. Identification of relevant features of objects contained in information granules makes it possible to formulate hypotheses about the significance of the objects, construct new granules containing sample objects during interactions with the environment, use GC to measure the nearness of complex granules, and identify infomorphisms between systems of information granules.

Consider, for instance, image processing. In spite of the continuous progress in the area, a human being assumes a dominant and very much uncontested position when it comes to understanding and interpreting images.

Surely, we do not focus our attention on individual pixels but rather transform them using techniques such as non-linear diffusion and group them together in pixel windows (complex objects) relative to selected features. The parts of an image are then drawn together in information granules containing objects (clusters of pixels) with vectors of values of functions representing object features that constitute information granule descriptions. This signals a remarkable trait of humans that have the ability to construct information granules, compare them, recognize patterns, transform and learn from them, arrive at explanations about perceived patterns, formulate assertions, and construct approximations of granules of objects of interest.

As another example, consider a collection of time series. From our perspective we can describe them in a semiqualitative manner by pointing at specific regions of such signals. Specialists can effortlessly interpret ECG signals. They distinguish some segments of such signals and interpret their combinations.

Experts can seamlessly interpret temporal readings of sensors and assess the status of the monitored system. Again, in all these situations, the individual samples of the signals are not the focal point of the analysis and the ensuing signal interpretation. We always granulate all phenomena (no matter if they are originally discrete or analog in their nature). Time is another important variable that is subjected to granulation. We use milliseconds, seconds, minutes, days, months, and years. Depending on specific problem we have in mind and who the user is, the size of the information granules (time intervals) can vary quite dramatically. To the high-level management, time intervals of quarters of year or a few years can be meaningful temporal information granules on basis of which one develops any predictive model. For those in charge of everyday operation of a dispatching plant, minutes and hours could form a viable scale of time granulation. For the designer of high-speed integrated circuits and digital systems, the temporal information granules concern nanoseconds, microseconds, and, perhaps, milliseconds. Even such commonly encountered and simple examples are convincing enough to lead us to ascertain that (a) information granules are the key components of knowledge representation and processing, (b) the level of granularity of information granules (their size, to be more descriptive) becomes crucial to problem description and an overall strategy of problem solving, (c) there is no universal level of granularity of information; the size of granules is problem oriented and user dependent.

What has been said so far touched a qualitative aspect of the problem. The challenge is to develop a computing framework within which all these representation and processing endeavors can be formally realized. The common platform emerging within this context comes under the name of granular computing. In essence, it is an emerging paradigm of information processing that has its roots in Leibnitz's ideas [1] in Cantor's set theory, Zadeh's fuzzy information granulation [8], and Pawlak's discovery of elementary sets [9] (see also [10–14]).

While we have already noticed a number of important conceptual and computational constructs built in the domain of system modeling, machine learning, image processing, pattern recognition, and data compression in which various abstractions (and ensuing information granules) came into existence, GC becomes innovative and intellectually proactive in several fundamental ways:

- The information granulation paradigm leads to formal frameworks that epitomize and synthesize what has been done informally in science and engineering for centuries.
- With the emergence of unified frameworks for granular processing, we get a better grasp as to the role of interaction between various, possibly distributed, GC machines and visualize infomorphisms between them that facilitate classification and approximate reasoning.
- GC brings together the existing formalisms of set theory (interval analysis), fuzzy sets, and rough sets under the same roof by clearly visualizing some fundamental commonalities and synergies.
- Interestingly, the inception of information granules is highly motivated. We do not form information granules without reason. Information granules are an evident realization of the fundamental paradigm of scientific discovery.

This volume is one of the first, if not *the* first, comprehensive compendium on GC. There are several fundamental goals of this project. First, by capitalizing on several fundamental and well-established frameworks of fuzzy sets, interval analysis, and rough sets, we build unified foundations of computing with information granules. Second, we offer the reader a systematic and coherent exposure of the concepts, design methodologies, and detailed algorithms. In general, we decided to adhere to the top-down strategy of the exposure of the material by starting with the ideas along with some motivating notes and afterward proceeding with the detailed design that materializes in specific algorithms, applications, and case studies.

We have made the handbook self-contained to a significant extent. While an overall knowledge of GC and its subdisciplines would be helpful, the reader is provided with all necessary prerequisites. If suitable, we have augmented some parts of the material with a step-by-step explanation of more advanced concepts supported by a significant amount of illustrative numeric material.

We are strong proponents of the down-to-earth presentation of the material. While we maintain a certain required level of formalism and mathematical rigor, the ultimate goal is to present the material so

that it also emphasizes its applied side (meaning that the reader becomes fully aware of direct implications of the presented algorithms, modeling, and the like).

This handbook is aimed at a broad audience of researchers and practitioners. Owing to the nature of the material being covered and the way it is organized, we hope that it will appeal to the well-established communities including those active in computational intelligence (CI), pattern recognition, machine learning, fuzzy sets, neural networks, system modeling, and operations research. The research topic can be treated in two different ways. First, as one the emerging and attractive areas of CI and GC, thus attracting researchers engaged in some more specialized domains. Second, viewed as an *enabling* technology whose contribution goes far beyond the communities and research areas listed above, we envision a genuine interest from a vast array of research disciplines (engineering, economy, bioinformatics, etc).

We also hope that the handbook will also serve as a highly useful reference material for graduate students and senior undergraduate students in a variety of courses on CI, artificial intelligence, pattern recognition, data analysis, system modeling, signal processing, operations research, numerical methods, and knowledge-based systems.

In the organization of the material we followed a top-down approach by splitting the content into four main parts. The first one, fundamentals and methodology, covers the essential background of the leading contributing technologies of GC, such as interval analysis, fuzzy sets, and rough sets. We also offer a comprehensive coverage of the underlying concepts along with their interpretation. We also elaborate on the representative techniques of GC. A special attention is paid to the development of granular constructs, say, fuzzy sets, that serve as generic abstract constructs reflecting our perception of the world and a way of an effective problem solving. A number of highly representative algorithms (say, cognitive maps) are presented. Next, in Part II, we move on the hybrid constructs of GC where a variety of symbiotic developments of information granules, such as interval-valued fuzzy sets, type-2 fuzzy sets and shadowed sets, are considered. In the last part, we concentrate on a diversity of applications and case studies.

W. Pedrycz gratefully acknowledges the support from Natural Sciences and Engineering Research Council of Canada and Canada Research Chair program. Andrzej Skowron has been supported by the grant from the Ministry of Scientific Research and Information Technology of the Republic of Poland.

Our thanks go to the authors who enthusiastically embraced the idea and energetically agreed to share their expertise and research results in numerous domains of GC. The reviewers offered their constructive thoughts on the submissions, which were of immense help and contributed to the quality of the content of the handbook.

We are grateful for the truly professional support we have received from the staff of John Wiley, especially Kate Griffiths and Debbie Cox, who always provided us with words of encouragement and advice that helped us keep the project on schedule.

Editors-in-Chief
Edmonton – Warsaw – El Paso
May 2007

References

- [1] G.W. Leibniz. *New Essays on Human Understanding* (1705). Cambridge University Press, Cambridge, UK, 1982.
- [2] L.A. Zadeh. Fuzzy sets and information granularity. In: M.M. Gupta, R.K. Ragade, and R.R. Yager (eds), *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, 1979, 3–18.
- [3] L.A. Zadeh. Toward a generalized theory of uncertainty (GTU) – an outline. *Inf. Sci.*, 172 (2005) 1–40.
- [4] Z. Pawlak. Information systems-theoretical foundations. *Inf. Syst.* 6(3) (1981) 205–218.
- [5] J.F. Peters and A. Skowron. Zdzisław Pawlak: Life and work, transaction on rough sets V. *Springer Lect. Not. Comput. Sci.* 4100 (2006) 1–24..
- [6] Z. Pawlak and A. Skowron. Rudiments of rough sets. *Inf. Sci.* 177(1) (2007) 3–27.

- [7] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer Academic Publishers, Dordrecht, 2003.
- [8] L.A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.
- [9] Z. Pawlak. Rough sets. In: *Theoretical Aspects of Reasoning About Data-Theory and Decision Library, Series D: System Theory, Knowledge Engineering and Problem Solving*, Vol. 9. Kluwer Academic Publishers, Dordrecht, (1991).
- [10] J. Hobbs. Granulation. In: *Proceedings of the 9th IJCAI 85*, Los Angeles, California, August 18–23, 1985, pp. 432–435.
- [11] Z. Pawlak. Rough sets. *Int. J. Comput. Inf. Sci.* 11 (1982) 341–356.
- [12] Z. Pawlak. *Rough Sets. Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, Dordrecht, 1991.
- [13] W. Pedrycz (ed). *Granular Computing: An Emerging Paradigm*. Physica-Verlag, Heidelberg, 2001.
- [14] S.K. Pal, L. Polkowski, and A. Skowron (eds). *Rough-Neural Computing: Techniques for Computing with Words*. Cognitive Technologies, Springer-Verlag, Heidelberg, 2004.

Foreword

Granular Computing – co-authored by professors A. Bargiela and W. Pedrycz, and published in 2003 – was the first book on granular computing [1]. It was a superlative work in all respects. *Handbook of Granular Computing* is a worthy successor. Significantly, the co-editors of the handbook, Professors Pedrycz, Skowron, and Kreinovich are, respectively, the leading contributors to the closely interrelated fields of granular computing, rough set theory, and interval analysis – an interrelationship which is accorded considerable attention in the handbook. The articles in the handbook are divided into three groups: foundations of granular computing, interval analysis, fuzzy set theory, and rough set theory; hybrid methods and models of granular computing; and applications and case studies. One cannot but be greatly impressed by the vast panorama of applications extending from medical informatics and data mining to time-series forecasting and the internet. Throughout the handbook, the exposition is aimed at reader friendliness and deserves high marks in all respects.

What is granular computing? The preface and the chapters of this handbook provide a comprehensive answer to this question. In the following, I take the liberty of sketching my perception of granular computing – a perception in which the concept of a generalized constraint plays a pivotal role. An earlier view may be found in my 1998 paper ‘Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems’ [2].

Basically, granular computing differs from conventional modes of computation in that the objects of computation are not values of variables but information about values of variables. Furthermore, information is allowed to be imperfect; i.e., it may be imprecise, uncertain, incomplete, conflicting, or partially true. It is this facet of granular computing that endows granular computing with a capability to deal with real-world problems which are beyond the reach of bivalent-logic-based methods which are intolerant of imprecision and partial truth. In particular, through the use of generalized-constraint-based semantics, granular computing has the capability to compute with information described in natural language.

Granular computing is based on fuzzy logic. There are many misconceptions about fuzzy logic. To begin with, fuzzy logic is not fuzzy. Basically, fuzzy logic is a precise logic of imprecision. Fuzzy logic is inspired by two remarkable human capabilities. First, the capability to reason and make decisions in an environment of imprecision, uncertainty, incompleteness of information, and partiality of truth. And second, the capability to perform a wide variety of physical and mental tasks based on perceptions, without any measurements and any computations. The basic concepts of graduation and granulation form the core of fuzzy logic, and are the principal distinguishing features of fuzzy logic. More specifically, in fuzzy logic everything is or is allowed to be graduated, i.e., be a matter of degree or, equivalently, fuzzy. Furthermore, in fuzzy logic everything is or is allowed to be granulated, with a granule being a clump of attribute values drawn together by indistinguishability, similarity, proximity, or functionality. The concept of a generalized constraint serves to treat a granule as an object of computation. Graduated granulation, or equivalently fuzzy granulation, is a unique feature of fuzzy logic. Graduated granulation is inspired by the way in which humans deal with complexity and imprecision.

The concepts of graduation, granulation, and graduated granulation play key roles in granular computing. Graduated granulation underlies the concept of a linguistic variable, i.e., a variable whose values are words rather than numbers. In retrospect, this concept, in combination with the associated concept of a fuzzy if–then rule, may be viewed as a first step toward granular computing.

Today, the concept of a linguistic variable is used in almost all applications of fuzzy logic. When I introduced this concept in my 1973 paper ‘Outline of a new approach to the analysis of complex systems and decision processes’ [3], I was greeted with scorn and derision rather than with accolades. The derisive comments reflected a deep-seated tradition in science – the tradition of according much more respect to numbers than to words. Thus, in science, progress is equated to progression from words to numbers. In fuzzy logic, in moving from numerical to linguistic variables, we are moving in a countertraditional direction. What the critics did not understand is that in moving in the countertraditional direction, we are sacrificing precision to achieve important advantages down the line. This is what is called ‘the fuzzy logic gambit.’ The fuzzy logic gambit is one of the principal rationales for the use of granular computing.

In sum, to say that the *Handbook of Granular Computing* is an important contribution to the literature is an understatement. It is a work whose importance cannot be exaggerated. The coeditors, the authors, and the publisher, John Wiley, deserve our thanks, congratulations, and loud applause.

Lotfi A. Zadeh
Berkeley, California

References

- [1] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer Academic Publishers, Dordrecht, 2003.
- [2] L.A. Zadeh. Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. *Soft Comput.* 2 (1998) 23–25.
- [3] L.A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Syst. Man Cybern.* SMC-3 (1973) 28–44.

Biographies

Witold Pedrycz (M'88-SM'90-F'99) received the MSc, PhD, and DSci from the Silesian University of Technology, Gliwice, Poland. He is a professor and Canada Research Chair in computational intelligence in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He is also with the Polish Academy of Sciences, Systems Research Institute, Warsaw, Poland.

His research interests encompass computational intelligence, fuzzy modeling, knowledge discovery and data mining, fuzzy control including fuzzy controllers, pattern recognition, knowledge-based neural networks, granular and relational computing, and software engineering. He has published numerous papers in these areas. He is also an author of 11 research monographs. Witold Pedrycz has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. He serves as an editor-in-chief of *IEEE Transactions on Systems Man and Cybernetics – Part A* and associate editor of *IEEE Transactions on Fuzzy Systems*. He is also an editor-in-chief of information sciences. Dr. Pedrycz is a recipient of the prestigious Norbert Wiener Award from the IEEE Society of Systems, Man, and Cybernetics as well as K.S. Fu Award from the North American Fuzzy Information Society.

Andrzej Skowron received the PhD and DSci from the University of Warsaw in Poland. In 1991 he received the Scientific Title of Professor. He is a Full Professor in the Faculty of Mathematics, Computer Science and Mechanics at Warsaw University. Andrzej Skowron is the author of numerous scientific publications and editor of many books and special issues of scientific journals. His areas of expertise include reasoning with incomplete information, approximate reasoning, soft computing methods and applications, rough sets, rough mereology, granular computing, synthesis and analysis of complex objects, intelligent agents, knowledge discovery systems, and advanced data mining techniques, decision support systems, adaptive and autonomous systems. He was the supervisor of more than 20 PhD theses. He was also involved in several national and international research and commercial projects relating to data mining (fraud detection and web mining), control of unmanned vehicles, medical decision support systems, and approximate reasoning in distributed environments among many others.

Since 1995 he is the editor-in-chief of *Fundamenta Informaticae* journal and a member of editorial boards of several other journals including *Knowledge Discovery and Data*. He is the coeditor-in-chief of the journal *LNCS Transactions on Rough Sets* published by Springer. Andrzej Skowron was the president of the International Rough Set Society from 1996 to 2000. He served or is currently serving on the program committees of almost 100 international conferences and workshops as program committee member, program chair, or cochair. He has delivered numerous invited talks at international conferences, including a plenary talk at the *16th IFIP World Computer Congress* (Beijing, 2000). Throughout his career, Andrzej Skowron has won many awards for his achievements, including awards from the Ministry of Science, the Rector of Warsaw University, the Ministry of Education, Mazur's Award of the Polish Mathematical Society, and Janiszewski's Award of the Polish Mathematical Society. In 2003 he received the title of Honorary Professor from Chongqing University of Post and Telecommunication (China). In 2005 he received the ACM Recognition of Service Award for contributions to ACM and the award from International Rough Sets Society for the outstanding research results.

Dr. Vladik Kreinovich received his MSc in mathematics and computer science from St Petersburg University, Russia, in 1974 and PhD from the Institute of Mathematics, Soviet Academy of Sciences,

Novosibirsk, in 1979. In 1975–1980, he worked with the Soviet Academy of Sciences, in particular, in 1978–1980, with the Special Astrophysical Observatory (representation and processing of uncertainty in radioastronomy). In 1982–1989, he worked on error estimation and intelligent information processing for the National Institute for Electrical Measuring Instruments, Russia. In 1989, he was a visiting scholar at Stanford University. Since 1990, he is with the Department of Computer Science, University of Texas at El Paso. Also, he served as an invited professor in Paris (University of Paris VI), Hong Kong, St Petersburg, Russia, and Brazil.

His main interests include representation and processing of uncertainty, especially interval computations and intelligent control. He has published 3 books, 6 edited books, and more than 700 papers. He is member of the editorial board of the international journal *Reliable Computing* (formerly, *Interval Computations*) and several other journals. He is also the comaintainer of the international website on interval computations, <http://www.cs.utep.edu/interval-comp>.

He is foreign member of the Russian Academy of Metrological Sciences, recipient of the 2003 El Paso Energy Foundation Faculty Achievement Award for Research awarded by the University of Texas at El Paso, and a corecipient of the 2005 Star Award from the University of Texas System.

René Alt is a professor of computer sciences at the Pierre et Marie Curie University in Paris (UPMC). He received his master diploma in mathematics from UPMC in 1968, the Doctorate in Computer Sciences (PhD) of UPMC in 1971, and was Docteur es Sciences from UPMC in 1981. He has been professor of computer sciences at the University of Caen (France) from 1985 to 1991. He was head of the faculty of computer sciences of UPMC from 1997 to 2001 and vice president of the administrative council of UPMC from 2002 to 2006. René Alt's fields of interest are the numerical solution of differential equations, computer arithmetic, round-off error propagation, validation of numerical software, parallel computing, and image processing.

S. Asharaf received the BTech from the Cochin University of Science and Technology, Kerala, and the Master of Engineering from the Indian Institute of Science, where he is working toward a PhD. His research interests include data clustering, soft computing, and support vector machines. He is one of the recipients of IBM best PhD student award in 2006.

Rosangela Ballini received her BSc degree in applied mathematics from the Federal University of São Carlos (UFSCar), SP, Brazil, in 1996. In 1998, she received the MSc degree in mathematics and computer science from the University of São Paulo (USP), SP, Brazil, and the PhD degree in electrical engineering from the State University of Campinas (Unicamp), SP, Brazil, in 2000. Currently, she is professor of the Department of Economic Theory, Institute of Economics (IE), Unicamp. Her research interests include time series forecasting, neural networks, fuzzy systems, and non-linear optimization.

Mohua Banerjee received her BSc (Hons) degree in mathematics, and the MSc, MPhil, and PhD degrees in pure mathematics from the University of Calcutta in 1985, 1987, 1989, and 1995, respectively. During 1995–1997, she was a research associate at the Machine Intelligence Unit, Indian Statistical Institute, Calcutta. In 1997, she joined the Department of Mathematics and Statistics, Indian Institute of Technology, Kanpur, as lecturer, and is currently Assistant Professor in the same department. She was an associate of The Institute of Mathematical Sciences, Chennai, India, during 2003–2005. Her main research interests lie in modal logics and rough sets. She has made several research visits to institutes in India and abroad. She is a member of the Working Group for the Center for Research in Discrete Mathematics and its Applications (CARDMATH), Department of Science and Technology (DST), Government of India. She serves in the reviewer panel of many international journals. Dr. Banerjee was awarded the Indian National Science Academy Medal for Young Scientists in 1995.

Edurne Barrenechea is an assistant lecturer at the Department of Automatics and Computation, Public University of Navarra, Spain. Having received an MSc in computer science at the País Vasco University in 1990. She worked as analyst programmer in Bombas Itur from 1990 to 2001 and then she joined the Public University of Navarra as associate lecturer. She obtained the PhD in computer science in 2005.

Her research interests are fuzzy techniques for image processing, fuzzy sets theory, interval type-2 fuzzy sets theory, neural networks, and industrial applications of soft computing techniques. She is member of the European Society for Fuzzy Logic and Technology (EUSFLAT).

Jan G. Bazan is an Assistant Professor in the Institute of Mathematics at the University of Rzeszow in Poland. He received his PhD degree in 1999 from the University of Warsaw in Poland. His recent research interests focus on rough set theory, granular computing, knowledge discovery, data mining techniques, reasoning with incomplete information, approximate reasoning, decision support systems, and adaptive systems. He is the author or coauthor of more than 40 scientific publications and he was involved in several national and international research projects relating to fraud detection, web mining, risk pattern detection, and automated planning of the treatment among other topics.

Taner Bilgiç received his BSc and MSc in industrial engineering from the Middle East Technical University, Ankara, Turkey, in 1987 and 1990, respectively. He received a PhD in industrial engineering from the University of Toronto in 1995. The title of his dissertation is ‘Measurement-Theoretic Frameworks for Fuzzy Set Theory with Applications to Preference Modelling.’ He spent 2 years at the Enterprise Integration Laboratory in Toronto as a research associate. Since 1997, he has been a faculty member at the Department of Industrial Engineering at Bogazici University in Istanbul, Turkey.

Isabelle Bloch is a professor at ENST (Signal and Image Processing Department), CNRS UMR 5141 LTCI. Her research interests include three-dimensional (3D) image and object processing, 3D and fuzzy mathematical morphology, decision theory, information fusion, fuzzy set theory, belief function theory, structural pattern recognition, spatial reasoning, and medical imaging.

Gloria Bordogna received her Laurea degree in Physics at the Università degli Studi di Milano, Italy, in 1984. In 1986 she joined the Italian National Research Council, where she presently holds the position of a senior researcher at the Institute for the Dynamics of Environmental Processes. She is also a contract professor at the faculty of Engineering of Bergamo University, where she teaches information retrieval and geographic information systems. Her research activity concerns soft computing techniques for managing imprecision and uncertainty affecting both textual and spatial information. She is coeditor of a special issue of *JASIS* and three volumes published by Springer-Verlag on uncertainty and impression management in databases. She has published over 100 papers in international journals, in the proceedings of international conferences, and in books. She participated at the program committee of international conferences such as *FUZZIEEE*, *ECIR*, *ACM SIGIR*, *FQAS*, *EUROFUSE*, *IJCAI2007*, *ICDE 2007*, and *ACM SAC ‘Information Access and Retrieval’ track* and served as a reviewer of journals such as *JASIST*, *IEEE Transactions on Fuzzy Systems*, *Fuzzy Sets and Systems*, and *Information Processing and Management*.

Giovanni Bortolan received the doctoral degree from the University of Padova, Padova, Italy in 1978. He is senior researcher at the Institute of Biomedical Engineering, Italian National Research Council (ISIB-CNR), Padova, Italy. He has published numerous papers in the areas of medical informatics and applied fuzzy sets. He is actively pursuing research in medical informatics in computerized electrocardiography, neural networks, fuzzy sets, data mining, and pattern recognition.

Humberto Bustince is an Associate Professor at the Department of Automatics and Computation, Public University of Navarra, Spain. He holds a PhD degree in mathematics from Public University of Navarra from 1994. His research interests are fuzzy logic theory, extensions of fuzzy sets (type-2 fuzzy sets and Atanassov’s intuitionistic fuzzy sets), fuzzy measures, aggregation operators, and fuzzy techniques for image processing. He is the author of more than 30 peer-reviewed research papers and is member of IEEE and European Society for Fuzzy Logic and Technology (EUSFLAT).

Cory J. Butz received the BSc, MSc, and PhD degrees in computer science from the University of Regina, Saskatchewan, Canada, in 1994, 1996, and 2000, respectively. His research interests include uncertainty reasoning, database systems, information retrieval, and data mining.

Oscar Castillo was awarded Doctor of Science (DSc) from the Polish Academy of Sciences. He is a professor of computer science in the Graduate Division, Tijuana Institute of Technology, Tijuana, Mexico. In addition, he is serving as research director of computer science and head of the research group on fuzzy logic and genetic algorithms. Currently, he is president of Hispanic American Fuzzy Systems Association (HAFSA) and vice president of International Fuzzy Systems Association (IFSA) in charge of publicity. Professor Castillo is also vice chair of the Mexican Chapter of the Computational Intelligence Society (IEEE). Professor Castillo is also general chair of the IFSA 2007 World Congress to be held in Cancun, Mexico. He also belongs to the Technical Committee on Fuzzy Systems of IEEE and to the Task Force on ‘Extensions to Type-1 Fuzzy Systems.’ His research interests are in type-2 fuzzy logic, intuitionistic fuzzy logic, fuzzy control, neuro-fuzzy, and genetic–fuzzy hybrid approaches. He has published over 60 journal papers, 5 authored books, 10 edited books, and 150 papers in conference proceedings.

Gianpiero Cattaneo is a Full Professor in ‘dynamical system theory’ at the Università di Milano, Bicocca. Previously, he was an Associate Professor in ‘mathematical methods of physics’ (from 1974 to 1984) and researcher of ‘theoretical physics’ (from 1968 to 1974). From 1994 to 1997, he was a regular visiting professor at the London School of Economics (Department of Logic and Scientific Methods), where, since 1998, he had a position of research associate at ‘The Centre for the Philosophy of Natural and Social Science.’ From 1997 to 1999, he was Maitre de Conferences at the Nancy-Metz Academy and Maitre de Conferences at ‘la Ecole Normale Supérieure’ in Lyon: Laboratoire de l’Informatique du Parallélisme. He is member of the editorial board of the *Transactions on Rough Sets*, LNCS (Springer-Verlag), the Scientific Committee of the ‘International Quantum Structures Association (IQSA)’; the International Advisory Board of the ‘European School of Advanced Studies in Methods for Management of Complex Systems’ (Pavia); International Federation of Information Processing (IFIP): Working group on cellular automata. Moreover, he is scientific coordinator of a biannual 2006–2007 ‘Program of International Collaboration’ between France and Italy, involving the universities of Nice, Marseille, Ecole Normale Supérieure de Lyon, Marne-la-Vallée, Milano-Bicocca, and Bologna. He was a member of numerous program committees of international conferences. His research activities, with results published on international journals in more than 140 papers, are centered on topological chaos, cellular automata and related languages, algebraic approach to fuzzy logic and rough sets, axiomatic foundations of quantum mechanics, and realization of reversible gates by quantum computing techniques.

Davide Ciucci received a PhD in 2004 in computer science from the University of Milan. Since 2005, he has held a permanent position as a researcher at the University of Milano-Bicocca, where he delivered a course on fuzzy logic and rough sets. His research interests are about a theoretical algebraic approach to imprecision, with particular attention to many-valued logics, rough sets, and their relationship. Recently, he got involved in the semantic web area, with a special interest in fuzzy ontology and fuzzy description logics. He has been a member committee of several conferences about rough and fuzzy sets, co-organizer of a special session at the Joint Rough Set Symposium JRS07. His webpages, with a list of publications, can be found at www.fislab.disco.unimib.it.

Chris Cornelis is a postdoctoral researcher at the Department of Applied Mathematics and Computer Science at Ghent University (Belgium) funded by the Research Foundation – Flanders. His research interests include various models of imperfection (fuzzy rough sets, bilattices and interval-valued fuzzy sets); he is currently focusing on their application to personalized information access and web intelligence.

Martine De Cock is a professor at the Department of Applied Mathematics and Computer Science at Ghent University (Belgium). Her current research efforts are directed toward the development and the use of computational intelligent methods for next-generation web applications.

E.I. Papageorgiou was born in Larisa in 1975, Greece. She obtained the physics degree in 1997, MSc in medical physics in 2000, and PhD in computer science in July 2004 from the University of Patras. From 2004 to 2006, she was a postdoctoral researcher at the Department of Electrical and Computer

Engineering, University of Patras (Greece), on developing new models and methodologies based on soft computing for medical decision support systems. From 2000 to 2006, she was involved in several research projects related to the development of new algorithms and methods for complex diagnostic and medical decision support systems. Her main activities were the development of innovative learning algorithms for fuzzy cognitive maps and intelligent expert systems for medical diagnosis and decision-making tasks. From 2004 to 2005, she was appointed as lecturer at the Department of Electrical and Computer Engineering at the University of Patras. Currently, she is Assistant Professor at the Department of Informatics and Computer Technology, Technological Educational Institute of Lamia, and adjunct Assistant Professor at the University of Central Greece. She has coauthored more than 40 journals and conference papers, book chapters, and technical reports, and has more than 50 citations to her works. Her interests include expert systems, intelligent algorithms and computational intelligence techniques, intelligent decision support systems, and artificial intelligence techniques for medical applications. Dr. E.I. Papageorgiou was a recipient of a scholarship of Greek State Scholarship Foundation 'I.K.Y.' during her PhD studies (2000–2004), and from 2006 to May 2007, she was also a recipient of the postdoctoral research fellowship from the Greek State Scholarship Foundation 'I.K.Y.'

Paulo Fazendeiro received the BS degree in mathematics and informatics in 1995 (with honors) and the equivalent of MS degree in computer science in 2001, all from the University of Beira Interior, Portugal. He is preparing his dissertation on the relationships between accuracy and interpretability of fuzzy systems as a partial fulfillment of the requirements for the informatics engineering PhD degree. He joined the University of Beira Interior in 1995, where he is currently a lecturer in the Informatics Department. His research interests include application of fuzzy set theory and fuzzy systems, data mining, evolutionary algorithms, multiobjective optimization, and clustering techniques with applications to image processing. Dr. Fazendeiro is a member of the Portuguese Telecommunications Institute and the Informatics Laboratory of the University of Algarve.

Mario Fedrizzi received the MSc degree in mathematics in 1973 from the University of Padua, Italy. Since 1976, he has been an Assistant Professor; since 1981, an Associate Professor; and since 1986, a Full Professor with Trento University, Italy. He served as a chairman of the Institute of Informatics from 1985 to 1991 and as a dean of the Faculty of Economics and Business Administration from 1989 to 1995. His research focused on utility and risk theory, stochastic dominance, group decision making, fuzzy decision analysis, fuzzy regression analysis, and consensus modeling in uncertain environments, decision support systems. He has authored or coauthored books and more than 150 papers, which appeared in international proceedings and journals, e.g., *European Journal of Operational Research*, *Fuzzy Sets and Systems*, *IEEE Transactions on Systems, Man and Cybernetics*, *Mathematical Social Sciences*, *Quality and Quantity*, and *International Journal of Intelligent Systems*. He was also involved in consulting activities in the areas of information systems and DSS design and implementation, office automation, quality control, project management, expert systems, and neural nets in financial planning. From 1995 to 2006, he was appointed as chairman of a bank and of a real-estate company, and as a member of the board of directors of Cedacri, the largest Italian banking information systems outsourcing company, and of Unicredit Banca.

Fernando Antonio Campos Gomide received the BSc degree in electrical engineering from the Polytechnic Institute of the Pontifical Catholic University of Minas Gerais (IPUC/PUC-MG) Belo Horizonte, Brazil; the MSc degree in electrical engineering from the State University of Campinas (Unicamp), Campinas, Brazil; and the PhD degree in systems engineering from Case Western Reserve University (CWRU), Cleveland, Ohio, USA. He is professor of the Department of Computer Engineering and Automation (DCA), Faculty of Electrical and Computer Engineering (FEEC) of Unicamp, since 1983. His interest areas include fuzzy systems, neural and evolutionary computation, modeling, control and optimization, logistics, decision making, and applications. Currently, he serves on editorial boards of *Fuzzy Sets and Systems*, *Intelligent Automation and Soft Computing*, *IEEE Transactions on SMC-B*, *Fuzzy Optimization and Decision Making*, and *Mathware and Soft Computing*. He is a regional editor of the *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, and *Journal of Advanced Computational Intelligence*.

Anna Gomolińska received a PhD in mathematics from Warsaw University in 1993. Her doctoral thesis, written under the supervision of Cecilia M. Rauszer, was entitled ‘Logical Methods of Knowledge Representation Under Incomplete Information.’ She works as a teacher in the Department of Mathematics of Białystok University. She was a visiting scholar at Uppsala University in 1994 as well as a research fellow at the Swedish Collegium for Advanced Studies (SCAS) in Uppsala in 1995 and at the CNR Institute of Computer Science (IASI-CNR) in Rome, 2002. Anna Gomolińska has been the author or coauthor of around 40 research articles. Her scientific interests are rough sets, multiagent systems, game theory, and logical aspects of computer science and artificial intelligence. Since 2001 she has been a member of the research group led by Professor Andrzej Skowron from Warsaw University.

Siegfried Gottwald, born in 1943, teaches mathematics and logic at Leipzig University since 1972. He got his PhD in mathematics there in 1969 and his habilitation degree in 1977. He became University Docent in Logic there in 1979 and Associate Professor in 1987. Since 1992, he is Full Professor for ‘non-classical and mathematical logic’ at Leipzig University and head of the ‘Institute for Logic and Philosophy of Science’ there. He has been active in research on fuzzy logic, fuzzy sets, and fuzzy methodologies for over three decades now. His main topics include the fundamentals of fuzzy set theory, many-valued logic and their relationship to fuzzy sets and vague notions, fuzzy relation equations and their relationship to fuzzy control, as well as fuzzy logic and approximate reasoning. He is also interested in the history and philosophy of logic and mathematics. He has published several books on many-valued logic, fuzzy sets, and their applications, was coauthor of a textbook on calculus and of a reader in the history of logic, and coedited and coauthored a biographical dictionary of mathematicians. He was a visiting scholar at the Department of Computer Science TH Darmstadt, at the Departments of Philosophy at the University of California in Irvine, and at the Indiana University in Bloomington, IN. Actually, he is area editor for ‘non-classical logics and fuzzy set theory’ of the international journal *Fuzzy Sets and Systems*, member of the editorial boards of *Multiple-Valued Logic and Soft Computing* and of *Information Sciences*, as well as of the consulting board of former editors of *Studia Logica*. In 1992 he was honored with the research award ‘Technische Kommunikation’ of the (German) Alcatel-SEL-Foundation.

Salvatore Greco has been a Full Professor at the Faculty of Economics of Catania University since 2001. His main research interests are in the field of multicriteria decision aid (MCDA), in the application of the rough set approach to decision analysis, in the axiomatic foundation of multicriteria methodology, and in the fuzzy integral approach to MCDA. In these fields he cooperates with many researchers of different countries. He received the Best Theoretical Paper Award, by the Decision Sciences Institute (Athens, 1999). Together with Benedetto Matarazzo, he organized the Seventh International Summer School on MCDA (Catania, 2000). He is the author of many articles published in important international journals and specialized books. He has been a Visiting Professor at Poznan Technical University and at the University of Paris Dauphine. He has been an invited speaker at important international conferences. He is a referee of the most relevant journals in the field of decision analysis.

Maria Letizia Guerra is an Associate Professor at the University of Bologna (Italy), where she currently teaches mathematics for economics and finance. She received a PhD in computational methods for finance from Bergamo University in 1997; her current research activity examines stochastic and fuzzy models for derivatives pricing and risk management.

Daniel Gómez is a Full Professor in the Department of Statistics and Operational Research III at the Faculty of Statistics, Complutense University of Madrid, Spain. He has held a PhD in mathematics from Complutense University since 2003. He is the author of more than 20 research papers in refereed journals and more than 10 papers as book chapters. His research interests are in multicriteria decision making, preference representation, aggregation, classification problems, fuzzy sets, and graph theory.

Jerzy W. Grzymala-Busse is a professor of electrical engineering and computer science at the University of Kansas since August of 1993. His research interests include data mining, machine learning, knowledge discovery, expert systems, reasoning under uncertainty, and rough set theory. He has

published three books and over 200 articles. He is a member of editorial boards of the *Foundations of Computing and Decision Science*, *International Journal of Knowledge-Based Intelligent Engineering Systems*, *Fundamenta Informaticae*, *International Journal of Hybrid Intelligent System*, and *Transactions on Rough Sets*. He is a vice president of the International Rough Set Society and member of the Association for Computing Machinery, American Association for Artificial Intelligence, and Upsilon Pi Epsilon.

Courtney Ryan Gwaltney has a BSc degree from the University of Kansas and a PhD degree from the University of Notre Dame, both in chemical engineering. He received the 2006 Eli J. and Helen Shaheen Graduate School Award for excellence in research and teaching at Notre Dame. He is currently employed by BP.

Tu Bao Ho is a professor at the School of Knowledge Science, Japan Advanced Institute of Science and Technology, Japan. He received his MSc and PhD from Marie and Pierre Curie University in 1984 and 1987, respectively, and habilitation from Paris Dauphine University in 1998. His research interests include knowledge-based systems, machine learning, data mining, medical informatics, and bioinformatics. Tu Bao Ho is a member of editorial board of the following international journals: *Studia Informatica*, *Knowledge and Systems Sciences*, *Knowledge and Learning*, and *Business Intelligence and Data Mining*. He is also an associate editor of *Journal of Intelligent Information and Database Systems*, a review board member of *International Journal of Applied Intelligence*, and a member of the Steering Committee of PAKDD (*Pacific-Asia Conferences on Knowledge Discovery and Data Mining*).

Frank Höppner received his MSc and PhD in computer science from the University of Braunschweig in 1996 and 2003, respectively. He is now professor for information systems at the University of Applied Sciences Braunschweig/Wolfenbuttel in Wolfsburg (Germany). His main research interest is knowledge discovery in databases, especially clustering and the analysis of sequential data.

Eyke Hüllermeier, born in 1969, holds MS degrees in mathematics and business computing, both from the University of Paderborn (Germany). From the Computer Science Department of the same university he obtained his PhD in 1997 and a habilitation degree in 2002. He spent 2 years from 1998 to 2000 as a visiting scientist at the Institut de Recherche en Informatique de Toulouse (France) and held appointments at the Universities of Dortmund, Marburg, and Magdeburg afterwards. Recently, he joined the Department of Mathematics and Computer Science at Marburg University (Germany), where he holds an appointment as a Full Professor and heads the Knowledge Engineering and Bioinformatics Lab. Professor Hüllermeier's research interests include methodical foundations of machine learning and data mining, fuzzy set theory, and applications in bioinformatics. He has published numerous research papers on these topics in respected journals and major international conferences. Professor Hüllermeier is a member of the IEEE, the IEEE Computational Intelligence Society, and a board member of the European Society for Fuzzy Logic and Technology (EUSFLAT). Moreover, he is on the editorial board of the journals *Fuzzy Sets and Systems*, *Soft Computing*, and *Advances in Fuzzy Systems*.

Andrzej Jankowski received his PhD from Warsaw University, where he worked for more than 15 years, involved in pioneering research on the algebraic approach to knowledge representation and reasoning structures based on topos theory and evolution of hierarchies of metalogics. For 3 years, he worked as a visiting professor in the Department of Computer Science at the University of North Carolina, Charlotte, USA. He has unique experience in managing complex IT projects in Central Europe, for example; he was inventor and the project manager of such complex IT projects for government like POLTAX (one of the biggest tax modernization IT project in Central Europe) and e-POLTAX (e-forms for tax system in Poland). He accumulated the extensive experience in the government, corporate, industry, and finance sectors. He also supervised several AI-based commercial projects such as intelligent fraud detection and an intelligent search engine. Andrzej Jankowski is one of the founders of the Polish–Japanese Institute of Information Technology and for 5 years he served as its deputy rector for research and teaching.

Janusz Kacprzyk MSc in computer science and automatic control, PhD in systems analysis, DSc in computer science, professor since 1997, and member of the Polish Academy of Sciences since 2002. Since 1970 with the Systems Research Institute, Polish Academy of Sciences, currently as professor and deputy director for research. Visiting professor at the University of North Carolina, University of Tennessee, Iona College, University of Trento, and Nottingham Trent University. Research interests include soft computing, fuzzy logic and computing with words, in decisions and optimization, control, database querying, and information retrieval. 1991–1995: IFSA vice president, 1995–1999: in IFSA Council, 2001–2005: IFSA treasurer, 2005: IFSA president-elect, IFSA fellow, IEEE Fellow. Recipient of numerous awards, notably 2005 IEEE CIS Pioneer Award for seminal works on multistage fuzzy control, notably fuzzy dynamic programming, and the sixth Kaufmann Prize and Gold Medal for seminal works on the application of fuzzy logic and economy and managements. Editor of three Springer's book series: *Studies in Fuzziness and Soft Computing*, *Advances in Soft Computing*, and *Studies in Computational Intelligence*. On editorial boards of 20 journals. Author of 5 books, (co)editor of 30 volumes, and (co)author of 300 papers. Member of IPC at 150 conferences.

Frank Klawonn received his MSc and PhD in mathematics and computer science from the University of Braunschweig in 1988 and 1992, respectively. He has been a visiting professor at Johannes Kepler University in Linz (Austria) in 1996 and at Rhodes University in Grahamstown (South Africa) in 1997. He is now the head of the Lab for Data Analysis and Pattern Recognition at the University of Applied Sciences in Wolfenbüttel (Germany). His main research interests focus on techniques for intelligent data analysis especially clustering and classification. He is an area editor of the *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* and a member of the editorial board of the *International Journal of Information Technology and Intelligent Computing*, *Fuzzy Sets and Systems*, as well as *Mathware & Soft Computing*.

Erich Peter Klement received his PhD in Mathematics in 1971 from the University of Innsbruck, Austria. He is a professor of mathematics and chairman of the Department of Knowledge-Based Mathematical Systems at the Johannes Kepler University, Linz, Austria. He held long-term visiting research positions at the University of California, Berkeley (USA), the Université Aix-Marseille II (France), and the Tokyo Institute of Technology (Japan), and he worked as a visiting professor at the Universities of Klagenfurt (Austria), Cincinnati (Ohio, USA), and Trento (Italy). His major research interest is in the foundations of fuzzy logic and fuzzy control as well as in the application in probability and statistics, game theory, and image and signal processing. He is author/coauthor of three monographs, coeditor of six edited volumes, and author/coauthor of 90 papers in international journals and edited volumes. He served on the editorial board of nine international scientific journals, and he is a member of IEEE, the European Association for Fuzzy Logic and Technology, and the American and the Austrian Mathematical Society.

Donald H. Kraft Professor, Department of Computer Science, Louisiana State University, Baton Rouge, LA. He is an editor of *Journal of the American Society for Information Science and Technology* (JASIST), and editorial board member of *Information Retrieval*, *International Journal of Computational Intelligence Research* (IJCIR), and *Journal of Digital Information Management* (JDIM). In other professional activities he served as a summer faculty of U.S. Air Force Office of Scientific Research (AFOSR), a research associate of Wright-Patterson Air Force Base, Ohio, USA. He worked on a project, contracted through Research and Development Laboratories (RDL), to do an exploratory study of weighted fuzzy keyword retrieval and automatic generation of hypertext links for CASHE:PVS, a hypermedia system of human engineering documents and standards for use in design.

Yan Li received the BSc and MSc degrees in mathematics in 1998 and 2001, respectively, from the College of Computer and Mathematics, Hebei University, PR China. She received her PhD degree in computer science from the Department of Computing, the Hong Kong Polytechnic University. She is currently an Assistant Professor of the School of Computer and Mathematics, Hebei University, PR China. Her interests include fuzzy mathematics, case-based reasoning, rough set theory, and information retrieval. She is a member of the IEEE.

Youdong Lin has BS and MS degrees in chemical engineering from Tsinghua University. He received the PhD degree in chemical engineering from the University of Notre Dame, where he is currently a research associate. He received the 2004 SGI Award for Computational Sciences and Visualization for his outstanding research at Notre Dame.

Pawan Lingras' undergraduate education from Indian Institute of Technology, Bombay, India, was followed by graduate studies at the University of Regina, Canada. His areas of interests include artificial intelligence, information retrieval, data mining, web intelligence, and intelligent transportation systems.

Weldon A. Lodwick was born and raised in São Paulo, Brazil, to U.S. parents, where he lived through high school. He came to live in the USA and went to Muskingum College in New Concord, Ohio, USA, where he graduated from this college in 1967, with major in mathematics (honors), a minor in physics, and an emphasis in philosophy. He obtained his masters degree from the University of Cincinnati in 1969 and a PhD in mathematics from Oregon State University. He left Oregon State University in 1977 to begin work at Michigan State University as a systems analyst for an international project for food production potential working in the Dominican Republic, Costa Rica, Nicaragua, Honduras, and Jamaica. In addition, he developed software for food production analysis for Syria. His job consisted of developing software, geographical information systems, statistical models, linear programming models, analysis, and training for transfer to the various countries in which the project was working. While in Costa Rica, Dr. Lodwick worked directly with the Organization of American States (IICA), with some of their projects in Nicaragua and Honduras that had similar emphasis as that of Michigan State University. In 1982, he was hired by the Department of Mathematics of the University of Colorado at Denver where currently he is a Full Professor of mathematics.

Vincenzo Loia received the PhD in computer science from the University of Paris VI, France, in 1989 and the bachelor degree in computer science from the University of Salerno in 1984. From 1989 he is faculty member at the University of Salerno where he teaches operating-system-based systems and multiagent-based systems. His current position is as professor and head of the Department of Mathematics and Computer Science. He was principal investigator in a number of industrial R&D projects and in academic research projects. He is author of over 100 original research papers in international journals, book chapters, and international conference proceedings. He edited three research books around agent technology, Internet, and soft computing methodologies. He is cofounder of the Soft Computing Laboratory and founder of the Multiagent Systems Laboratory, both at the Department of Mathematics and Computer Science. He is coeditor-in-chief of *Soft Computing*, an international Springer-Verlag journal. His current research interests focus on merging soft computing and agent technology to design technologically complex environments, with particular interest in web intelligence applications. Dr. Loia is chair of the IEEE Emergent Technologies Technical Committee in the IEEE Computational Intelligence. He is also member of the International Technical Committee on Media Computing, IEEE Systems, Man and Cybernetics Society.

Marina Hirota Magalhães received her BSc degree in applied mathematics and computation from the State University of Campinas (Unicamp), SP, Brazil, in 2001. In 2004, she received the MSc degree in electrical engineering from the State University of Campinas (Unicamp), SP, Brazil. Currently, she is a PhD candidate in the National Institute of Research Space (INPE), SP, Brazil. Her research interests include time series forecasting, fuzzy systems, and neural networks.

Ferdinando Di Martino is professor of computer science at the Faculty of Architecture of Naples University Federico II. Since 1990 he has participated to national and international research projects in artificial intelligence and soft computing. He has published numerous papers on well-known international journals and his main interests concern applications of fuzzy logic to image processing, approximate reasoning, geographic information systems, and fuzzy systems.

Benedetto Matarazzo is a Full Professor at the Faculty of Economics of Catania University. He has been member of the committee of scientific societies of operational researches. He is organizer and member of the program committee and he has been invited speaker in many scientific conferences. He is member of the editorial boards of the *European Journal of Operational Research*, *Journal of Multi-Criteria Decision Analysis*, and *Foundations of Computing and Decision Sciences*. He has been chairman of the Program Committee of EURO XVI (Brussels, 1998). His research is in the fields of MCDA and rough sets. He has been an invited professor at, and cooperates with, several European universities. He received the Best Theoretical Paper Award, by the Decision Sciences Institute (Athens, 1999). He is member of the Organizing Committee of the International Summer School on MCDA, of which he organized the first (Catania, 1983) and the seventh (Catania, 2000) editions.

Patricia Melin was awarded Doctor of Science (DSc) from the Polish Academy of Sciences. She is a professor of computer science in the Graduate Division, Tijuana Institute of Technology, Tijuana, Mexico. In addition, she is serving as director of Graduate Studies in Computer Science and head of the research group on fuzzy logic and neural networks. Currently, she is vice president of Hispanic American Fuzzy Systems Association (HAFSA) and is also chair of the Mexican Chapter of the Computational Intelligence Society (IEEE). She is also program chair of the IFS 2007 World Congress to be held in Cancun, Mexico. She also belongs to the Committee of Women in Computational Intelligence of the IEEE and to the New York Academy of Sciences. Her research interests are in type-2 fuzzy logic, modular neural networks, pattern recognition, fuzzy control, neuro-fuzzy and genetic-fuzzy hybrid approaches. She has published over 50 journal papers, 5 authored books, 8 edited books, and 150 papers in conference proceedings.

Jerry M. Mendel received the PhD degree in electrical engineering from the Polytechnic Institute of Brooklyn, Brooklyn, NY. Currently, he is professor of electrical engineering at the University of Southern California in Los Angeles, where he has been since 1974. He has published over 450 technical papers and is author and/or editor of eight books, including *Uncertain Rule-based Fuzzy Logic Systems: Introduction and New Directions* (Prentice-Hall, 2001). His present research interests include type-2 fuzzy logic systems and their applications to a wide range of problems, including smart oil field technology and computing with words. He is a life fellow of the IEEE and a distinguished member of the IEEE Control Systems Society. He was president of the IEEE Control Systems Society in 1986, and is presently chairman of the Fuzzy Systems Technical Committee and an elected member of the Administrative Committee of the IEEE Computational Intelligence Society. Among his awards are the 1983 Best Transactions Paper Award of the IEEE Geoscience and Remote Sensing Society, the 1992 Signal Processing Society Paper Award, the 2002 *Transactions on Fuzzy Systems* Outstanding Paper Award, a 1984 IEEE Centennial Medal, an IEEE Third Millennium Medal, and a Pioneer Award from the *IEEE Granular Computing Conference*, May 2006, for outstanding contributions in type-2 fuzzy systems.

Radko Mesiar received his PhD degree from the Comenius University Bratislava and the DSc degree from the Czech Academy of Sciences, Prague, in 1979 and 1996, respectively. He is a professor of mathematics at the Slovak University of Technology, Bratislava, Slovakia. His major research interests are in the area of uncertainty modeling, fuzzy logic, and several types of aggregation techniques, non-additive measures, and integral theory. He is coauthor of a monograph on triangular norms, coeditor of three edited volumes, and author/coauthor of more than 100 journal papers and chapters in edited volumes. He is an associate editor of four international journals. Dr. Mesiar is a member of the European Association for Fuzzy Logic and Technology and of the Slovak Mathematical Society. He is a fellow researcher at UTIA AV CR Prague (since 1995) and at IRAFM Ostrava (since 2005).

Andrea Mesiarová-Zemánková graduated from the Faculty of Mathematics, Physics and Informatics of the Comenius University, Bratislava, in 2002. She defended her PhD thesis in July 2005 at the Mathematical Institute of the Slovak Academy of Sciences, Bratislava. At the moment, she is a researcher at the Mathematical Institute of the Slovak Academy of Sciences. Her major scientific interests are triangular norms and aggregation operators.

Sushmita Mitra is a professor at the Machine Intelligence Unit, Indian Statistical Institute, Kolkata. From 1992 to 1994 she was in the RWTH, Aachen, Germany, as a DAAD fellow. She was a visiting professor in the Computer Science Departments of the University of Alberta, Edmonton, Canada, in 2004 and 2007; Meiji University, Japan, in 1999, 2004, 2005, and 2007; and Aalborg University Esbjerg, Denmark, in 2002 and 2003. Dr. Mitra received the National Talent Search Scholarship (1978–1983) from NCERT, India, the IEEE TNN Outstanding Paper Award in 1994 for her pioneering work in neuro-fuzzy computing, and the CIMPA-INRIA-UNESCO Fellowship in 1996. She is the author of the books *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing and Data Mining: Multimedia, Soft Computing, and Bioinformatics* published by John Wiley. Dr. Mitra has guest edited special issues of journals, and is an associate editor of *Neurocomputing*. She has more than 100 research publications in referred international journals. According to the Science Citation Index (SCI), two of her papers have been ranked third and fifteenth in the list of top-cited papers in engineering science from India during 1992–2001. Dr. Mitra is a senior member of IEEE and a fellow of the Indian National Academy of Engineering. She served in the capacity of program chair, tutorial chair, and as member of program committees of many international conferences. Her current research interests include data mining, pattern recognition, soft computing, image processing, and bioinformatics.

Javier Montero is an Associate Professor at the Department of Statistics and Operational Research, Faculty of Mathematics, Complutense University of Madrid, Spain. He holds a PhD in mathematics from Complutense University since 1982. He is the author of more than 50 research papers in refereed journals such as *Behavioral Science*, *European Journal of Operational Research*, *Fuzzy Sets and Systems*, *Approximate Reasoning*, *Intelligent Systems*, *General Systems*, *Kybernetes*, *IEEE Transactions on Systems, Man and Cybernetics*, *Information Sciences*, *International Journal of Remote Sensing*, *Journal of Algorithms*, *Journal of the Operational Research Society*, *Lecture Notes in Computer Science*, *Mathware and Soft Computing*, *New Mathematics and Natural Computation*, *Omega-International Journal of Management Sciences*, *Soft Computing and Uncertainty*, and *Fuzziness and Knowledge-Based Systems*, plus more than 40 papers as book chapters. His research interests are in preference representation, multicriteria decision making, group decision making, system reliability theory, and classification problems, mainly viewed as application of fuzzy sets theory.

Hung Son Nguyen is an Assistant Professor at Warsaw University and a member of International Rough Set society. He received his MS and PhD from Warsaw University in 1994 and 1997, respectively. His main research interests are fundamentals and applications of rough set theory, data mining, text mining, granular computing, bioinformatics, intelligent multiagent systems, soft computing, and pattern recognition. On these topics he has published more than 80 research papers in edited books, international journals, and conferences. He is the coauthor of ‘IEEE/WIC/ACM International Conference on Web Intelligence (WI 2005) Best Paper Award.’ Dr. Hung Son Nguyen is a member of the editorial board of the international journals *Transaction on Rough Sets*, *Data Mining and Knowledge Discovery*, and *ERCIM News*, and the assistant to the editor-in-chief of *Fundamenta Informaticea*. He has served as a program cochair of RSCTC’06, as a PC member, and a reviewer of various other conferences and journals.

Sinh Hoa Nguyen is an Assistant Professor at the Polish Japanese Institute of Information Technology in Warsaw Poland. She received her MSc and PhD from Warsaw University in 1994 and 2000, respectively. Her research interests include rough set theory, data mining, granular computing, intelligent multiagent systems, soft computing, and pattern recognition; on these topics she has published more than 50 research papers in edited books, international journals, and conferences. Recently, she has concentrated on developing efficient methods for learning multilayered classifiers from data, using concept ontology as domain knowledge. Dr. Sinh Hoa Nguyen has also served as a reviewer of many journals and a PC member of various conferences.

Trung T. Nguyen has received MSc in computer science from the Department of Mathematics of the Warsaw University in 1993. He is currently completing a PhD thesis at the Department of Mathematics

of the Warsaw University, while working at the Polish–Japanese Institute of Information Technology in Warsaw, Poland. His principal research interests include rough sets, handwritten recognition, approximate reasoning, and machine learning.

Hajime Nobuhara is Assistant Professor in the Department of Intelligent Interaction Technologies of Tsukuba University. He was also Assistant Professor in Tokyo Institute of Technology and postdoctoral fellow c/o University of Alberta (Canada) and a member of the Institute of Electrical and Electronics Engineers (IEEE). His interests mainly concern fuzzy logic and its applications to image processing, publishing numerous, and various papers in famous international journals.

Hannu Nurmi worked as a research assistant of Academy of Finland during 1971–1973. He spent the academic year 1972–1973 as a senior ASLA-Fulbright fellow at Johns Hopkins University, Baltimore, MD. In 1973–1974, he was an assistant at the Department of Political Science, University of Turku. From 1974 till 1995, Nurmi was the Associate Professor of methodology of the social sciences at the University of Turku. In 1978 he was a British Academy Wolfson fellow at the University of Essex, UK. From 1991 till 1996, he was the dean of Faculty of Social Sciences, University of Turku. From 1995 onward, he has been the professor of political science, University of Turku. The fall quarter of 1998 Nurmi spent as the David and Nancy Speer/Government of Finland Professor of Finnish Studies at University of Minnesota, USA. Currently, i.e., from 2003 till 2008, he is on leave from his political science chair on being nominated an academy professor in the Academy of Finland. Nurmi is the author or coauthor of 10 scientific monographs and well over 150 scholarly articles. He has supervised or examined some 20 PhD theses in Finland, Norway, Germany, Czech Republic, and the Netherlands. He is an editorial board member in four international journals and in one domestic scientific one.

Miguel Pagola is an associate lecturer at the Department of Automatics and Computation, Public University of Navarra, Spain. He received his MSc in industrial engineering at the Public University of Navarra in 2000. He enjoyed a scholarship within a research project developing intelligent control strategies from 2000 to 2002 and then he joined the Public University of Navarra as associate lecturer. His research interests are fuzzy techniques for image processing, fuzzy set theory, interval type-2 fuzzy set theory, fuzzy control systems, genetic algorithms, and neural networks. He was a research visitor at the DeMonfort University. He is a member of the European Society for Fuzzy Logic and Technology (EUSFLAT).

Sankar K. Pal is the director of the Indian Statistical Institute, Calcutta. He is also a professor, distinguished scientist, and the founding head of Machine Intelligence Unit. He received the MTech and PhD degrees in radio physics and electronics in 1974 and 1979, respectively, from the University of Calcutta. In 1982 he received another PhD in electrical engineering along with DIC from Imperial College, University of London. Professor Pal is a fellow of the IEEE, USA, Third World Academy of Sciences, Italy, International Association for Pattern Recognition, USA, and all the four National Academies for Science/Engineering in India. His research interests include pattern recognition and machine learning, image processing, data mining, soft computing, neural nets, genetic algorithms, fuzzy sets, rough sets, web intelligence, and bioinformatics. He is a coauthor of ten books and about three hundred research publications. Professor Pal has served as an editor, associate editor, and a guest editor of a number of journals including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Neural Networks*, *IEEE Computer*, *Pattern Recognition Letters*, *Neurocomputing*, *Information Sciences*, and *Fuzzy Sets and Systems*.

Gabriella Pasi completed her Laurea degree in computer science at the Università degli Studi di Milano, Italy, and the PhD degree in computer science at the Université de Rennes, France. She worked as a researcher at the National Council of Research in Italy from April 1985 until February 2005. She is now an Associate Professor at the Università Degli Studi di Milano Bicocca, Milano, Italy. Her research activity mainly concerns the modeling and design of flexible systems (i.e., systems able to manage imprecision and uncertainty) for the management and access to information, such as information retrieval systems, information filtering systems, and database management systems. She also works at the definition of

techniques of multicriteria decision making and group decision making. She is a member of organizing and program committees of several international conferences. She has coedited seven books and several special issues of international journals. She has published more than 150 papers in international journals, books, and proceeding of international conferences. She is coeditor of seven books and several special issues of international journals. Since 2001 she is a member of the editorial board of the journals *Mathware and Soft Computing* and *ACM Applied Computing Review*, and since 2006 she has been a member of the editorial board of *Fuzzy Sets and Systems*. She has been the coordinator of the European Project PENG (Personalized News Content Programming). This is a STREP (Specific Targeted Research or Innovation Project), within the VI Framework Programme, Priority II, Information Society Technology. She organized several international events among which the European Summer School in Information Retrieval (ESSIR 2000) and FQAS 2006, and she co-organizes every year the track ‘Information Access and Retrieval’ within the ACM Symposium on Applied Computing.

James F. Peters, PhD (1991), is a Full Professor in the Department of Electrical and Computer Engineering (ECE) at the University of Manitoba. Currently, he is coeditor-in-chief of the *Transactions on Rough Sets* journal published by Springer, cofounder and researcher in the Computational Intelligence Laboratory in the ECE Department (1996–), and current member of the Steering Committee, International Rough Sets Society. Since 1997, he has published numerous articles about approximation spaces, systems that learn adaptively, and classification in refereed journals, edited volumes, international conferences, symposia, and workshops. His current research interests are in approximation spaces (near sets), pattern recognition (ethology and image processing), rough set theory, reinforcement learning, biologically inspired designs of intelligent systems (vision systems that learn), and the extension of ethology (study of behavior of biological organisms) in the investigation of intelligent systems behavior.

Lech Polkowski was born in 1946 in Poland. He graduated from Warsaw University of Technology in 1969 and from Warsaw University in 1977. He obtained his PhD in theoretical mathematics from Warsaw University in 1982, Doctor of Science (habilitation) in 1994 in mathematical foundation of computer science, and has been professor titular since 2000. Professor Polkowski lectured in Warsaw University of Technology, Ohio University (Athens, Ohio, USA), and Polish–Japanese Institute of Information Technology. His papers are quoted in monographs of topology and dimension theory. Since 1992 he has been interested in rough sets, mostly foundations and relations to theoretical paradigms of reasoning. He has published extensively on the topology of rough set spaces, logics for reasoning with rough sets, mereological foundations of rough sets, rough mereology, granulation theory, granulated data systems, multiagent systems, and rough cognitive computing.

Anna Maria Radzikowska is an Assistant Professor at the Faculty of Mathematics and Information Science, Warsaw University of Technology (Poland). Her research interests include logical and algebraic methods for representing, analyzing, and reasoning about knowledge. Currently, her research focuses on hybrid fuzzy rough approaches to analyzing data in information systems.

Sheela Ramanna received a PhD in computer science from Kansas State University. She is a Full Professor and past chair of the Applied Computer Science Department at the University of Winnipeg, Canada. She serves on the editorial board of the *TRS Journal* and is one of the program cochairs for *RSFDGrC'07 Conference*. She is currently the secretary for the International Rough Set Society. She has served on program committees of many past international conferences, including *RSEISP 2007*, *ECML/PKDD 2007*, *IAT/WI 2007*, and *IICAI2007*. She has published numerous papers on rough set methods in software engineering and intelligent systems. Her research interests include rough set theory in requirements engineering and software quality and methodologies for intelligent systems.

Jaroslav Ramík holds an MSc and PhD degree in mathematics from the Faculty of Mathematics and Physics, Charles University in Prague (Czech Republic). He is the author of numerous monographs, books, papers, and research works in optimization, including fuzzy mathematical programming, multicriteria decision making, fuzzy control, and scheduling. Since 1990 he has been a professor and head of the

Department of Mathematical Methods in Economics at the Silesian University Opava, School of Business Administration in Karviná.

Leon Reznik is a professor of computer science at the Rochester Institute of Technology, New York. He received his BS/MS degree in computer control systems in 1978 and a PhD degree from St Petersburg Polytechnic Institute in 1983. He has worked in both industry and academia in the areas of control, system, software and information engineering, and computer science. Professor Reznik is an author of the textbook *Fuzzy Controllers* (Butterworth-Heinemann, 1997) and an editor of *Fuzzy System Design: Social and Engineering Applications* (Physica-Verlag, 1998), *Soft Computing in Measurement and Information Acquisition* (Springer, 2003), and *Advancing Computing and Information Sciences* (Cary Graphic Arts Press, 2005). Dr. Reznik's research has been concentrated on study and development of fuzzy and soft computing models applications. He pioneered a new research direction where he is applying fuzzy and soft computing models to describe measurement results with applications in sensor networks.

Shounak Roychowdhury received his BEng in computer science and engineering from Indian Institute of Science, Bangalore, India, 1990. In 1997, he received MS in computer science from the University of Tulsa, OK. In between he has worked as researcher in LG's research laboratories in Seoul, Korea. Currently, he is a senior member of technical staff at Oracle Corporation. His current interests include fuzzy theory, data mining, and databases. At the same time he is also a part-time graduate student at the University of Texas at Austin.

Susanne Saminger-Platz graduated from the Technical University Vienna, Austria, in 2000. She defended her PhD in mathematics at the Johannes Kepler University, Linz, Austria, in 2003. She is an Assistant Professor at the Department of Knowledge-Based Mathematical Systems, Johannes Kepler University, Linz, Austria, and currently on a sabbatical year at the Dipartimento di Matematica 'Ennio De Giorgi' of the Universit del Salento, Italy. Her major research interests focus on the preservation of properties during uni- and bipolar aggregation processes and therefore relate to such diverse fields as fuzzy logic, preference and uncertainty modeling, decision making, and probabilistic metric spaces. She is author/coauthor of several journal papers and chapters in edited volumes. She is further a member of the European Association for Fuzzy Logic and Technology (EUSFLAT), of the EURO Working Group on Fuzzy Sets (EUROFUSE), and of the Austrian Mathematical Society.

Salvatore Sessa is professor of computer science at the Faculty of Architecture of Naples University Federico II. His main research interests are devoted to applications of fuzzy logic to image processing, approximate reasoning, geographic information systems, and fuzzy systems. He has published and edited several monographies and numerous papers on well-known international journals. He is coeditor of the section 'Recent Literature' of the journal *Fuzzy Sets and Systems*.

Simon C.K. Shiu is an Assistant Professor at the Department of Computing, Hong Kong Polytechnic University, Hong Kong. He received an MSc degree in computing science from the University of Newcastle Upon Tyne, UK, in 1985, an MSc degree in business systems analysis and design from City University, London, in 1986, and a PhD degree in computing from Hong Kong Polytechnic University in 1997. He worked as a system analyst and project manager between 1985 and 1990 in several business organizations in Hong Kong. His current research interests include case-based reasoning, machine learning, and soft computing. He has coguest edited a special issue on soft case-based reasoning of the journal *Applied Intelligence*. Dr. Shiu is a member of the British Computer Society and the IEEE.

Luke David Simoni has a BS degree in chemical engineering from the Michigan Technological University. He is currently a PhD student at the University of Notre Dame, where he holds an Arthur J. Schmitt Presidential Fellowship.

Roman Slowiński, professor and founding head of the Laboratory of Intelligent Decision Support Systems within the Institute of Computing Science, Poznan University of Technology, Poland. He received

the PhD in operations research and habilitation in computing science from the Poznan University of Technology, in 1977 and 1981, respectively. He has been professor on European Chair at the University of Paris Dauphine and invited professor at the Swiss Federal Institute of Technology in Lausanne and at the University of Catania. His research concerns operational research and artificial intelligence, including multiple-criteria decision analysis, preference modeling, project scheduling, knowledge-based decision support in medicine, technology, and economics, and rough set theory approach to knowledge and data engineering. He is laureate of the EURO Gold Medal (1991) and Doctor *Honoris Causa* of Polytechnic Faculty of Mons (2000) and University of Paris Dauphine (2001). Since 1999, he has been editor-in-chief of the *European Journal of Operational Research*. Since 2004, he has been elected member of the Polish Academy of Sciences. In 2005, he received the most prestigious Polish scientific award from the Foundation for Polish Science.

Laerte Sorini is an Assistant Professor in Urbino University (Italy), where he teaches mathematics and informatics. He received his BA in mathematics from Bologna University. His current research deals with numerical aspects in simulation of stochastic differential equations and in implementation of fuzzy systems. When not teaching or writing, Laerte enjoys political debate and motorcycling.

Mark Allen Stadtherr is professor of chemical and biomolecular engineering at the University of Notre Dame. He has a BChE degree from the University of Minnesota and a PhD degree from the University of Wisconsin. He was awarded the 1998 Computing in Chemical Engineering Award by the American Institute of Chemical Engineers. His research interests include the application of interval methods to global optimization, non-linear algebraic equation solving, and systems of ordinary differential equations.

Luciano Stefanini is a Full Professor at the University of Urbino (Italy) where he currently teaches mathematics and informatics. He received his BA in mathematics from the University of Bologna in 1974 and specialized in numerical analysis in 1975. From 1975 to 1982 he has been with the ENI Group for industrial research in computational mathematics and operations research. In 1982 he started with the University of Urbino. He has directed various research and applied projects in industry and in public sectors. His research activity has produced papers covering fields in numerical analysis and statistical computing, operations research, combinatorial optimization and graph theory, distribution management and transportation, geographic information systems, mathematical finance and game theory, fuzzy numbers and calculus.

Jaroslaw Stepaniuk holds a PhD degree in mathematical foundations of computer science from the University of Warsaw in Poland and a Doctor of Science (habilitation) degree in computer science from the Institute of Computer Science Polish Academy of Sciences. Jaroslaw Stepaniuk is associate professor in the Faculty of Computer Science at Bialystok University of Technology and is the author of more than 130 scientific publications. His areas of expertise include reasoning with incomplete information, approximate reasoning, soft computing methods and applications, rough sets, granular computing, synthesis and analysis of complex objects, intelligent agents, knowledge discovery systems, and advanced data mining techniques.

C.D. Stylios is an electrical engineer (Aristotle University of Thessaloniki, 1992); he received his PhD from the Department of Electrical and Computer Engineering, University of Patras, Greece (1999). He is Assistant Professor at the Department of Informatics and Telecommunications Technology, Technological Education Institute of Epirus, Greece, and director of Knowledge and Intelligent Computing Laboratory (March 2006–today). Since 1999, he is a senior researcher at Laboratory for Automation and Robotics, University of Patras, Greece, and since 2004 is an external consultant at Patras Science Park. He was adjunct assistant professor at Computer Science Department, University of Ioannina, Greece (2000–2004). He has published over 60 journals and conference papers, book chapters, and technical reports. His research interests include soft computing methods, computational intelligent techniques, modeling of complex systems, intelligent systems, decision support systems, hierarchical systems, and artificial

intelligence techniques for medical applications. He is a member of IEEE and the National Technical Chamber of Greece.

Peter Sussner is an Assistant Professor at the Department of Applied Mathematics of the State University of Campinas. He also acts as a researcher for the Brazilian national science foundation CNPq and holds a membership of the IEEE Computational Intelligence Society. He has previously worked as a researcher at the Center of Computer Vision and Visualization at the University of Florida where he completed his PhD in mathematics – partially supported by a Fulbright Scholarship – in 1996. Peter Sussner has regularly published articles in refereed international journals, book chapters, and conference proceedings in the areas of artificial neural networks, fuzzy systems, computer vision, mathematical imaging, and global optimization. His current research interests include neural networks, fuzzy systems, mathematical morphology, and lattice algebra.

Piotr Synak is one of the founders of Infobright, Inc., which has developed market-leading compression technologies implemented through a revolutionary, rough set theory based view of databases and data storage. He obtained his PhD in computer science in 2004 from the Polish Academy of Sciences. Since 1996 he has worked at the Polish–Japanese Institute of Information Technology in Poland and currently holds the position of Assistant Professor. He is the author of several papers related to rough sets and spatiotemporal reasoning.

Manuel Tarrazo teaches corporate finance and investments courses at the School of Business of the University of San Francisco, where he is an Associate Professor of finance. His research interest includes the application of conventional (calculus, probabilistic methods, combinatorial optimization) and emerging methodologies (fuzzy sets, approximate equations, neural networks) to portfolio optimization, fixed-income analysis, asset allocation, and corporate financial planning. He has published research in the following journals: *The European Journal of Operational Research*, *Applied Numerical Mathematics*, *Fuzzy Optimization and Decision Making*, *Financial Services Review*, *Advances in Financial Planning and Forecasting*, *Advances in Financial Education*, *Financial Technology*, *International Journal of Business*, *Journal of Applied Business and Economics*, *Midwest Review of Finance and Insurance*, *Research Papers in Management and Business*, *Revista Alta Dirección*, and *The International Journal of Business Research*. In addition, he has made over 35 professional presentations, earning three ‘Best Study’ awards, and published the following monographs: ‘Practical Applications of Approximate Equations in Finance and Economics,’ Quorum Publishers, Greenwood Publishing Group, January 2001; ‘Advanced Spreadsheet Modeling for Portfolio Management,’ coauthored with Gregory Alves, Kendall/Hunt, 1996. Professor Tarrazo is a native from Spain, where he obtained a *Licenciatura* at the Universidad Complutense de Madrid. He worked as a financial manager before completing his doctoral education at the State University of New York at Albany, NY.

İ. Burhan Türkşen joined the Faculty of Applied Science and Engineering at the University of Toronto and became professor emeritus in 2003. In December 2005, he was appointed as the head of department of Industrial Engineering at TOBB Economics and Technology University in Ankara Turkey. He was the president of International Fuzzy Systems Association (IFSA) during 1997–2001 and past president of IFSA during 2001–2003. Currently, he is the president, CEO, and CSO of Information Intelligence Corporation (IIC). He received the outstanding paper award from NAFIPS in 1986, ‘L.A. Zadeh Best Paper Award’ from Fuzzy Theory and Technology in 1995, ‘Science Award’ from Middle East Technical University, and an ‘Honorary Doctorate’ from Sakarya University. He is a foreign member in the Academy of Modern Sciences. Currently, he is a fellow of IFSA, IEEE, and WIF (World Innovation Foundation). He has published around 300 papers in scientific journals and conference proceedings. More than 600 authors have made references to his published works. His book entitled *An Ontological and Epistemological Perspective of Fuzzy Theory* was published by Elsevier in January 2006.

Julio J. Valdés is a senior research officer at the National Research Council Canada, Institute for Information technology. He has a PhD in mathematics and his areas of interest are artificial intelligence

(mathematical foundations of uncertainty processing and machine learning), computational intelligence (fuzzy logic, neural networks, evolutionary algorithms, rough sets, probabilistic reasoning), data mining, virtual reality, hybrid systems, image and signal processing, and pattern recognition. He is member of the IEEE Computational Intelligence Society and the International Neural Network Society. He has been coeditor of two special issues of the *Neural Network Journal* and has more than 150 publications in journals and international conferences.

Marcos Eduardo Valle recently completed his PhD in applied mathematics at the State University of Campinas (UNICAMP), Brazil, under the supervision of Dr. Sussner. His doctoral research was financially supported by a scholarship from the Brazilian national science foundation CNPq. Currently, Dr. Valle is working as a visiting professor, funded by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), at the Department of Applied Mathematics at the State University of Campinas. His research interests include fuzzy set theory, neural networks, and mathematical morphology.

José Valente de Oliveira received the PhD (1996), MSc (1992), and the ‘Licenciado’ degrees in electrical and computer engineering, all from the IST, Technical University of Lisbon, Portugal. Currently, he is an Assistant Professor in the Faculty of Science and Technology of the University of Algarve, Portugal, where he served as deputy dean from 2000 to 2003. Dr. Valente de Oliveira was recently appointed director of the UALG-iLAB, The University of Algarve Informatics Lab, a research laboratory whose pursuits in what concerns computational intelligence includes fuzzy sets, fuzzy and intelligent systems, data mining, machine learning, and optimization. During his first sabbatical year (2004/2005) he was with the University of Alberta, Canada, as a visiting professor. Dr. Valente de Oliveira is an associated editor of the *Journal of Intelligent & Fuzzy Systems* (IOS Press) and coeditor of the book *Advances in Fuzzy Clustering and Its Applications* (Wiley 2007).

Mario Veniero, BSc, is senior software engineer at the LASA research group at the University of Salerno. He is an IEEE member and his main research interests are in the area of software agents, soft computing, semantic web, and distributed systems. Since 1998 he was investigating the area of software agents and involved in a number of industrial R&D and academic research projects based on hybrid approach of computational intelligence and agent technologies. He is author of several of original papers in book chapters and in international conference proceedings.

Jean Vignes is emeritus professor at the Pierre et Marie Curie University in Paris (UPMC) since 1998. He has received the diploma of mathématiques supérieures from the University of Toulouse in 1956 and the diploma of research engineer from the French Petroleum Institute (IFP) school in 1959. He was Docteur ès sciences from UPMC in 1969. He has been professor of computer sciences both at IFP school from 1964 to 1998 and at UPMC from 1969 to 1998. Furthermore, he was scientific adviser at IFP from 1969 to 1998. His interest areas include computer arithmetic, round-off error propagation, and validation of numerical software. He has created a stochastic method called CESTAC (Contrôle et Estimation Stochastique des Arrondis de Calcul) for estimating the effect of round-off error propagation and uncertainties of data in every computed result which is at the origin of a software named CADNA (Control of Accuracy and Debugging for Numerical Applications), which automatically implements the CESTAC method in scientific codes. The CESTAC method is also the basis of stochastic arithmetic. He has obtained the award of computer sciences from the French Academy of Sciences for his work in the field of the estimation of the accuracy of computed results. He was also vice president of International Association for Mathematics and Computers in Simulation (IMACS). He is a member of the editorial boards of *Mathematics and Computers in Simulation*, *Applied Numerical Mathematics*, *Numerical Algorithms*, and the *International Journal of Pure and Applied Mathematics*. He is an honorary member of IMACS.

Junzo Watada received his BSc and MS degrees in electrical engineering from Osaka City University, Japan, and PhD on ‘fuzzy analysis and its applications’ from Osaka Prefecture University, Japan. He is a professor of management engineering, knowledge engineering, and soft computing at Graduate School of Information, Production & Systems, Waseda University, since 2003, after having contributed for 13 years

as a professor of human informatics and knowledge engineering, to the School of Industrial Engineering at Osaka Institute of Technology, Japan. He was with Faculty of Business Administration, Ryukoku University, for 8 years. Before moving to academia, he was with Fujitsu Ltd. Co., where he worked on development of software systems as a senior system engineer for 7 years.

Arkadiusz Wojna is an Assistant Professor at the Institute of Informatics, Warsaw University. His research interests include machine learning, analogy-based reasoning, decision support systems, data mining, and knowledge discovery. He received the PhD degree in computer science from Warsaw University in 2005. He is the author and coauthor of conference and journal publications on rough sets, analogy-based reasoning, and machine learning and coauthor of the rough set exploration system. He served on the program committees of the *International Conference on Rough Sets and Current Trends in Computing* (RSCTC-2006), the *International Conference on Rough Sets and Knowledge Technology* (RSKT-2006), the *Joint Rough Set Symposium* (JRS-2007), and the *Indian International Conference on Artificial Intelligence* (IICAI-2005 and IICAI-2007).

Yiyu Yao received his BEng (1983) in Computer Science from Xi'an Jiaotong University, and MSc (1988) and PhD (1991) in computer science from the University of Regina. Currently, he is a professor of computer science with the Department of Computer Science, University of Regina, Canada, and an adjunct professor of International WIC Institute, Beijing University of Technology, Xi'an Jiaotong University, and Chongqing University of Posts and Telecommunication. Dr. Yao's research interests include web intelligence, information retrieval, uncertainty management (fuzzy sets, rough sets, interval computing, and granular computing), data mining, and intelligent information systems. He has published over 200 papers in international journals and conferences and has been invited to give talks at many international conferences and universities.

Sawomir Zadrony is an Associate Professor (PhD 1994, DSc 2006) at the Systems Research Institute, Polish Academy of Sciences. His current scientific interests include applications of fuzzy logic in database management systems, information retrieval, decision support, and data analysis. He is the author and coauthor of about 100 journal and conference papers. He has been involved in the design and implementation of several prototype software packages. He is also a teacher at the Warsaw School of Information Technology in Warsaw, Poland, where his interests focus on information retrieval and database management systems.

Bo Zhang, computer scientist, is a fellow of Chinese Academy of Sciences. He was born in March 1935. He is a professor of Computer Science and Technology Department of Tsinghua University, Beijing, China. In 1958 he graduated from Automatic Control Department of Tsinghua University. From 1980 to 1982, he visited University of Illinois at Urbana – Champaign, USA, as a scholar. Now he serves as the chairman of Academic Committee of Information Science and Technology College in Tsinghua University.

Ling Zhang, computer scientist. He was born in May 1937. He is a professor of Computer Science Department of Anhui University, Hefei, China. In 1961 he graduated from Mathematics and Astronomy Department of Nanjing University, China. Now he serves as the director of Artificial Intelligence Institute, Anhui University.

Part One

Fundamentals and Methodology of Granular Computing Based on Interval Analysis, Fuzzy Sets and Rough Sets

1

Interval Computation as an Important Part of Granular Computing: An Introduction

Vladik Kreinovich

1.1 Brief Outline

The main goal of this chapter is to introduce interval computations to people who are interested in using the corresponding techniques. In view of this goal, we will not only *describe* these techniques, but also do our best to outline the *problems* for which these techniques have been originally invented.

We start with explaining why computations in general are needed in practice. Then, we describe the uncertainty related to all these practical applications and, in particular, interval uncertainty. This will bring us to the main problem of interval computations.

In the following sections, we will briefly describe the history of interval computations, main interval techniques, and we list a few typical applications of these techniques.

1.2 Why Computations Are Needed in Practical Problems: A Brief Reminder

In accordance with the above outline, before we explain the specific role of *interval* computations, we will recall where and why *computations* in general are needed.

Let us recall what practical problems we need to solve in the first place. To understand why computations are needed in practice, let us recall what practical problems we need to solve. Crudely speaking, most of the practical problems can be classified into three classes:

- We want to *learn* what is happening in the world; in particular, we want to know the numerical values of different quantities (distances, masses, charges, coordinates, etc.).
- On the basis of these values, we would like to *predict* how the state of the world will change over time.
- Finally, we would like to find out what *changes* we need to make in the world so that these changes will lead to the desired results.

It should be emphasized that this classification is very crude: a real-life problem often involves solving subproblems of all three above-described types.

The above classification is related to the distinction between science and engineering. The above classification may sound unusual, but in reality, it is related to the well-known classification of creative activity into engineering and science:

- The tasks of learning the current state of the world and predicting the future state of the world are usually classified as *science*.
- The tasks of finding the appropriate change are usually classified as *engineering*.

Example.

- Measuring the river flow at different locations and predicting how this river flow will change over time are problems of science.
- Finding the best way to change this flow (e.g., by building dams or levees) is a problem of engineering.

Computations are needed for all three classes of problems. In the following text, we will analyze the problems of these three types one by one. We will see that in all three cases, a large amount of computation is needed.

How we learn the current state of the world: sometimes, it is (relatively) straightforward.

Let us start with the first class of practical problems: the problem of learning the state of the world. As we have mentioned, this means, in particular, that we want to know the numerical values of different quantities y that characterize this state.

Some quantities y we can simply directly measure. For example, when we want to know the current state of a patient in a hospital, we can measure the patient's body temperature, blood pressure, weight, and many other important characteristics.

In some situations, we do not even need to measure: we can simply ask an expert, and the expert will provide us with an approximate value \tilde{y} of the quantity y .

How we learn the current state of the world: sometimes, it is not easy. Some quantities we can simply directly measure. However, many other quantities of interest are difficult or even important to measure or estimate directly.

Examples. Examples of such quantities include the amount of oil in a given well or the distance to a star. Let us explain this situation on the example of measuring distances:

- We can estimate the distance between two nearby houses by simply placing a measuring tape between them.
- If we are interested in measuring the distance between two cities, in principle, it is possible to do it directly, by driving or walking from one to another. (It is worth mentioning that while such a direct measurement is possible *in principle*, it is *not* a reasonable practical way.)
- If we are interested in measuring the distance to a star, then, at present, it is not possible to directly measure this distance.

How we can measure difficult-to-measure quantities. Since we cannot directly measure the values of these quantities, the only way to learn some information about them is to

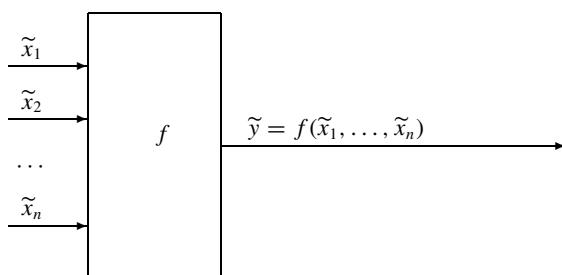
- measure (or ask an expert to estimate) some other easier-to-measure quantities x_1, \dots, x_n , and then
- estimate y based on the measured values \tilde{x}_i of these auxiliary quantities x_i .

Examples.

- To estimate the amount of oil in a given well, we perform *seismic* experiments: we set up small explosions at some locations and measure the resulting seismic waves at different distances from the location of the explosion.

- To find the distance to a faraway star, we measure the direction to the star from different locations on Earth (and/or at different seasons) and the coordinates of (and the distances between) the locations of the corresponding telescopes.

To learn the current value of the desired quantity, we often need a lot of computations. To estimate the value of the desired quantity y , we must know the relation between y and the easier-to-measure (or easier-to-estimate) quantities x_1, \dots, x_n . Specifically, we want to use the estimates of x_i to come up with an estimate for y . Thus, the relation between y and x_i must be given in the form of an *algorithm* $f(x_1, \dots, x_n)$ which transforms the values of x_i into an estimate for y . Once we know this algorithm f and the measured values \tilde{x}_i of the auxiliary quantities, we can estimate y as $\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n)$.



In different practical situations, we have algorithms f of different complexity. For example, to find the distance to a star, we can usually have an explicit analytical formula coming from geometry. In this case, f is a simple formula.

On the other hand, to find the amount of oil, we must numerically solve a complex partial differential equation. In this case, f is a complex iterative algorithm for solving this equation.

There are many such practical cases when the algorithm f requires a lot of computations. Thus, the need to learn the current state of the world indeed often leads to the need to perform a large number of computations.

Comment: the notion of indirect measurement. We started with the situation in which we cannot estimate the value of the desired quantity y by simply directly measuring (or directly estimating) this value. In such situations, we can use the above two-stage process, as a result of which we get an indirect estimate for y .

In the case when the values x_i are obtained by measurement, this two-stage process does involve measurement. To distinguish it from *direct* measurements (i.e., measurements which directly measure the values of the desired quantity), the above two-stage process is called an *indirect* measurement.

Computations are needed to predict the future state of the world. Once we know the values of the quantities y_1, \dots, y_m which characterize the current state of the world, we can start predicting the future state of the world, i.e., the future values of these quantities.

To be able to predict the future value z of each of these quantities, we must know exactly how this value z depends on the current values y_1, \dots, y_m . Specifically, we want to use the known estimates \tilde{y}_i for y_i to come up with an estimate for z . Thus, the relation between z and y_i must be given in the form of an *algorithm* $g(y_1, \dots, y_m)$ which transforms the values of y_i into an estimate for z . Once we know this algorithm g and the estimates \tilde{y}_i for the current values of the quantities y_i , we can estimate z as $\tilde{z} = g(\tilde{y}_1, \dots, \tilde{y}_m)$.

Again, the corresponding algorithm g can be very complicated and time consuming. So, we often need a large number of computations to make the desired predictions.

This is, e.g., how weather is predicted now: weather prediction requires so many computations that it can only be performed on fast supercomputers.

The general notion of data processing. So far, we have analyzed two different classes of practical problems:

- the problem of *learning* the current state of the world (i.e., the problem of indirect measurement) and
- the problem of *predicting* the future state of the world.

From the *practical* viewpoint, these two problems are drastically different. However, as we have seen, from the *computational* viewpoint, these two problems are very similar. In both problems,

- we start with the estimates $\tilde{x}_1, \dots, \tilde{x}_n$ for the quantities x_1, \dots, x_n , and then
- we apply the known algorithm f to these estimates, resulting in an estimate $\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n)$ for the desired quantity y .

In both cases, this algorithm can be very time consuming. The corresponding (often time consuming) computational part of each of these two classes of problems – applying a known algorithm to the known values – is called *data processing*.

Comment. Since the computational parts of these two classes of problems are similar, it is important to describe the difference between these two classes of problems. As we can see from the above descriptions, the only *difference* between the two classes is where the original inputs \tilde{x}_i come from:

- In the problem of learning the current state of the world, the inputs \tilde{x}_i come from *direct* measurements (or direct expert estimation).
- In contrast, in the problem of predicting the future state of the world, the inputs \tilde{y}_i come from the learning stage – e.g., they may come from *indirect* measurements.

Decision making, design, control. Once we know the current state of the world and we know how to predict the consequences of different decisions (designs, etc.), it is desirable to find a decision (design, etc.) which guarantees the given results.

Depending on what we want from this design, we can subdivide all the problems from this class into two subclasses.

In both subclasses, the design must satisfy some constraints. Thus, we are interested in finding a design that satisfies all these constraints.

- In some practical situations, satisfaction of all these constraints is all we want. In general, there may be several possible designs which satisfy given constraints. In the problems from the first subclass, we do not have any preferences for one of these designs – any one of them will suffice. Such problems are called the problems of *constraint satisfaction*.
- In other practical situations, we do have a clear preference between different designs x . This preference is usually described in terms of an *objective function* $F(x)$ – a function for which more preferable designs x correspond to larger values of $F(x)$. In such situation, among all the designs which satisfy given constraints, we would like to find a design x for which the value $F(x)$ of the given objective function is the largest. Such problems are called *optimization problems*.

Both constraint satisfaction and optimization often require a large number of computations (see, e.g., [1]).

Comment. Our main objective is to describe interval computations. They were originally invented for the first two classes of problems, i.e., for data processing, but they turned out to be very useful for the third class (constraint satisfaction and optimization) as well.

1.3 In Real-Life Computations, We Need to Take Uncertainty into Account

Need for computations: reminder. In the previous section, we described the importance of computations. In particular, computations constituting data processing process the values which come from measurements (direct or indirect) and from expert estimations.

Let us start with the problem of learning the values of the physical quantities. Let us start with the problems from the first class – the problems of learning the values of the physical quantities. In these problems, computations are needed to transform the results $\tilde{x}_1, \dots, \tilde{x}_n$ of direct measurements (or direct expert estimations) into the estimate $\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n)$ of the desired quantity y .

In the case of both measurements and expert estimates, the estimates \tilde{x}_i are only approximately equal to the (unknown) actual values x_i of the corresponding quantities. Let us elaborate on this statement.

Measurements are never exact.

- From the *philosophical* viewpoint, measurements cannot be exact because
 - the actual value of the quantity is a general real number; so, in general, we need *infinitely* many bits to describe the exact value, while
 - after every measurement, we gain only a *finite* number of bits of information (e.g., a finite number of binary digits in the binary expansion of the number).
- From the *physical* viewpoint, there is always some difficult-to-delete noise which is mixed with the measurement results.

Expert estimates are never absolutely exact either.

- First of all, as with the measurements, expert estimates cannot be absolutely exact, because an expert generates only a finite amount of information.
- Second, from the commonsense viewpoint, experts are usually even less accurate than (sometimes superprecise) are measuring instruments.

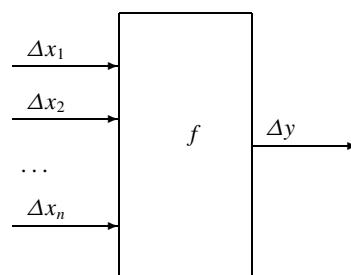
In both cases, there is usually a non-zero approximation error. The difference $\Delta x_i \stackrel{\text{def}}{=} \tilde{x}_i - x_i$ between the (approximate) estimate \tilde{x}_i and the (unknown) actual value x_i of the quantity x_i is called the *approximation error*.

In particular, if \tilde{x}_i is obtained by measurement, this difference is called the *measurement error*.

Uncertainty in inputs leads to uncertainty in the result of data processing. We assume that the quantities x_1, \dots, x_n that we directly measure or directly estimate are related to the desired quantity y by a known relation $y = f(x_1, \dots, x_n)$.

Because of this relation, we estimate the value y as $\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n)$. Since the values \tilde{x}_i are, in general, different from the (unknown) actual values x_i , the result $\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n)$ of applying the algorithm f to the estimates \tilde{x}_i is, in general, different from the result $y = f(x_1, \dots, x_n)$ of applying this algorithm to the actual values x_i . Thus, the estimate \tilde{y} is, in general, different from the actual value y of the desired quantity: $\Delta y \stackrel{\text{def}}{=} \tilde{y} - y \neq 0$.

It is therefore desirable to find out the uncertainty Δy caused by the uncertainties Δx_i in the inputs:



Comment. In the above argument, we assumed that the relation f provides the *exact* relation between the variables x_1, \dots, x_n and the desired value y . In this case, in the ideal case when we plug in the actual (unknown) values of x_i into the algorithm f , we get the exact value $y = f(x_1, \dots, x_n)$ of y .

In many real-life situations, the relation f between x_i and y is only *approximately* known. In this case, even if we know the exact values of x_i , substituting these values into the approximate function f will not provide us with the exact value of y . In such situations, there is even more uncertainty in y :

- First, there is an uncertainty in y caused by the uncertainty in the inputs.
- Second, there is a *model uncertainty* caused by the fact that the known algorithm f provides only an approximate description of the dependence between the inputs and the output.

Interval computations enable us to estimate the uncertainty in y caused by the uncertainty of the inputs. If there is also a model uncertainty, it has to be estimated separately and added to the uncertainty produced by the interval computations techniques.

In many practical problems, it is important to estimate the inaccuracy of the results of data processing. In many practical applications, it is important to know not only the desired estimate for the quantity y , but also how accurate this estimate is.

For example, in geophysical applications, it is not enough to know that the amount of oil in a given oil field is about 100 million tons. It is important to know how accurate this estimate is.

If the amount is 100 ± 10 , this means that the estimates are good enough, and we should start exploring this oil field. On the other hand, if it is 100 ± 200 , this means that it is quite possible that the actual value of the desired quantity y is 0; i.e., there is no oil at all. In this case, it may be prudent to perform additional measurements before we invest a lot of money into drilling oil wells.

The situation becomes even more critical in medical emergencies: it is not enough to have an estimate of the blood pressure or the body temperature to make a decision (e.g., whether to perform a surgery); it is important that even with the measurement uncertainty, we are sure about the diagnosis – and if we are not, maybe it is desirable to perform more accurate measurements.

Problems of the second class (prediction related): uncertainty in initial values leads to uncertainty of predicted values. In the prediction problems, we start with the estimates \tilde{y}_i of the current values of the known quantities; we then apply the prediction algorithm g and produce the prediction $\tilde{z} = g(\tilde{y}_1, \dots, \tilde{y}_m)$ for the desired future value z .

We have already mentioned that, in general, the estimates \tilde{y}_i of the current values of the quantities y_i are different from the (unknown) actual values y_i of these quantities. Therefore, even if the prediction algorithm is absolutely exact, i.e., if the future value of z is equal to $g(y_1, \dots, y_m)$, the prediction result \tilde{z} will be different from the actual future value z .

Comment. In many practical situations, the prediction algorithm is only approximately known, so in general (just as for the problems from the first class), there is also a model uncertainty – an additional component of uncertainty.

1.4 From Probabilistic to Interval Uncertainty: Case of Indirect Measurements

Let us start with the uncertainty of learning the values of the desired quantities. In the previous section, we have shown that the uncertainties in the results of direct measurements and/or direct expert estimations lead to an uncertainty in our estimates of the current values of the physical quantities. These uncertainties, in turn, lead to an uncertainty in the predicted values.

We are interested in the uncertainties occurring in problems of both classes: learning the current values and predicting the future values. Since the uncertainty in the future values comes from the uncertainty in the current values, it is reasonable to start with analyzing the uncertainty of the learned values.

Let us start with indirect measurements. In the situation of learning the current values of the physical quantities, there are two possible situations:

- when the (estimates for the) values of the auxiliary quantities x_i come from direct measurements and
- when these estimates come from the expert estimation.

(Of course, it is also possible that some estimates come from measurement and some from expert estimation.)

There is a lot of experience of handling measurement uncertainty, so we will start our analysis with measurement uncertainty. After that, we will explain how similar techniques can handle expert uncertainty.

Case of direct measurements: what can we know about Δx_i . To estimate the uncertainty Δy caused by the measurement uncertainties Δx_i , we need to have some information about these original uncertainties Δx_i .

The whole idea of uncertainty is that we do not know the exact value of x_i . (Hence, we do not know the exact value of Δx_i .) In other words, there are several possible values of Δx_i . So, the first thing we would like to know is what is the *set* of possible values of Δx_i .

We may also know that some of these possible values are more frequent than the others. In other words, we may also have some information about the *probabilities* of different possible values Δx_i .

We need to go from theoretical possibility to practical situations. Up to now, we have analyzed the situation on a purely theoretical level: what kind of information can we have *in principle*.

From the viewpoint of practical applications, it is desirable to analyze what information we actually have.

First piece of information: upper bound on the measurement error. The manufacturers of a measuring device usually provide us with an upper bound Δ_i for the (absolute value of) possible measurement errors, i.e., with the bound Δ_i for which we are guaranteed that $|\Delta x_i| \leq \Delta_i$.

The need for such a bound comes from the very nature of a measurement process. Indeed, if no such bound is provided, this means that the actual value x_i can be as different from the ‘measurement result’ \tilde{x}_i as possible. Such a value \tilde{x}_i is not a measurement, but a wild guess.

Enter intervals. Since the (absolute value of the) measurement error $\Delta x_i = \tilde{x}_i - x_i$ is bounded by the given bound Δ_i , we can therefore guarantee that the actual (unknown) value of the desired quantity belongs to the interval

$$\mathbf{x}_i \stackrel{\text{def}}{=} [\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i].$$

Example. For example, if the measured value of a quantity is $\tilde{x}_i = 1.0$ and the upper bound Δ_i on the measurement error is 0.1, this means that the (unknown) actual value of the measured quantity can be anywhere between $1 - 0.1 = 0.9$ and $1 + 0.1 = 1.1$; i.e., it can take any value from the interval $[0.9, 1.1]$.

Often, we also know probabilities. In many practical situations, we not only know the interval $[-\Delta_i, \Delta_i]$ of possible values of the measurement error; we also know the probability of different values Δx_i within this interval [2].

In most practical applications, it is assumed that the corresponding measurement errors are normally distributed with 0 means and known standard deviation.

Numerous engineering techniques are known (and widely used) for processing this uncertainty (see, e.g., [2]).

How we can determine these probabilities. In practice, we can determine the desired probabilities of different values of Δx_i by comparing

- the result \tilde{x}_i of measuring a certain quantity with this instrument and
- the result $\tilde{x}_{i \text{ st}}$ of measuring the same quantity by a standard (much more accurate) measuring instrument.

Since the standard measuring instrument is much more accurate than the one we use, i.e., $|\tilde{x}_{i\text{st}} - x_i| \ll |\tilde{x}_i - x_i|$, we can assume that $\tilde{x}_{i\text{st}} = x_i$, and thus that the difference $\tilde{x}_i - \tilde{x}_{i\text{st}}$ between these two measurement results is practically equal to the measurement error $\Delta x_i = \tilde{x}_i - x_i$.

Thus, the empirical distribution of the difference $\tilde{x}_i - \tilde{x}_{i\text{st}}$ is close to the desired probability distribution of the measurement error.

In some important practical situations, we cannot determine these probabilities. In many practical cases, by using standard measuring instruments, we can determine the probabilities of different values of Δx_i . There are two cases, however, when this determination is not done:

- First is the case of *cutting-edge* measurements, e.g., measurements in fundamental science. When the Hubble telescope detects the light from a distant galaxy, there is no ‘standard’ (much more accurate) telescope floating nearby that we can use to calibrate the Hubble: the Hubble telescope is the best we have.
- The second case is the case of real *industrial* applications (such as measurements on the shop floor). In this case, in principle, every sensor can be thoroughly calibrated, but sensor calibration is so costly – usually costing several orders of magnitude more than the sensor itself – that manufacturers rarely do it (only if it is absolutely necessary).

In both cases, we have no information about the probabilities of Δx_i ; the only information we have is the upper bound on the measurement error.

Case of interval uncertainty. In this case, after performing a measurement and getting a measurement result \tilde{x}_i , the only information that we have about the actual value x_i of the measured quantity is that it belongs to the interval $\mathbf{x}_i = [\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$.

In other words, we do not know the actual value x_i of the i th quantity. Instead, we know the *granule* $[\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$ that contains x_i .

Resulting computational problem. In this situation, for each i , we know the interval \mathbf{x}_i of possible values of x_i , and we need to find the range

$$\mathbf{y} \stackrel{\text{def}}{=} \{f(x_1, \dots, x_n) : x_1 \in \mathbf{x}_1, \dots, x_n \in \mathbf{x}_n\}$$

of the given function $f(x_1, \dots, x_n)$ over all possible tuples $x = (x_1, \dots, x_n)$ with $x_i \in \mathbf{x}_i$.

The desired range is usually also an interval. Since the function $f(x_1, \dots, x_n)$ is usually continuous, this range is also an interval; i.e., $\mathbf{y} = [\underline{y}, \bar{y}]$ for some \underline{y} and \bar{y} . So, to find this range, it is sufficient to find the endpoints \underline{y} and \bar{y} of this interval.

From traditional (numerical) computations to interval computations. In traditional data processing, we know the estimates \tilde{x}_i of the input values, and we use these estimates to compute the estimate \tilde{y} for the desired quantity y . The corresponding algorithm is a particular case of *computations* (which often require a large amount of computing power).

When we take uncertainty in the account, we have a similar problem, in which

- as inputs, instead of the *numerical* estimates \tilde{x}_i for x_i , we have *intervals* of possible values of x_i , and
- as an output, instead of a numerical estimate \tilde{y} for y , we want to compute the interval $[\underline{y}, \bar{y}]$ of possible values of y .

The corresponding computations are therefore called *interval computations*.

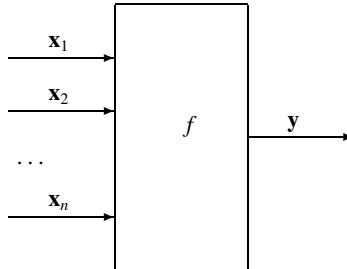
Let us formulate the corresponding problem of interval computations in precise terms.

The main problem of interval computations: a precise description. We are given

- an integer n ,
- n intervals $\mathbf{x}_1 = [\underline{x}_1, \bar{x}_1], \dots, \mathbf{x}_n = [\underline{x}_n, \bar{x}_n]$, and
- an algorithm $f(x_1, \dots, x_n)$ which transforms n real numbers into a real number $y = f(x_1, \dots, x_n)$.

We need to compute the endpoints \underline{y} and \bar{y} of the interval

$$\mathbf{y} = [\underline{y}, \bar{y}] = \{f(x_1, \dots, x_n) : x_1 \in [\underline{x}_1, \bar{x}_1], \dots, [x_n, \bar{x}_n]\}.$$



Interval computations are also important for the second class of problems: predicting future. In the prediction problem, we start with the known information about the current values y_1, \dots, y_m of the physical quantities. On the basis of this information, we would like to derive the information about possible future value $z = g(y_1, \dots, y_m)$ of each quantity of interest z .

We have already mentioned that in many practically important situations, we can only determine the intervals $[\underline{y}_i, \bar{y}_i]$ of possible values of y_i . In this case, the only information that we can deduce about z is that z belongs to the range

$$\mathbf{z} = \{g(y_1, \dots, y_m) : y_1 \in [\underline{y}_1, \bar{y}_1], \dots, y_m \in [\underline{y}_m, \bar{y}_m]\}.$$

The problem of computing this range is also the problem of interval computations:

- We know intervals of possible values of the input.
- We know the algorithm that transforms the input into the output.
- We want to find the interval of possible values of the output.

Thus, interval computations are also important for the prediction problem.

1.5 Case of Expert Uncertainty

How can we describe expert uncertainty. So far, we have analyzed measurement uncertainty. As we have mentioned earlier, expert estimates also come with uncertainty. How can we estimate and process this uncertainty?

Probabilistic approach: its possibility and limitations. For a measuring instrument, we know how to estimate the probability distribution of the measurement error:

- Ideally, we should compare the measurement results with the actual values of the measured quantity. The resulting differences form a sample from the actual distribution of measurement error. On the basis of this sample, we can determine the probability distribution of the measurement error.
- In practice, since we cannot determine the exact actual value of the quantity, we use an approximate value obtained by using a more accurate measuring instrument. On the basis of the sample of the corresponding differences, we can still determine the probability distribution of the measurement error.

In principle, we can do the same for expert estimates. Namely, to estimate the quality of expert estimates, we can consider the cases when the quantity estimates by an expert were consequently measured. Usually, measurements are much more accurate than expert estimates; i.e., $|\tilde{x}_{\text{meas}} - x| \ll |\tilde{x} - x|$, where x is the

(unknown) value of the estimated quantity, \tilde{x} is the expert estimate for this quantity, and \tilde{x}_{meas} is the result of the consequent measurement of this same quantity. In comparison with expert estimates, we can therefore consider measurement results as approximately equal to the actual values of the quantity: $\tilde{x}_{\text{meas}} - \tilde{x} \approx \tilde{x} - x$. Thus, by considering the differences $\tilde{x}_{\text{meas}} - \tilde{x}$ as a sample from the unknown probability distribution, we can determine the probability distribution of the expert estimation error.

If we have such a probability distribution, then we can use traditional well-developed statistical methods to process expert estimates – the same way we can process measurement results for which we know the distribution of measurement errors.

To determine a probability distribution from the empirical data, we need a large sample: the larger the sample, the more accurate the results.

- A measuring instrument takes a small portion of a second to perform a measurement. Thus, with a measuring instrument, we can easily perform dozens, hundreds, and even thousands of measurements. So, we can have samples which are large enough to determine the corresponding probability distribution with reasonable accuracy.
- On the other hand, for an expert, a single estimate may require a lot of analysis. As a result, for each expert, there are usually few estimates, and it is often not possible to determine the distribution from these estimates.

Experts can produce interval bounds. A measuring instrument usually simply produces a number; it cannot be easily modified to also produce an information about the measurement uncertainty, such as the upper bound on the measurement error.

In contrast, an expert is usually able not only to supply us with an estimate, but also to provide us with an accuracy of this estimate. For example, an expert can estimate the age of a person as $\tilde{x} = 30$ and indicate that this is 30 plus minus $\Delta = 5$. In such a situation, what the expert is actually saying is that the actual (unknown) value of the estimated quantity should be in the interval $[\tilde{x} - \Delta, \tilde{x} + \Delta]$.

Interval computations are needed to handle interval uncertainty in expert estimates. Let us now consider a typical situation of data processing. We are interested in some quantity y which is difficult to estimate directly. To estimate y , we ask experts to estimate the values of the auxiliary quantities x_1, \dots, x_n which are related to y by a known dependence $y = f(x_1, \dots, x_n)$.

On the basis of the expert estimates \tilde{x}_i and the expert estimates Δ_i of their inaccuracy, we conclude that the actual (unknown) value of the each quantity x_i belongs to the interval $\mathbf{x}_i \stackrel{\text{def}}{=} [\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$. Thus, we can conclude that the actual value of $y = f(x_1, \dots, x_n)$ belongs to the interval range

$$[\underline{y}, \bar{y}] \stackrel{\text{def}}{=} \{f(x_1, \dots, x_n) : x_1 \in \mathbf{x}_1, \dots, x_n \in \mathbf{x}_n\}.$$

The problem of computing this range is exactly the problem of interval computations.

From interval to fuzzy uncertainty. Usually, experts can provide guaranteed bounds Δ_i on the inaccuracy of their estimates. Often, however, in addition to these (rather wide) bounds, experts can also produce narrower bounds – which are, however, true only with a certain degree of certainty.

For example, after estimating the age as 30,

- in addition to saying that an estimation inaccuracy is always ≤ 5 (with 100% certainty),
- an expert can also say that with 90% certainty, this inaccuracy is ≤ 4 , and
- with 70% certainty, this inaccuracy is ≤ 2 .

Thus, instead of a single interval $[30 - 5, 30 + 5] = [25, 35]$ that is guaranteed to contain the (unknown) age with certainty 100%, the expert also produces a narrower interval $[30 - 4, 30 + 4] = [26, 34]$ which contains this age with 90% certainty, and an even narrower interval $[30 - 2, 30 + 2] = [28, 32]$ which contains the age with 70% certainty. So, we have three intervals which are *nested* in the sense that every interval corresponding to a smaller degree of certainty is contained in the interval corresponding to the larger degree of certainty: $[28, 32] \subseteq [26, 34] \subseteq [25, 35]$.

In general, instead of a single interval, we have a nested family of intervals corresponding to different degrees of certainty. Such a nested family of intervals can be viewed as a *fuzzy number* [3, 4]: for every value x , we can define the degree $\mu(x)$ to which x is possible as 1 minus the largest degree of certainty α for which x belongs to the α -interval.

Interval computations are needed to process fuzzy data. For expert estimates, for each input i , we may have different intervals $\mathbf{x}_i(\alpha)$ corresponding to different degrees of certainty α . Our objective is then to produce the corresponding intervals for $y = f(x_1, \dots, x_n)$.

For $\alpha = 1$, i.e., for intervals in which the experts are 100% confident, it is natural to take $\mathbf{y}(1) = f(\mathbf{x}_1(1), \dots, \mathbf{x}_n(1))$. Similarly, for each α , if we want to consider beliefs at this level α , then we can combine the corresponding intervals $\mathbf{x}_i(\alpha)$ into the desired interval

$$\mathbf{y}(\alpha) \text{ for } y: \mathbf{y}(\alpha) = f(\mathbf{x}_1(\alpha), \dots, \mathbf{x}_n(\alpha)).$$

It turns out that the resulting fuzzy number is exactly what we would get if we simply apply Zadeh's extension principle to the fuzzy numbers corresponding to x_i [3–5].

So, in processing fuzzy expert opinions, we also need interval computations.

1.6 Interval Computations Are Sometimes Easy but In General, They Are Computationally Difficult (NP-Hard)

Interval computations are needed in practice: a reminder. In the previous sections, we have explained why interval computations are needed in many practical problems. In other words, in many practical situations, we know n intervals $\mathbf{x}_1, \dots, \mathbf{x}_n$, know an algorithm $f(x_1, \dots, x_n)$, and need to find the range of the function f on these intervals:

$$[\underline{y}, \bar{y}] = \{f(x_1, \dots, x_n) : x_1 \in \mathbf{x}_1, \dots, x_n \in \mathbf{x}_n\}.$$

Let us first analyze the computational complexity of this problem. Before we start explaining how to solve this problem, let us make a useful detour.

Until the 1930s, researchers believed that every mathematical problem can be solved. Under this belief, once we have a mathematical problem of practical importance, we should try to solve it in its entire generality.

Starting with the famous Gödel's result, it is well known that some mathematical problems cannot be solved in the most general case. For such problems, attempts to solve them in their most general form would be a futile waste of time. At best, we can solve some important class of such problems or get an approximate solution. To avoid this waste of efforts, before we start solving a difficult problem, it is desirable to first analyze whether this problem can be solved in its utmost generality.

This strategy was further clarified in the 1970s, when it turned out, crudely speaking, that some problems cannot be *efficiently* solved; such difficult problems are called *NP-hard* (see [1, 6, 7] for detailed description). If a problem is NP-hard, then it is hopeless to search for a general efficient solution; we must look for efficient solutions to subclasses of this problem and/or approximate solutions.

Comment. Strictly speaking, NP-hardness does not necessarily mean that the problem is computationally difficult: this is true only under a hypothesis $\text{NP} \neq P$, which is widely believed but not proved yet. (It is probably the most well known open problem in theoretical computer science.)

Interval computations are sometimes easy: case of monotonicity. In some cases, it is easy to estimate the desired range. For example, the arithmetic average

$$E = \frac{x_1 + \dots + x_n}{n}$$

is a monotonically increasing function of each of its n variables x_1, \dots, x_n . So,

- the smallest possible value \underline{E} of the average E is attained when each value x_i is the smallest possible ($x_i = \underline{x}_i$), and
- the largest possible value \overline{E} of the average E is attained when $x_i = \bar{x}_i$ for all i .

In other words, the range \mathbf{E} of E is equal to $[E(\underline{x}_1, \dots, \underline{x}_n), E(\bar{x}_1, \dots, \bar{x}_n)]$, where

$$\underline{E} = \frac{1}{n} \cdot (\underline{x}_1 + \dots + \underline{x}_n)$$

and

$$\overline{E} = \frac{1}{n} \cdot (\bar{x}_1 + \dots + \bar{x}_n).$$

In general, if $f(x_1, \dots, x_n)$ is a monotonically increasing function of each of its n variables, then

- The smallest possible value \underline{y} of the function f over given intervals $[\underline{x}_i, \bar{x}_i]$ is attained when all its inputs x_i take the smallest possible values $x_i = \underline{x}_i$. In this case, $\underline{y} = f(\underline{x}_1, \dots, \underline{x}_n)$.
- The largest possible value \overline{y} of the function f over given intervals $[\underline{x}_i, \bar{x}_i]$ is attained when all its inputs x_i take the largest possible values $x_i = \bar{x}_i$. In this case, $\overline{y} = f(\bar{x}_1, \dots, \bar{x}_n)$.

Thus, we have an explicit formula for the desired range:

$$[\underline{y}, \overline{y}] = [f(\underline{x}_1, \dots, \underline{x}_n), f(\bar{x}_1, \dots, \bar{x}_n)].$$

A similar formula can be written down if the function $f(x_1, \dots, x_n)$ is increasing with respect to some of its inputs and decreasing with respect to some others. In this case, to compute \overline{y} , we must take

- $x_i = \bar{x}_i$ for all the variables x_i relative to which f is increasing, and
- $x_j = \underline{x}_j$ for all the variables x_j relative to which f is decreasing.

Similarly, to compute \underline{y} , we must take

- $x_i = \underline{x}_i$ for all the variables x_i relative to which f is increasing, and
- $x_j = \bar{x}_j$ for all the variables x_j relative to which f is decreasing.

Case of linear functions $f(x_1, \dots, x_n)$. In the previous section, we showed how to compute the range of a function which is monotonic in each of its variables – and it can be increasing relative to some of them and decreasing relative to some others.

An example of such a function is a general linear function $f(x_1, \dots, x_n) = c_0 + \sum_{i=1}^n c_i \cdot x_i$. Substituting $x_i = \tilde{x}_i - \Delta x_i$ into this expression, we conclude that

$$y = f(x_1, \dots, x_n) = c_0 + \sum_{i=1}^n c_i \cdot (\tilde{x}_i - \Delta x_i) = c_0 + \sum_{i=1}^n c_i \cdot \tilde{x}_i - \sum_{i=1}^n c_i \cdot \Delta x_i.$$

By definition, $\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n) = c_0 + \sum_{i=1}^n c_i \cdot \tilde{x}_i$, so we have

$$\Delta y = \tilde{y} - y = \sum_{i=1}^n c_i \cdot \Delta x_i.$$

The dependence of Δy on Δx_i is linear: it is increasing relative to x_i if $c_i \geq 0$ and decreasing if $c_i < 0$. So, to find the largest possible value Δ of Δy , we must take

- the largest possible value $\Delta x_i = \Delta_i$ when $c_i \geq 0$, and
- the smallest possible value $\Delta x_i = -\Delta_i$ when $c_i < 0$.

In both cases, the corresponding term in the sum has the form $|c_i| \cdot \Delta_i$, so we can conclude that

$$\Delta = \sum_{i=1}^n |c_i| \cdot \Delta_i.$$

Similarly, the smallest possible value of Δy is equal to $-\Delta$. Thus, the range of possible values of y is equal to $[\underline{y}, \bar{y}] = [\tilde{y} - \Delta, \tilde{y} + \Delta]$.

Interval computations are, in general, computationally difficult. We have shown that for linear functions, we can easily compute the interval range.

Linear functions often occur in practice, because an arbitrary function can be usually expanded in Taylor series and then we can keep only a few first terms to get a good description of the actual dependence. If we keep only linear terms, then we get a linear approximation to the original dependence.

If the accuracy of this linear approximation is not sufficient, then it is natural to also consider quadratic terms. A natural question is ‘is the corresponding interval computations problem still feasible?’

Alas, it turns out that for quadratic functions, interval computations problem is, in general, NP-hard; this was first proved in [8]. Moreover, it turns out that it is NP-hard not just for some rarely used exotic quadratic functions: it is known that the problem of computing the exact range $V = [\underline{V}, \bar{V}]$ for the variance

$$V = \frac{1}{n} \cdot \sum_{i=1}^n (x_i - E)^2 = \frac{1}{n} \cdot \sum_{i=1}^n x_i^2 - \left(\frac{1}{n} \cdot \sum_{i=1}^n x_i \right)^2$$

over interval data $x_i \in [\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$ is, in general, NP-hard (see, e.g., [9, 10]). To be more precise, there is a polynomial-time algorithm for computing \underline{V} , but computing \bar{V} is, in general, NP-hard.

Historical comment. NP-hardness of interval computations was first proved in [11, 12]. A general overview of computational complexity of different problems of data processing and interval computations is given in [1].

1.7 Maximum Entropy and Linearization: Useful Techniques for Solving Many Practical Cases of Interval Computations Problem, Their Advantages and Limitations

In many practical situations, an approximate estimate is sufficient. The NP-hardness result states that computing the *exact* range $[\underline{y}, \bar{y}]$, i.e., in other words, computing the *exact* values of the endpoints \underline{y} and \bar{y} , is NP-hard.

In most practical problems, however, it is not necessary to produce the exact values of the range; good approximate values will be quite sufficient.

Computing the range with guaranteed accuracy is still NP-hard. Thus, we arrive at the following natural question. Suppose that we fix an accuracy ε , and we consider the problem of computing \underline{y} and \bar{y} with this accuracy, i.e., the problem of computing the values \bar{Y} and \underline{Y} for which $|\bar{Y} - \bar{y}| \leq \varepsilon$ and $|\underline{Y} - \underline{y}| \leq \varepsilon$.

In this case, we can guarantee that $\underline{Y} - \varepsilon \leq \underline{y} \leq \bar{Y} + \varepsilon$ and $\bar{Y} - \varepsilon \leq \bar{y} \leq \underline{Y} + \varepsilon$. So, if we succeed in computing the estimate \underline{Y} and \bar{Y} , then we do not have the exact range, but we have an ε -approximation for the (unknown) desired range y : namely, we know that

$$[\underline{Y} + \varepsilon, \bar{Y} - \varepsilon] \subseteq y \subseteq [\underline{Y} - \varepsilon, \bar{Y} + \varepsilon].$$

Is the problem of computing such values \underline{Y} and \bar{Y} computationally simpler? Alas, it turns out that this new problem is still NP-hard (see, e.g., [1]).

In some practical problems, it is OK to have estimates which are not guaranteed. The difficulty of solving the general problem of interval computations comes from the fact that we are looking for *guaranteed* bounds for \underline{y} and \bar{y} .

In some practical problems, we are not 100% sure that our algorithm $f(x_1, \dots, x_n)$ is absolutely correct. This happens, e.g., in prediction problems, where the dynamic equations used for prediction are only approximately known anyway.

In such situations, it is OK to have estimates sometimes deviating from the desired range.

Possible approaches to this problem. In order to describe possible approaches to this problem, let us first recall what properties of our problem make it computationally complex. By relaxing these properties, we will be able to come up with computationally efficient algorithms.

We have mentioned that in some practical situations, we know the probability distributions of the estimation errors Δx_i . In such situations, the problem of estimating the effect of these approximation errors Δx_i on the result of data processing is computationally easy. Namely, we can use Monte-Carlo simulations (see, e.g., [13]), when for several iterations $k = 1, \dots, N$, we do the following:

- Simulate the inputs $\Delta x_i^{(k)}$ according to the known probability distributions.
- Substitute the resulting simulated values $x_i^{(k)} = \tilde{x}_i - \Delta x_i^{(k)}$ into the algorithm f , producing $y^{(k)} = f(x_1^{(k)}, \dots, x_n^{(k)})$.
- And then use the sample of the differences $\Delta y^{(k)} = \tilde{y} - y^{(k)}$ to get the probability distribution of Δy .

Thus, the first difficulty of interval computations comes from the fact that we do not know the probability distribution. However, the mere fact that we do not know this distribution does not necessarily make the problem computationally complex. For example, even when we restrict ourselves to interval uncertainty, for linear functions f , we still have a feasible algorithm for computing the range. Thus, the complexity of the general interval computations problem is caused by the following two properties of this general problem:

- first, that we do not know the probability distribution for the inputs Δx_i , and
- second, that the function $f(x_1, \dots, x_n)$ is non-linear.

To be able to perform efficient computations, we must relax one of these properties. Thus, we arrive at two possible ways to solve this problem:

- First, we can select one of the possible probability distributions.
- Second, we can approximate the original function f by a linear one.

Let us describe these two ideas in more detail.

First idea: selecting a probability distribution. As we have mentioned, in many cases, we know the probability distribution for approximation errors Δx_i . Interval uncertainty corresponds to the case when we have only a partial information about this probability distribution: namely, the only thing we know about this distribution is that it is located (with probability 1) somewhere on the interval $[-\Delta_i, \Delta_i]$.

This distribution could be uniform on this interval, could be a truncated Gaussian distribution, and could be a 1-point degenerate distribution, in which the value Δx_i is equal to one fixed value from this interval with probability 1.

Situations in which we have partial information about the probability distributions are common in statistics. In such situations, we have several different probability distributions which are all consistent with the given knowledge. One way to handle these situations is to select one of these distributions, the one which is, in some sense, the most reasonable to select.

Simplest case: Laplace's principle of indifference. The approach started with the early nineteenth-century work of the famous mathematician Pierre Simon Laplace, who analyzed the simplest

of such situations, when we have finitely many (n) alternatives and have no information about their probabilities. In this simple situation, the original situation is invariant with respect to arbitrary permutations of the original alternatives. So, it is reasonable to select the probabilities which reflect this symmetry – i.e., equal probabilities $p_1 = \dots = p_n$. Since the total probability $\sum_{i=1}^n p_i$ must be equal to 1, we thus conclude that $p_1 = \dots = p_n = 1/n$. This idea is called *Laplace's principle of indifference*.

General case: maximum entropy approach. Laplace's simple idea can be naturally applied to the more general case, when we have partial information about the probabilities, i.e., when there are several possible distributions which are consistent with our knowledge. In this case, it is reasonable to view these distributions as possible alternatives. So, we discretize the variables (to make sure that the overall number of alternatives is finite) and then consider all possible distributions as equally probable. As the discretization constant tends to 0, we should get a distribution of the class of all (non-discretized) distributions.

It turns out that in the limit, only one such distribution has probability 1: namely, the distribution which has the largest possible value of the entropy $S \stackrel{\text{def}}{=} - \int \rho(x) \cdot \ln(\rho(x)) dx$. (Here $\rho(x)$ denotes the probability density.) For details on this maximum entropy approach and its relation to interval uncertainty and Laplace's principle of indifference, see, e.g., [14–16].

Maximum entropy method for the case of interval uncertainty. One can easily check that for a single variable x_1 , among all distributions located on a given interval, the entropy is the largest when this distribution is *uniform* on this interval. In the case of several variables, we can similarly conclude that the distribution with the largest value of the entropy is the one which is uniformly distributed in the corresponding box $\mathbf{x}_1 \times \dots \times \mathbf{x}_n$, i.e., a distribution in which

- each variable Δx_i is uniformly distributed on the corresponding interval $[-\Delta_i, \Delta_i]$, and
- variables corresponding to different inputs are statistically independent.

This is indeed one of the main ways how interval uncertainty is treated in engineering practice: if we only know that the value of some variable x_i is in the interval $[\underline{x}_i, \bar{x}_i]$ and we have no information about the probabilities, then we assume that the variable x_i is uniformly distributed on this interval.

Limitations of the maximum entropy approach. To explain the limitations of this engineering approach, let us consider the simplest possible algorithm $y = f(x_1, \dots, x_n) = x_1 + \dots + x_n$. For simplicity, let us assume that the measured values of all n quantities are 0s, $\tilde{x}_1 = \dots = \tilde{x}_n = 0$, and that all n measurements have the same error bound Δ_x ; i.e., $\Delta_1 = \dots = \Delta_n = \Delta_x$.

In this case, $\Delta y = \Delta x_1 + \dots + \Delta x_n$. Each of n component measurement errors can take any value from $-\Delta_x$ to Δ_x , so the largest possible value of Δy is attained when all of the component errors attain the largest possible value $\Delta x_i = \Delta_x$. In this case, the largest possible value Δ of Δy is equal to $\Delta = n \cdot \Delta_x$.

Let us see what the maximum entropy approach will predict in this case. According to this approach, we assume that Δx_i are independent random variables, each of which is uniformly distributed on the interval $[-\Delta_x, \Delta_x]$. According to the central limit theorem [17, 18], when $n \rightarrow \infty$, the distribution of the sum of n independent identically distributed bounded random variables tends to Gaussian. This means that for large values n , the distribution of Δy is approximately normal.

Normal distribution is uniquely determined by its mean and variance. When we add several independent variables, their means and variances add up. For each uniform distribution Δx_i on the interval $[-\Delta_x, \Delta_x]$ of width $2\Delta_x$, the probability density is equal to $\rho(x) = (1/2\Delta_x)$, so the mean is 0 and the variance is

$$V = \int_{-\Delta_x}^{\Delta_x} x^2 \cdot \rho(x) dx = \frac{1}{2\Delta_x} \cdot \int_{-\Delta_x}^{\Delta_x} x^2 dx = \frac{1}{2\Delta_x} \cdot \frac{1}{3} \cdot x^3 \Big|_{-\Delta_x}^{\Delta_x} = \frac{1}{3} \cdot \Delta_x^2.$$

Thus, for the sum Δy of n such variables, the mean is 0 and the variance is equal to $(n/3) \cdot \Delta_x^2$. Thus, the standard deviation is equal to $\sigma = \sqrt{V} = \Delta_x \cdot \sqrt{n}/\sqrt{3}$.

It is known that in a normal distribution, with probability close to 1, all the values are located within the $k \cdot \sigma$ vicinity of the mean: for $k = 3$, it is true with probability 99.9%; for $k = 6$, it is true with

probability $10^{-6}\%$; and so on. So, practically with certainty, Δy is located within an interval $k \cdot \sigma$ which grows with n as \sqrt{n} .

For large n , we have $k \cdot \Delta_x \cdot \sqrt{n}/\sqrt{3} \ll \Delta_x \cdot n$, so we get a serious underestimation of the resulting measurement error. This example shows that estimates obtained by selecting a single distribution can be very misleading.

Linearization: main idea. As we have mentioned earlier, another way to handle the complexity of the general interval computations problem is to approximate the original expression $y = f(x_1, \dots, x_n) = f(\tilde{x}_1 - \Delta x_1, \dots, \tilde{x}_n - \Delta x_n)$ by linear terms in its Taylor expansion:

$$y \approx f(\tilde{x}_1, \dots, \tilde{x}_n) - \sum_{i=1}^n \frac{\partial f}{\partial x_i} \cdot \Delta x_i,$$

where the partial derivatives are computed at the midpoint $\tilde{x} = (\tilde{x}_1, \dots, \tilde{x}_n)$. Since $f(\tilde{x}_1, \dots, \tilde{x}_n) = \tilde{y}$, we conclude that $\Delta y = \tilde{y} - y = \sum_{i=1}^n c_i \cdot \Delta x_i$, where $c_i = \partial f / \partial x_i$.

We already know how to compute the interval range for a linear function, and the resulting formula is $\Delta = \sum_{i=1}^n |c_i| \cdot \Delta_i$. Thus, to compute Δ , it is sufficient to know the partial derivatives c_i .

Linearization: how to compute. A natural way to compute partial derivatives comes directly from their definition. By definition, a partial derivative is defined as a limit

$$\frac{\partial f}{\partial x_i} = \lim_{h \rightarrow 0} \frac{f(\tilde{x}_1, \dots, \tilde{x}_{i-1}, \tilde{x}_i + h, \tilde{x}_{i+1}, \dots, \tilde{x}_n) - f(\tilde{x}_1, \dots, \tilde{x}_n)}{h}.$$

In turn, a limit, by its definition, means that when the value of h is small, the corresponding ratio is very close to the partial derivative. Thus, we can estimate the partial derivative as the ratio

$$c_i = \frac{\partial f}{\partial x_i} \approx \frac{f(\tilde{x}_1, \dots, \tilde{x}_{i-1}, \tilde{x}_i + h, \tilde{x}_{i+1}, \dots, \tilde{x}_n) - f(\tilde{x}_1, \dots, \tilde{x}_n)}{h}$$

for some small value h .

After we have computed n such ratios, we can then compute the desired bound Δ on $|\Delta y|$ as $\Delta = \sum_{i=1}^n |c_i| \cdot \Delta_i$.

Linearization: how to compute faster. The above algorithm requires that we call the data processing algorithm $n + 1$ times: first to compute the value $\tilde{y} = f(\tilde{x}_1, \dots, \tilde{x}_n)$ and then n more times to compute the values

$$f(\tilde{x}_1, \dots, \tilde{x}_{i-1}, \tilde{x}_i + h, \tilde{x}_{i+1}, \dots, \tilde{x}_n),$$

and thus the corresponding partial derivatives.

In many practical situations, the data processing algorithms are time consuming, and we process large amounts of data, with the number n of data points in thousands. In this case, the use of the above linearization algorithm would require thousands time longer than data processing itself – which itself is already time consuming. Is it possible to estimate Δ faster?

The answer is ‘yes,’ it is possible to have an algorithm which estimates Δ by using only a constant number of calls to the data processing algorithm f (for details, see, e.g., [19, 20]).

In some situations, we need a guaranteed enclosure. In many application areas, it is sufficient to have an approximate estimate of y . However, in some applications, it is important to guarantee that the (unknown) actual value y of a certain quantity does not exceed a certain threshold y_0 . The only way to guarantee this is to have an interval $\mathbf{Y} = [\underline{Y}, \bar{Y}]$ which is guaranteed to contain y (i.e., for which $\mathbf{y} \subseteq \mathbf{Y}$) and for which $\bar{Y} \leq y_0$.

For example, in nuclear engineering, we must make sure that the temperatures and the neutron flows do not exceed the critical values; when planning a spaceflight, we want to guarantee that the spaceship lands on the planet and does not fly past it.

The interval \mathbf{Y} which is guaranteed to contain the actual range \mathbf{y} is usually called an *enclosure* for this range. So, in such situations, we need to compute either the original range or at least an enclosure for this range. Computing such an enclosure is also one of the main tasks of interval computations.

1.8 Interval Computations: A Brief Historic Overview

Before we start describing the main interval computations techniques, let us briefly overview the history of interval computations.

Prehistory of interval computations: interval computations as a part of numerical mathematics. The notion of interval computations is reasonably recent: it dates from the 1950s. But the main problem is known since Archimedes who used guaranteed two-sided bounds to compute π (see, e.g., [21]).

Since then, many useful guaranteed bounds have been developed for different numerical methods. There have also been several general descriptions of such bounds, often formulated in terms similar to what we described above. For example, in the early twentieth-century, the concept of a function having values which are bounded within limits was discussed by W.H. Young in [22]. The concept of operations with a set of multivalued numbers was introduced by R.C. Young, who developed a formal algebra of multivalued numbers [23]. The special case of closed intervals was further developed by P.S. Dwyer in [24].

Limitations of the traditional numerical mathematics approach. The main limitation of the traditional numerical mathematics approach to error estimation was that often no clear distinction was made between approximate (non-guaranteed) and guaranteed (=interval) error bounds.

For example, for iterative methods, many papers on numerical mathematics consider the rate of convergence as an appropriate measure of approximation error. Clearly, if we know that the error decreases as $O(1/n)$ or as $O(a^{-n})$, we gain some information about the corresponding algorithms – and we also gain a knowledge that for large n , the second method is more accurate. However, in real life, we make a fixed number n of iterations. If the only information we have about the approximation error is the above asymptotics, then we still have no idea how close the result of n th iteration is to the actual (desired) value.

It is therefore important to emphasize the need for guaranteed methods and to develop techniques for producing *guaranteed* estimates. Such guaranteed estimates are what interval computations are about.

Origins of interval computations. Interval computations were independently invented by three researchers in three different parts of the world: by M. Warmus in Poland [25, 26], by T. Sunaga in Japan [27], and by R. Moore in the USA [28–35].

The active interest in interval computations started with Moore's 1966 monograph [34]. This interest was enhanced by the fact that in addition to estimates for general numerical algorithms, Moore's monograph also described *practical* applications which have already been developed in his earlier papers and technical reports: in particular, interval computations were used to make sure that even when we take all the uncertainties into account, the trajectory of a spaceflight is guaranteed to reach the moon.

Since then, interval computations have been actively used in many areas of science and engineering [36, 37].

Comment. An early history of interval computations is described in detail in [38] and in [39]; early papers on interval computations can be found on the interval computations Web site [36].

1.9 Interval Computations: Main Techniques

General comment about algorithms and parsing. Our goal is to find the range of a given function $f(x_1, \dots, x_n)$ on the given intervals $\mathbf{x}_1 = [\underline{x}_1, \bar{x}_1], \dots, \mathbf{x}_n = [\underline{x}_n, \bar{x}_n]$.

This function $f(x_1, \dots, x_n)$ is given as an algorithm. In particular, we may have an explicit analytical expression for f , in which case this algorithm simply consists of computing this expression.

When we talk about algorithms, we usually mean an algorithm (program) written in a high-level programming language like Java or C. Such programming languages allow us to use arithmetic expressions and many other complex constructions. Most of these constructions, however, are not directly implemented inside a computer. Usually, only simple arithmetic operations are implemented: addition, subtraction, multiplication, and $1/x$ (plus branching). Even division a/b is usually not directly supported; it is performed as a sequence of two elementary arithmetic operations:

- First, we compute $1/b$.
- And then, we multiply a by $1/b$.

When we input a general program into a computer, the computer *parses* it; i.e., represents it as sequence of elementary arithmetic operations.

Since a computer performs this parsing anyway, we can safely assume that the original algorithm $f(x_1, \dots, x_n)$ is already represented as a sequence of such elementary arithmetic operations.

Interval arithmetic. Let us start our analysis of the interval computations techniques with the simplest possible case when the algorithm $f(x_1, \dots, x_n)$ simply consists of a single arithmetic operation: addition, subtraction, multiplication, or computing $1/x$.

Let us start by estimating the range of the addition function $f(x_1, x_2) = x_1 + x_2$ on the intervals $[\underline{x}_1, \bar{x}_1]$ and $[\underline{x}_2, \bar{x}_2]$. This function is increasing with respect to both its variables. We already know how to compute the range $[\underline{y}, \bar{y}]$ of a monotonic function. So, the range of addition is equal to $[\underline{x}_1 + \underline{x}_2, \bar{x}_1 + \bar{x}_2]$.

The desired range is usually denoted as $f(\mathbf{x}_1, \dots, \mathbf{x}_n)$; in particular, for addition, this notation takes the form $\mathbf{x}_1 + \mathbf{x}_2$. Thus, we can define ‘addition’ of two intervals as follows:

$$[\underline{x}_1, \bar{x}_1] + [\underline{x}_2, \bar{x}_2] = [\underline{x}_1 + \underline{x}_2, \bar{x}_1 + \bar{x}_2].$$

This formula makes perfect intuitive sense: if one town has between 700 and 800 thousand people and it merges with a nearby town whose population is between 100 and 200 thousand, then

- the smallest possible value of the total population of the new big town is when both populations are the smallest possible, i.e., $700 + 100 = 800$, and
- the largest possible value is when both populations are the largest possible, i.e., $800 + 200 = 1000$.

The subtraction function $f(x_1, x_2) = x_1 - x_2$ is increasing with respect to x_1 and decreasing with respect to x_2 , so we have

$$[\underline{x}_1, \bar{x}_1] - [\underline{x}_2, \bar{x}_2] = [\underline{x}_1 - \bar{x}_2, \bar{x}_1 - \underline{x}_2].$$

These operations are also in full agreement with common sense. For example, if a warehouse originally had between 6.0 and 8.0 tons and we moved between 1.0 and 2.0 tons to another location, then the smallest amount left is when we start with the smallest possible value 6.0 and move the largest possible value 2.0, resulting in $6.0 - 2.0 = 4.0$. The largest amount left is when we start with the largest possible value 8.0 and move the smallest possible value 1.0, resulting in $8.0 - 1.0 = 7.0$.

For multiplication $f(x_1, x_2) = x_1 \cdot x_2$, the direction of monotonicity depends on the actual values of x_1 and x_2 : e.g., when $x_2 > 0$, the product increases with x_1 ; otherwise it decreases with x_1 . So, unless we know the signs of the product beforehand, we cannot tell whether the maximum is attained at $x_1 = \underline{x}_1$ or at $x_1 = \bar{x}_1$. However, we know that it is always attained at one of these endpoints. So, to find the range

of the product, it is sufficient to try all $2 \cdot 2 = 4$ combinations of these endpoints:

$$[\underline{x}_1, \bar{x}_1] \cdot [\underline{x}_2, \bar{x}_2] =$$

$$[\min(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \bar{x}_2, \bar{x}_1 \cdot \underline{x}_2, \bar{x}_1 \cdot \bar{x}_2), \max(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \bar{x}_2, \bar{x}_1 \cdot \underline{x}_2, \bar{x}_1 \cdot \bar{x}_2)].$$

Finally, the function $f(x_1) = 1/x_1$ is decreasing wherever it is defined (when $x_1 \neq 0$), so if $0 \notin [\underline{x}_1, \bar{x}_1]$, then

$$\frac{1}{[\underline{x}_1, \bar{x}_1]} = \left[\frac{1}{\bar{x}_1}, \frac{1}{\underline{x}_1} \right].$$

The formulas for addition, subtraction, multiplication, and reciprocal of intervals are called formulas of *interval arithmetic*.

Computational complexity of interval arithmetic. Interval addition requires two additions of numbers; interval subtraction requires two subtraction of numbers, and dividing 1 by an interval requires two divisions of 1 by a real number. In all these operations, we need twice longer time to perform the corresponding interval operation than to perform an operation with real numbers.

The only exception is interval multiplication, which requires four multiplications of numbers. Thus, if we use the above formulas, we get, in the worst case, a four times increase in computation time.

Computational comment: interval multiplication can be performed faster. It is known that we can compute the interval product faster, by using only three multiplications [40, 41]. Namely,

- if $\underline{x}_1 \geq 0$ and $\underline{x}_2 \geq 0$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\underline{x}_1 \cdot \underline{x}_2, \bar{x}_1 \cdot \bar{x}_2]$;
- if $\underline{x}_1 \geq 0$ and $\underline{x}_2 \leq 0 \leq \bar{x}_2$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\bar{x}_1 \cdot \underline{x}_2, \bar{x}_1 \cdot \bar{x}_2]$;
- if $\underline{x}_1 \geq 0$ and $\bar{x}_2 \leq 0$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\bar{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \bar{x}_2]$;
- if $\underline{x}_1 \leq 0 \leq \bar{x}_1$ and $\underline{x}_2 \geq 0$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\underline{x}_1 \cdot \underline{x}_2, \bar{x}_1 \cdot \bar{x}_2]$;
- if $\underline{x}_1 \leq 0 \leq \bar{x}_1$ and $\underline{x}_2 \leq 0 \leq \bar{x}_2$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\min(\underline{x}_1 \cdot \bar{x}_2, \bar{x}_1 \cdot \underline{x}_2), \max(\underline{x}_1 \cdot \underline{x}_2, \bar{x}_1 \cdot \bar{x}_2)]$;
- if $\underline{x}_1 \leq 0 \leq \bar{x}_2$ and $\bar{x}_2 \leq 0$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\underline{x}_1 \cdot \underline{x}_2, \bar{x}_1 \cdot \bar{x}_2]$;
- if $\bar{x}_1 \leq 0$ and $\underline{x}_2 \geq 0$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\underline{x}_1 \cdot \bar{x}_2, \bar{x}_1 \cdot \underline{x}_2]$;
- if $\bar{x}_1 \leq 0$ and $\underline{x}_2 \leq 0 \leq \bar{x}_2$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\underline{x}_1 \cdot \bar{x}_2, \underline{x}_1 \cdot \underline{x}_2]$;
- if $\bar{x}_1 \leq 0$ and $\bar{x}_2 \leq 0$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\bar{x}_1 \cdot \bar{x}_2, \underline{x}_1 \cdot \bar{x}_2]$.

We see that in eight out of nine cases, we need only two multiplications, and the only case when we still need four multiplications is when $0 \in \mathbf{x}_1$ and $0 \in \mathbf{x}_2$. In this case, it can also be shown that three multiplications are sufficient:

- If $0 \leq |\underline{x}_1| \leq \bar{x}_1$ and $0 \leq |\underline{x}_2| \leq \bar{x}_2$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\min(\underline{x}_1 \cdot \bar{x}_2, \bar{x}_1 \cdot \underline{x}_2), \bar{x}_1 \cdot \bar{x}_2]$.
- If $0 \leq \bar{x}_1 \leq |\underline{x}_1|$ and $0 \leq \bar{x}_2 \leq |\underline{x}_2|$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\min(\underline{x}_1 \cdot \bar{x}_2, \bar{x}_1 \cdot \underline{x}_2), \underline{x}_1 \cdot \underline{x}_2]$.
- If $0 \leq |\underline{x}_1| \leq \bar{x}_1$ and $0 \leq \bar{x}_2 \leq |\underline{x}_2|$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\bar{x}_1 \cdot \underline{x}_2, \max(\underline{x}_1 \cdot \underline{x}_2, \bar{x}_1, \bar{x}_2)]$.
- If $0 \leq \bar{x}_1 \leq |\underline{x}_1|$ and $0 \leq |\underline{x}_2| \leq \bar{x}_2$, then $\mathbf{x}_1 \cdot \mathbf{x}_2 = [\underline{x}_1 \cdot \bar{x}_2, \max(\underline{x}_1 \cdot \underline{x}_2, \bar{x}_1, \bar{x}_2)]$.

Straightforward ('naive') interval computations: idea. We know how to compute the range for each arithmetic operation. Therefore, to compute the range $f(\mathbf{x}_1, \dots, \mathbf{x}_n)$, it is reasonable to do the following:

- first, to parse the algorithm f (this is done automatically by a compiler),
- and then to repeat the computations forming the program f step by step, replacing each operation with real numbers by the corresponding operation of interval arithmetic.

It is known that, as a result, we get an enclosure \mathbf{Y} for the desired range \mathbf{y} [34, 37].

Example where straightforward interval computations work perfectly. Let us start with an example of computing the average of two values $f(x_1, x_2) = 0.5 \cdot (x_1 + x_2)$. This function is increasing in both variables, so its range on the intervals

$$[\underline{x}_1, \bar{x}_1] \text{ and } [\underline{x}_2, \bar{x}_2] \text{ is equal to } [0.5 \cdot (\underline{x}_1 + \underline{x}_2), 0.5 \cdot (\bar{x}_1 + \bar{x}_2)].$$

A compiler will parse the function f into the following sequence of computational steps:

- we start with x_1 and x_2 ;
- then, we compute an intermediate value $x_3 = x_1 + x_2$;
- finally, we compute $y = 0.5 \cdot x_3$.

According to straightforward interval computations,

- we start with $\mathbf{x}_1 = [\underline{x}_1, \bar{x}_1]$ and $\mathbf{x}_2 = [\underline{x}_2, \bar{x}_2]$;
- then, we compute $\mathbf{x}_3 = \mathbf{x}_1 + \mathbf{x}_2 = [\underline{x}_1 + \underline{x}_2, \bar{x}_1 + \bar{x}_2]$;
- finally, we compute $\mathbf{y} = 0.5 \cdot \mathbf{x}_3$, and we get the desired range.

One can easily check that we also get the exact range for the general case of the arithmetic average and, even more generally, for an arbitrary linear function $f(x_1, \dots, x_n)$.

Can straightforward interval computations be always perfect? In straightforward interval computations, we replace each elementary arithmetic operation with the corresponding operation of interval arithmetic. We have already mentioned that this replacement increases the computation time at most by a factor of 4. So, if we started with the polynomial time, we still get polynomial time.

On the other hand, we know that the main problem of interval computations is NP-hard. This means, crudely speaking, that we cannot always compute the exact range by using a polynomial-time algorithm. Since straightforward interval computation is a polynomial-time algorithm, this means that in some cases, its estimates for the range are not exact. Let us describe a simple example when this happens.

Example where straightforward interval computations do not work perfectly. Let us illustrate straightforward interval computations on the example of a simple function $f(x_1) = x_1 - x_1^2$; we want to estimate its range when $x_1 \in [0, 1]$.

To be able to check how good is the resulting estimate, let us first find the actual range of f . According to calculus, the minimum and the maximum of a smooth (differentiable) function on an interval are attained either at one of the endpoints or at one of the extreme points, where the derivative of this function is equal to 0. So, to find the minimum and the maximum, it is sufficient to compute the value of this function at the endpoints and at all the extreme points:

- The largest of these values is the maximum.
- The smallest of these values is the minimum.

For the endpoints $x_1 = 0$ and $x_1 = 1$, we have $f(0) = f(1) = 0$. By differentiating this function and equating the derivative $1 - 2x_1$ to 0, we conclude that this function has only one extreme point $x_1 = 0.5$. At this point, $f(0.5) = 0.25$, so $\underline{y} = \min(0, 0, 0.25) = 0$ and $\bar{y} = \max(0, 0, 0.25) = 0.25$. In other words, the actual range is $\mathbf{y} = [0, 0.25]$.

Let us now apply straightforward interval computations. A compiler will parse the function into the following sequence of computational steps:

- we start with x_1 ;
- then, we compute $x_2 = x_1 \cdot x_1$;
- finally, we compute $y = x_1 - x_2$.

According to straightforward interval computations,

- we start with $\mathbf{x}_1 = [0, 1]$;
- then, we compute $\mathbf{x}_2 = \mathbf{x}_1 \cdot \mathbf{x}_1$;
- finally, we compute $\mathbf{Y} = \mathbf{x}_1 - \mathbf{x}_2$.

Here, $\mathbf{x}_2 = [0, 1] \cdot [0, 1] = [\min(0 \cdot 0, 0 \cdot 1, 1 \cdot 0, 1 \cdot 1), \max(0 \cdot 0, 0 \cdot 1, 1 \cdot 0, 1 \cdot 1)] = [0, 1]$, and so $\mathbf{Y} = [0, 1] - [0, 1] = [0 - 1, 1 - 0] = [-1, 1]$.

The resulting interval is the enclosure for the actual range $[0, 0.25]$ but it is much wider than this range. In interval computations, we say that this enclosure has *excess width*.

Reason for excess width. In the above example, it is easy to see why we have excess width. The range $[0, 1]$ for x_2 is actually exact. However, when we compute the range for \mathbf{y} as the difference $\mathbf{x}_1 - \mathbf{x}_2$, we use the general interval computations formulas which assume that x_1 and x_2 can independently take any values from the corresponding intervals \mathbf{x}_1 and \mathbf{x}_2 – i.e., all pairs $(x_1, x_2) \in \mathbf{x}_1 \times \mathbf{x}_2$ are possible. In reality, $x_2 = x_1^2$, so only the pairs with $x_2 = x_1^2$ are possible.

Interval computations go beyond straightforward technique. People who are vaguely familiar with interval computations sometimes erroneously assume that the above straightforward ('naive') technique is all there is in interval computations. In conference presentations (and even in published papers), one often encounters a statement: 'I tried interval computations, and it did not work.' What this statement usually means is that they tried the above straightforward approach and – not surprisingly – it did not work well.

In reality, interval computation is *not a single algorithm*, but a *problem* for which many different techniques exist. Let us now describe some of such techniques.

Centered form. One of such techniques is the centered form technique. This technique is based on the same Taylor series expansion ideas as linearization. We start by representing each interval $\mathbf{x}_i = [\underline{x}_i, \bar{x}_i]$ in the form $[\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$, where $\tilde{x}_i = (\underline{x}_i + \bar{x}_i)/2$ is the midpoint of the interval \mathbf{x}_i and $\Delta_i = (\bar{x}_i - \underline{x}_i)/2$ is the half-width of this interval.

After that, we use the Taylor expansion. In linearization, we simply ignore quadratic and higher order terms. Here, instead, we use the Taylor form with a remainder term. Specifically, the centered form is based on the formula

$$f(x_1, \dots, x_n) = f(\tilde{x}_1, \dots, \tilde{x}_n) + \sum_{i=1}^n \frac{\partial f}{\partial x_i}(\eta_1, \dots, \eta_n) \cdot (x_i - \tilde{x}_i),$$

where each η_i is some value from the interval \mathbf{x}_i .

Since $\eta_i \in \mathbf{x}_i$, the value of the i th derivative belongs to the interval range of this derivative on these intervals. We also know that $x_i - \tilde{x}_i \in [-\Delta_i, \Delta_i]$. Thus, we can conclude that

$$f(\mathbf{x}_1, \dots, \mathbf{x}_n) = f(\tilde{x}_1, \dots, \tilde{x}_n) + \sum_{i=1}^n \frac{\partial f}{\partial x_i}(\mathbf{x}_1, \dots, \mathbf{x}_n) \cdot [-\Delta_i, \Delta_i].$$

To compute the ranges of the partial derivatives, we can use straightforward interval computations.

Example. Let us illustrate this method on the above example of estimating the range of the function $f(x_1) = x_1 - x_1^2$ over the interval $[0, 1]$. For this interval, the midpoint is $\tilde{x}_1 = 0.5$; at this midpoint, $f(\tilde{x}_1) = 0.25$. The half-width is $\Delta_1 = 0.5$. The only partial derivative here is $\partial f / \partial x_1 = 1 - 2x_1$, its range on $[0, 1]$ is equal to $1 - 2 \cdot [0, 1] = [-1, 1]$. Thus, we get the following enclosure for the desired range \mathbf{y} :

$$\mathbf{y} \subseteq \mathbf{Y} = 0.25 + [-1, 1] \cdot [-0.5, 0.5] = 0.25 + [-0.5, 0.5] = [-0.25, 0.75].$$

This enclosure is narrower than the 'naive' estimate $[-1, 1]$, but it still contains excess width.

How can we get better estimates? In the centered form, we, in effect, ignored quadratic and higher order terms, i.e., terms of the type

$$\frac{\partial^2 f}{\partial x_i \partial x_j} \cdot \Delta x_i \cdot \Delta x_j.$$

When the estimate is not accurate enough, it means that this ignored term is too large. There are two ways to reduce the size of the ignored term:

- We can try to decrease this quadratic term.
- We can try to explicitly include higher order terms in the Taylor expansion formula, so that the remainder term will be proportional to, say, Δx_i^3 and thus be much smaller.

Let us describe these two ideas in detail.

First idea: bisection. Let us first describe the situation in which we try to minimize the second-order remainder term. In the above expression for this term, we cannot change the second derivative. The only thing we can decrease is the difference $\Delta x_i = x_i - \tilde{x}_i$ between the actual value and the midpoint. This value is bounded by the half-width Δ_i of the box. So, to decrease this value, we can subdivide the original box into several narrower subboxes. Usually, we divide it into two subboxes, so this subdivision is called *bisection*.

The range over the whole box is equal to the union of the ranges over all the subboxes. The width of each subbox is smaller, so we get smaller Δx_i and hopefully, more accurate estimates for ranges over each of these subboxes. Then, we take the union of the ranges over subboxes.

Example. Let us illustrate this idea on the above $x_1 - x_1^2$ example. In this example, we divide the original interval $[0, 1]$ into two subintervals $[0, 0.5]$ and $[0.5, 1]$. For both intervals, $\Delta x_1 = 0.25$.

In the first subinterval, the midpoint is $\tilde{x}_1 = 0.25$, so $f(\tilde{x}_1) = 0.25 - 0.0625 = 0.1875$. The range of the derivative is equal to $1 - 2 \cdot [0, 0.5] = 1 - [0, 1] = [0, 1]$; hence, we get an enclosure $0.1875 + [0, 1] \cdot [-0.25, 0.25] = [-0.0625, 0.4375]$.

For the second interval, $\tilde{x}_1 = 0.75$, so $f(0.75) = 0.1875$. The range of the derivative is $1 - 2 \cdot [0.5, 1] = [-1, 0]$; hence, we get an enclosure

$$0.1875 + [-1, 0] \cdot [-0.25, 0.25] = [-0.0625, 0.4375].$$

The union of these two enclosures is the same interval $[-0.0625, 0.4375]$. This enclosure is much more accurate than before.

Bisection: general comment. The more subboxes we consider, the smaller Δx_i and thus the more accurate the corresponding enclosures. However, once we have more boxes, we need to spend more time processing these boxes. Thus, we have a trade-off between computation time and accuracy: the more computation time we allow, the more accurate estimates we will be able to compute.

Additional idea: monotonicity checking. If the function $f(x_1, \dots, x_n)$ is monotonic over the original box $\mathbf{x}_1 \times \dots \times \mathbf{x}_n$, then we can easily compute its exact range. Since we used the centered form for the original box, this probably means that on that box, the function is not monotonic: for example, with respect to x_1 , it may be increasing at some points in this box and decreasing at other points.

However, as we divide the original box into smaller subboxes, it is quite possible that at least some of these subboxes will be outside the areas where the derivatives are 0, and thus the function $f(x_1, \dots, x_n)$ will be monotonic. So, after we subdivide the box into subboxes, we should first check monotonicity on each of these subboxes – and if the function is monotonic, we can easily compute its range.

In calculus terms, a function is increasing with respect to x_i if its partial derivative $\partial f / \partial x_i$ is non-negative everywhere on this subbox. Thus, to check monotonicity, we should find the range $[\underline{y}_i, \bar{y}_i]$ of this derivative: (We need to do it anyway to compute the centered form expression.)

- If $\underline{y}_i \geq 0$, this means that the derivative is everywhere non-negative and thus the function f is increasing in x_i .
- If $\bar{y}_i \leq 0$, this means that the derivative is everywhere non-positive and thus the function f is decreasing in x_i .

If $\underline{y}_i < 0 < \bar{y}_i$, then we have to use the centered form.

If the function is monotonic (e.g., increasing) only with respect to some of the variables x_i , then

- to compute \bar{y} , it is sufficient to consider only the value $x_i = \bar{x}_i$, and
- to compute \underline{y} , it is sufficient to consider only the value $x_i = \underline{x}_i$.

For such subboxes, we reduce the original problem to two problems with fewer variables, problems which are thus easier to solve.

Example. For the example $f(x_1) = x_1 - x_1^2$, the partial derivative is equal to $1 - 2 \cdot x_1$.

On the first subbox $[0, 0.5]$, the range of this derivative is $1 - 2 \cdot [0, 0.5] = [0, 1]$. Thus, the derivative is always non-negative, the function is increasing on this subbox, and its range on this subbox is equal to $[f(0), f(0.5)] = [0, 0.25]$.

On the second subbox $[0.5, 1]$, the range of the derivative is $1 - 2 \cdot [0.5, 1] = [-1, 0]$. Thus, the derivative is always non-positive, the function is decreasing on this subbox, and its range on this subbox is equal to $[f(1), f(0.5)] = [0, 0.25]$. The union of these two ranges is $[0, 0.25]$ – the exact range.

Comment. We got the exact range because of the simplicity of our example, in which the extreme point 0.5 of the function $f(x_1) = x_1 - x_1^2$ is exactly in the middle of the interval $[0, 1]$. Thus, when we divide the box in two, both subboxes have the monotonicity property. In the general case, the extremal point will be inside one of the subboxes, so we will have excess width.

General Taylor techniques. As we have mentioned, another way to get more accurate estimates is to use so-called *Taylor techniques*, i.e., to explicitly consider second-order and higher order terms in the Taylor expansion (see, e.g., [42–44] and references therein).

Let us illustrate the main ideas of Taylor analysis on the case when we allow second-order terms. In this case, the formula with a remainder takes the form

$$\begin{aligned} f(x_1, \dots, x_n) &= f(\tilde{x}_1, \dots, \tilde{x}_n) + \sum_{i=1}^n \frac{\partial f}{\partial x_i}(\tilde{x}_1, \dots, \tilde{x}_n) \cdot (x_i - \tilde{x}_i) \\ &\quad + \frac{1}{2} \cdot \sum_{i=1}^n \sum_{j=1}^m \frac{\partial^2 f}{\partial x_i \partial x_j}(\eta_1, \dots, \eta_n) \cdot (x_i - \tilde{x}_i) \cdot (x_j - \tilde{x}_j). \end{aligned}$$

Thus, we get the enclosure

$$\begin{aligned} f(\mathbf{x}_1, \dots, \mathbf{x}_n) &\subseteq f(\tilde{x}_1, \dots, \tilde{x}_n) + \sum_{i=1}^n \frac{\partial f}{\partial x_i}(\tilde{x}_1, \dots, \tilde{x}_n) \cdot [-\Delta_i, \Delta_i] \\ &\quad + \frac{1}{2} \cdot \sum_{i=1}^n \sum_{j=1}^m \frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{x}_1, \dots, \mathbf{x}_n) \cdot [-\Delta_i, \Delta_i] \cdot [-\Delta_j, \Delta_j]. \end{aligned}$$

Example. Let us illustrate this idea on the above example of $f(x_1) = x_1 - x_1^2$. Here, $\tilde{x}_1 = 0.5$, so $f(\tilde{x}_1) = 0.25$ and $\partial f / \partial x_1(\tilde{x}_1) = 1 - 2 \cdot 0.5 = 0$. The second derivative is equal to -2 , so the Taylor estimate takes the form $\mathbf{Y} = 0.25 - [-0.5, 0.5]^2$.

Strictly speaking, if we interpret Δx_1^2 as $\Delta x_1 \cdot \Delta x_1$ and use the formulas of interval multiplication, we get the interval $[-0.5, 0.5] \cdot [-0.5, 0.5] = [-0.25, 0.25]$, and thus the range $\mathbf{Y} = 0.25 - [-0.25, 0.25] = [0, 0.5]$ with excess width. However, we can view x^2 as a special function, for which the range over $[-0.5, 0.5]$ is known to be $[0, 0.25]$. In this case, the above enclosure $0.25 - [0, 0.25] = [0, 0.25]$ is actually the exact range.

Taylor methods: general comment. The more terms we consider in the Taylor expansion, the smaller the remainder term and thus the more accurate the corresponding enclosures. However, once we have more terms, we need to spend more time computing these terms. Thus, for Taylor methods, we also have a trade-off between computation time and accuracy: the more computation time we allow, the more accurate estimates we will be able to compute.

An alternative version of affine and Taylor arithmetic. The main idea of Taylor methods is to approximate the given function $f(x_1, \dots, x_n)$ by a polynomial of a small order plus an interval remainder term.

In these terms, straightforward interval computations can be viewed as 0th order Taylor methods in which all we have is the corresponding interval (or, equivalently, the constant term plus the remainder interval). To compute this interval, we repeated the computation of f step by step, replacing operations with numbers by operations with intervals.

We can do the same for higher order Taylor expansions as well. Let us illustrate how this can be done for the first-order Taylor terms. We start with the expressions $x_i = \tilde{x}_i - \Delta x_i$. Then, at each step, we keep a term of the type $a = \tilde{a} + \sum_{i=1}^n a_i \cdot \Delta x_i + \mathbf{a}$. (To be more precise, keep the coefficients \tilde{a} and a_i and the interval \mathbf{a} .)

Addition and subtraction of such terms are straightforward:

$$\begin{aligned} (\tilde{a} + \sum_{i=1}^n a_i \cdot \Delta x_i + \mathbf{a}) + (\tilde{b} + \sum_{i=1}^n b_i \cdot \Delta x_i + \mathbf{b}) &= (\tilde{a} + \tilde{b}) + \sum_{i=1}^n (a_i + b_i) \cdot \Delta x_i + (\mathbf{a} + \mathbf{b}); \\ (\tilde{a} + \sum_{i=1}^n a_i \cdot \Delta x_i + \mathbf{a}) - (\tilde{b} + \sum_{i=1}^n b_i \cdot \Delta x_i + \mathbf{b}) &= (\tilde{a} - \tilde{b}) + \sum_{i=1}^n (a_i - b_i) \cdot \Delta x_i + (\mathbf{a} - \mathbf{b}). \end{aligned}$$

For multiplication, we add terms proportional to $\Delta x_i \cdot \Delta x_j$ to the interval part:

$$\begin{aligned} (\tilde{a} + \sum_{i=1}^n a_i \cdot \Delta x_i + \mathbf{a}) \cdot (\tilde{b} + \sum_{i=1}^n b_i \cdot \Delta x_i + \mathbf{b}) &= (\tilde{a} \cdot \tilde{b}) + \sum_{i=1}^n (\tilde{a} \cdot b_i + \tilde{b} \cdot a_i) \cdot \Delta x_i \\ &\quad + (\tilde{a} \cdot \mathbf{b} + \tilde{b} \cdot \mathbf{a} + \sum_{i=1}^n a_i \cdot b_i \cdot [0, \Delta_i^2] + \sum_{i=1}^n \sum_{j \neq i} a_i \cdot b_j \cdot [-\Delta_i, \Delta_i] \cdot [\Delta_j, \Delta_j]). \end{aligned}$$

At the end, we get an expression of the above type for the desired quantity y : $y = \tilde{y} + \sum_{i=1}^n y_i \cdot \Delta x_i + \mathbf{y}$. We already know how to compute the range of a linear function, so we get the following enclosure for the final range: $\mathbf{Y} = \tilde{y} + [-\Delta, \Delta] + \mathbf{y}$, where $\Delta = \sum_{i=1}^n |y_i| \cdot \Delta_i$.

Example. For $f(x_1) = x_1 - x_1^2$, we first compute $x_2 = x_1^2$ and then $y = x_1 - x_2$. We start with the interval $\mathbf{x}_1 = \tilde{x}_1 - \Delta x_1 = 0.5 + (-1) \cdot \Delta_1 + [0, 0]$.

On the next step, we compute the square of this expression. This square is equal to $0.25 - \Delta x_1 + \Delta x_1^2$. Since $\Delta x_1 \in [-0.5, 0.5]$, we conclude that $\Delta x_1^2 \in [0, 0.25]$ and thus that $x_2 = 0.25 + (-1) \cdot \Delta x_1 + [0, 0.25]$.

For $y = x_1 - x_2$, we now have

$$\begin{aligned} y &= (0.5 - 0.25) + ((-1) - (-1)) \cdot \Delta x_1 + ([0, 0] - [0, 0.25]) \\ &= 0.25 + [-0.25, 0] = [0, 0.25]. \end{aligned}$$

This is actually the exact range for the desired function $f(x_1)$.

1.10 Applications of Interval Computations

General overview. Interval computations have been used in almost all areas of science and engineering in which we need guaranteed results, ranging from space exploration to chemical engineering to robotics to supercollider design. Many applications are listed in [37, 45]; some other are described in numerous books and articles (many of which are cited in the interval computations Web site [36]). Many important applications are described in the interval-related chapters of this handbook.

Most of these applications use special software tools and packages specifically designed for interval computations (see, e.g., [46]); a reasonably current list of such tools is available on the interval Web site [36].

Applications to control. One of the areas where guaranteed bounds are important is the area of control. Robust control methods, i.e., methods which stabilize a system (known with interval uncertainty) for all possible values of the parameters from the corresponding intervals, are presented, e.g., in [47, 48].

Applications to optimization: practical need. As we have mentioned earlier, one of the main objectives of engineering is to find the alternative which is the best (in some reasonable sense).

In many real-life situations, we have a precise description of what is the best; i.e., we have an *objective function* which assigns to each alternative $x = (x_1, \dots, x_n)$ a value $F(x_1, \dots, x_n)$, characterizing the overall quality of this alternative, and our goal is to find the alternative for which this quality metric attains the largest possible value.

In mathematical terms, we want to find the maximum M of a function $F(x_1, \dots, x_n)$ on a given set S , and we are also interested in finding out where exactly this maximum is attained.

Applications to optimization: idea. The main idea of using interval computations in optimization is as follows. If we compute the value of F at several points from S and then take the maximum m of the computed values, then we can be sure that the maximum M over *all* points from S is not smaller than m : $m \leq M$. Thus, if we divide the original set into subboxes and on one of these subboxes the range $[\underline{y}, \bar{y}]$ for f is $< m$, then we can guarantee that the desired maximum does not occur on this subbox. Thus, this subbox can be excluded from the future search.

This idea is implemented as the following branch-and-bound algorithm.

Applications to optimization: simple algorithm. For simplicity, let us describe this algorithm for the case when the original set S is a box.

On each step of this algorithm, we have:

- a collection of subboxes,
- interval enclosures for the range of F on each subbox, and
- a current lower bound m for the desired maximum M .

We start with the original box; as the initial estimate m , we take, e.g., the value of F at the midpoint of the original box.

On each step, we subdivide one or several of the existing subboxes into several new ones. For each new subbox, we compute the value of F at its midpoint; then, as a new bound m , we take the maximum of the old bound and of these new results. For each new subbox, we use interval computations to compute the enclosure $[\underline{Y}, \bar{Y}]$ for the range. If $\bar{Y} < m$, then the corresponding subbox is dismissed.

This procedure is repeated until all the subboxes concentrate in a small vicinity of a single point (or of a few points); this point is the desired maximum.

Example. Let us show how this algorithm will find the maximum of a function $F(x_1) = x_1 - x_1^2$ on the interval $[0, 1]$. We start with the midpoint value $m = 0.5 - 0.5^2 = 0.25$, so we know that $M \geq 0.25$.

For simplicity, let us use the centered form to compute the range of F . On the entire interval, as we have shown earlier, we get the enclosure $[-0.25, 0.75]$.

Let us now subdivide this box. In the computer, all the numbers are binary, so the easiest division is by 2, and the easiest subdivision of a box is bisection (division of one of the intervals into two equal subintervals). Since we use the decimal system, it is easier for us to divide by 5, so let us divide the original box into five subboxes $[0, 0.2], [0, 2, 0.4], \dots, [0.8, 1]$. All the values at midpoints are $\leq m$, so the new value of m is still 0.25.

The enclosure over $[0, 0.2]$ is

$$(0.1 - 0.1^2) + (1 - 2 \cdot [0, 0.2]) \cdot [-0.1, 0.1] = 0.09 - [-0.1, 0.1] = [-0.01, 0.19].$$

Since $0.19 < 0.25$, this subbox is dismissed. Similarly, the subbox $[0.8, 1]$ will be dismissed. For the box $[0.2, 0.4]$, the enclosure is

$$(0.3 - 0.3^2) + (1 - 2 \cdot [0.2, 0.4]) \cdot [-0.1, 0.1] = 0.21 - [-0.06, 0.06] = [0.15, 0.27].$$

Since $m = 0.25 < 0.27$, this subbox is not dismissed. Similarly, we keep boxes $[0.4, 0.6]$ and $[0.6, 0.8]$ – the total of three.

On the next step, we subdivide each of these three boxes, dismiss some more boxes, etc. After a while, the remaining subboxes will concentrate around the actual maximum point $x = 0.5$.

Applications to optimization: more sophisticated algorithms. Interval techniques are actually used in the best optimization packages which produce guaranteed results. Of course, these interval methods go beyond the above simple branch-and-bound techniques: e.g., they check for monotonicity to weed out subboxes where local maxima are possible only at the endpoints, they look for solutions to the equation $\partial f / \partial x_i = 0$, etc. (see, e.g., [49, 50]).

Optimization: granularity helps. In the above text, we assumed that we know the exact value of the objective function $F(x)$ for each alternative x . In reality, we often have only approximate predictions of this value $F(x)$, with some accuracy ε . In such situations, it does not make sense to waste time and optimize the function beyond this accuracy.

For example, in the simplest interval-based optimization algorithm, at each stage, we not only get the lower bound m for the desired maximum. We can also compute the upper bound \bar{M} – which can be found as the largest of the endpoints \bar{Y} of all subbox enclosures. Thus, $m \leq M = \max F(x) \leq \bar{M}$. Once we get $|\bar{M} - m| \leq \varepsilon$, we can guarantee that every value from the interval $[m, \bar{M}]$ is ε -close to M . Thus, we can produce any alternative from any of the remaining subboxes as a good enough solution.

This simple idea can often drastically decrease computation time.

Applications to mathematics. In addition to practical applications, there have been several examples when interval computations help in solving long-standing mathematical open problems.

The first such problem was the *double-bubble* problem. It is well known that of all sets with a given volume, a ball has the smallest surface area. What if we consider two sets of equal volumes, and count the area of both the outside boundaries and the boundary between the two sets? It has been conjectured that the smallest overall area is attained for the ‘double bubble’: we take two spheres, use a plane to cut off the top of one of them, do a similar cut with the second sphere, and bring them together at the cut (so that the boundary between them is a disk). The actual proof required to prove that for this configuration, the area is indeed larger than that for all other possible configurations. This proof was done by Haas *et al.* in [51], who computed an interval enclosure $[\underline{Y}, \bar{Y}]$ for all other configurations and showed that \bar{Y} is smaller than the area Y_0 corresponding to the double bubble.

Another well-known example is the *Kepler’s conjecture*. Kepler conjectured that the standard way of stacking cannonballs (or oranges), when we place some balls on a planar grid, place the next layer in the holes between them, etc., has the largest possible density. This hypothesis was proved in 1998 by T.C. Hales, who, in particular, used interval computations to prove that many other placements lead to a smaller density [52].

Beyond interval computations, towards general granular computing. In the previous text, we consider situations when we have either probabilistic, or interval, or fuzzy uncertainty.

In practice, we often have all kinds of uncertainty. For example, we may have *partial* information about probabilities: e.g., instead of the cumulative distribution function (cdf) $F(x) \stackrel{\text{def}}{=} \text{Prob}(\xi \leq x)$, we only know *bounds* $\underline{F}(x)$ and $\overline{F}(x)$ on this cdf. In this case, all we know about the probability distribution is that the actual (unknown) cdf $F(x)$ belongs to the corresponding interval $[\underline{F}(x), \overline{F}(x)]$. This probability-related interval is called a *probability box*, or a *p-box*, for short. In data processing, once we know the p-boxes corresponding to the auxiliary quantities x_i , we need to find the p-box corresponding to the desired quantity $y = f(x_1, \dots, x_n)$; such methods are described, e.g., in [53] (see also [54, 55]).

Similarly, in fuzzy logic, we considered the case when for every property A and for every value x , we know the exact value of the degree $\mu_A(x)$ to which x satisfies the property. In reality, as we have mentioned, experts can only produce interval of possible values of their degrees. As a result, *interval-valued* fuzzy sets more adequately describe expert opinions and thus, often, lead to better applications (see, e.g., [56] as well as the corresponding chapters from this handbook).

Overall, we need a *combination* of all these types of tools, a combination which is able to handle all kinds of granules, a combination termed *granular computing* (see, e.g., [57]).

Our Hopes

One of the main objectives of this handbook is that interested readers learn the techniques corresponding to different parts of granular computing – and when necessary, combine them. We hope that this handbook will further enhance the field of granular computing.

Acknowledgments

This work was supported in part by NSF grant EAR-0225670, by Texas Department of Transportation grant No. 0-5453, and by the Japan Advanced Institute of Science and Technology (JAIST) International Joint Research Grant 2006–2008. This work was partly done during the author’s visit to the Max Planck Institut für Mathematik.

References

- [1] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl. *Computational Complexity and Feasibility of Data Processing and Interval Computations*. Kluwer, Dordrecht, 1997.
- [2] S.G. Rabinovich. *Measurement Errors and Uncertainty. Theory and Practice*. Springer-Verlag, Berlin, 2005.
- [3] G. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, Upper Saddle River, NJ, 1995.
- [4] H.T. Nguyen and E.A. Walker. *A First Course in Fuzzy Logic*. CRC Press, Boca Raton, FL, 2005.
- [5] H.T. Nguyen. A note on the extension principle for fuzzy sets. *J. Math. Anal. Appl.* 64 (1978) 359–380.
- [6] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.F. Freeman, San Francisco, CA, 1979.
- [7] C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.
- [8] S. Vavasis. *Nonlinear Optimization: Complexity Issues*. Oxford University Press, New York, 1991.
- [9] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles. Computing variance for interval data is NP-hard. *ACM SIGACT News* 33(2) (2002) 108–118.
- [10] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles. Exact bounds on finite populations of interval data. *Reliab. Comput.* 11(3) (2005) 207–233.
- [11] A.A. Gaganov. *Computational Complexity of the Range of the Polynomial in Several Variables*. M.S. Thesis. Mathematics Department, Leningrad University, Leningrad, USSR, 1981.
- [12] A.A. Gaganov, Computational complexity of the range of the polynomial in several variables. *Cybernetics*, Vol. 21, (1985) 418–421.
- [13] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, 2004.
- [14] B. Chokr and V. Kreinovich. How far are we from the complete knowledge: complexity of knowledge acquisition in Dempster-Shafer approach. In: R.R. Yager, J. Kacprzyk, and M. Pedrizzi (eds). *Advances in the Dempster-Shafer Theory of Evidence*. Wiley, New York, 1994, pp. 555–576.

- [15] E.T. Jaynes and G.L. Bretthorst. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, UK, 2003.
- [16] G.J. Klir. *Uncertainty and Information: Foundations of Generalized Information Theory*. Wiley, Hoboken, NJ, 2005.
- [17] D.J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC, Boca Raton, FL, 2004.
- [18] H.M. Wadsworth (ed.). *Handbook of Statistical Methods for Engineers and Scientists*. McGraw-Hill, New York, 1990.
- [19] V. Kreinovich, J. Beck, C. Ferregut, A. Sanchez, G.R. Keller, M. Averill, and S.A. Starks. Monte-Carlo-type techniques for processing interval uncertainty, and their potential engineering applications. *Reliab. Comput.* 13(1) (2007) 25–69.
- [20] V. Kreinovich and S. Ferson. A new Cauchy-based black-box technique for uncertainty in risk analysis. *Reliab. Eng. Syst. Saf.* 85(1–3) (2004) 267–279.
- [21] Archimedes. On the measurement of the circle. In: T.L. Heath (ed.), *The Works of Archimedes*. Cambridge University Press, Cambridge, 1897; Dover edition, 1953, pp. 91–98.
- [22] W.H. Young. Sull due funzioni a piu valori constituite dai limiti d'una funzione di variabile reale a destra ed a sinistra di ciascun punto. *Rend. Acad. Lincei Cl. Sci. Fis.* 17(5) (1908) 582–587.
- [23] R.C. Young. The algebra of multi-valued quantities. *Math. Ann.* 104 (1931) 260–290.
- [24] P.S. Dwyer. *Linear Computations*. Wiley, New York, 1951.
- [25] M. Warmus. Calculus of approximations. *Bull. Acad. Pol. sci.* 4(5) (1956) 253–257.
- [26] M. Warmus. Approximations and inequalities in the calculus of approximations. Classification of approximate numbers. *Bull. Acad. Polon. Sci., Ser. Sci. Math. Astron. Phys.* 9 (1961) 241–245.
- [27] T. Sunaga. Theory of interval algebra and its application to numerical analysis. In: *RAAG Memoirs, Ggijutsu Bunken Fukuy-kai*. Research Association of Applied Geometry (RAAG), Tokyo, Japan, 1958, Vol. 2, 1958, pp. 29–46 (547–564).
- [28] R.E. Moore. *Automatic Error Analysis in Digital Computation*. Technical Report, Space Div. Report LMSD84821. Lockheed Missiles and Space Co., Sunnyvale, California, 1959.
- [29] R.E. Moore. *Interval Arithmetic and Automatic Error Analysis in Digital Computing*. Ph.D. Dissertation. Department of Mathematics, Stanford University, Stanford, CA, 1962. Published as Applied Mathematics and Statistics Laboratories Technical Report No. 25.
- [30] R.E. Moore. The automatic analysis and control of error in digital computing based on the use of interval numbers. In: L.B. Rall (ed.), *Error in Digital Computation*. Wiley, New York, 1965, Vol. I, pp. 61–130.
- [31] R.E. Moore. Automatic local coordinate transformations to reduce the growth of error bounds in interval computation of solutions of ordinary differential equations. In: L.B. Rall (ed.), *Error in Digital Computation*. Wiley, New York, 1965, Vol. II, pp. 103–140.
- [32] R.E. Moore, W. Strother, and C.T. Yang. *Interval Integrals*. Technical Report, Space Div. Report LMSD703073, Lockheed Missiles and Space Co., 1960.
- [33] R.E. Moore and C.T. Yang. *Interval Analysis I*. Technical Report, Space Div. Report LMSD285875. Lockheed Missiles and Space Co., 1959.
- [34] R.E. Moore. Prentice Hall, Englewood Cliffs, NJ, 1966.
- [35] R.E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.
- [36] Interval computations Web site, Helveticahttp://www.cs.utep.edu/interval-comp, 2008.
- [37] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis: With Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.
- [38] S. Markov and K. Okumura. The contribution of T. Sunaga to interval analysis and reliable computing. In: T. Csendes (ed.), *Developments in Reliable Computing*. Kluwer, Dordrecht, 1999, pp. 167–188.
- [39] R.E. Moore, The dawning. *Reliab. Comput.* 5 (1999) 423–424.
- [40] G. Heindl. *An Improved Algorithm for Computing the Product of Two Machine Intervals*. Interner Bericht IAGMPI- 9304. Fachbereich Mathematik, Gesamthochschule Wuppertal, 1993.
- [41] C. Hamzo and V. Kreinovich. On average bit complexity of interval arithmetic. *Bull. Eur. Assoc. Theor. Comput. Sci.* 68 (1999) 153–156.
- [42] M. Berz and G. Hoffstätter. Computation and application of Taylor polynomials with interval remainder bounds. *Reliab. Comput.* 4(1) (1998) 83–97.
- [43] A. Neumaier. Taylor forms – use and limits. *Reliab. Comput.* 9 (2002) 43–79.
- [44] N. Revol, K. Makino, and M. Berz. Taylor models and floating-point arithmetic: proof that arithmetic operations are validated in COSY. *J. Log. Algebr. Program.* 64(1) (2005) 135–154.
- [45] R.B. Kearfott and V. Kreinovich (eds). *Applications of Interval Computations*. Kluwer, Dordrecht, 1996.

-
- [46] R. Hammer, M. Hocks, U. Kulisch, and D. Ratz. *Numerical Toolbox for Verified Computing. I. Basic Numerical Problems*. Springer-Verlag, Heidelberg, New York, 1993.
 - [47] B.R. Barmish. *New Tools for Robustness of Linear Systems*. McMillan, New York, 1994.
 - [48] S.P. Bhattacharyya, H. Chapellat, and L. Keel. *Robust Control: The Parametric Approach*. Prentice-Hall, Englewood Cliffs, NJ, 1995.
 - [49] E.R. Hansen and G.W. Walster. *Global Optimization Using Interval Analysis*. MIT Press, Cambridge, MA, 2004.
 - [50] R.B. Kearfott. *Rigorous Global Search: Continuous Problems*. Kluwer, Dordrecht, 1996.
 - [51] J. Haas, M. Hutchings, and R. Schlafy. The double bubble conjecture. *Electron. Res. Announc. Am. Math. Soc.* 1 (1995) 98–102.
 - [52] T.C. Hales. A proof of the Kepler conjecture. *Ann. Math.* 162 (2005) 1065–1185.
 - [53] S. Ferson. *Risk Assessment with Uncertainty Numbers: Risk Calc*. CRC Press, Boca Raton, FL, 2002.
 - [54] V. Kreinovich, L. Longpré, S.A. Starks, G. Xiang, J. Beck, R. Kandathi, A. Nayak, S. Ferson, and J. Hajagos. Interval versions of statistical techniques, with applications to environmental analysis, bioinformatics, and privacy in statistical databases. *J. Comput. Appl. Math.* 199(2) (2007) 418–423.
 - [55] V. Kreinovich, G. Xiang, S.A. Starks, L. Longpré, M. Ceberio, R. Araiza, J. Beck, R. Kandathi, A. Nayak, R. Torres, and J. Hajagos. Towards combining probabilistic and interval uncertainty in engineering calculations: algorithms for computing statistics under interval uncertainty, and their computational complexity. *Reliab. Comput.* 12(6) (2006) 471–501.
 - [56] J.M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall, Englewood Cliffs, NJ, 2001.
 - [57] W. Pedrycz (ed.). *Granular Computing: An Emerging Paradigm*. Springer-Verlag, New York, 2001.

2

Stochastic Arithmetic as a Model of Granular Computing

René Alt and Jean Vignes

2.1 Introduction

Numerical simulation is used more and more frequently in the analysis of physical phenomena. A simulation requires several phases.

The first phase consists of constructing a physical model based on the results of experimenting with the phenomena. Next, the physical model is approximated by a mathematical model. Generally, the mathematical model contains algebraic expressions, ordinary or partial differential equations, or other mathematical features which are very complex and cannot be solved analytically. Thus, in the third phase the mathematical model must be transformed into a discrete model which can be solved with numerical methods on a computer. In the final phase the discrete model and the associated numerical methods must be translated into a scientific code by the use of a programming language. Unfortunately, when a code is run on a computer all the computations are performed using floating-point (FP) arithmetic which does not deal with real numbers but with ‘machine numbers’ consisting of a finite number of significant figures. Thus the arithmetic of the computer is merely an approximation of the exact arithmetic. It no longer respects the fundamental properties of the latter, so that every result provided by the computer always contains a round-off error, which is sometimes such that the result is false. It is therefore essential to validate all computer-generated results. Furthermore, the data used by the scientific code may contain some uncertainties. It is thus also necessary to estimate the influence of data errors on the results provided by the computer.

This chapter is made up in two parts. In the first part, after a brief recalling how round-off error propagation results from FP arithmetic, the CESTAC method (Contrôle et Estimation STochastique des Arrondis de Calcul) is summarized. This method is a probabilistic approach to the analysis of round-off error propagation and to the analysis of the influence that uncertainties in data have on computed results. It is presented from both a theoretical and a practical point of view. The CESTAC method gives rise to stochastic arithmetic which is presented as a model of granular computing in a similar fashion to interval arithmetic and interval analysis [1]. Theoretically, in stochastic arithmetic the granules are Gaussian random variables and the tools working on these granules are the operators working on Gaussian random variables. In practice, stochastic arithmetic is discretized and is termed *discrete stochastic arithmetic* (DSA). In this case granules of DSA are the samples provided by the CADNA (Control of Accuracy and

Debugging for Numerical Application) library which implements the CESTAC method. The construction of these granules and the tools working on them are detailed in this first part.

In the second part, the use of DSA via the CADNA library in three categories of numerical methods is presented. For finite methods, the use of DSA allows the detection of numerical instabilities and provides the number of exact significant digits of the results. For iterative methods the use of DSA allows iterations to be stopped as soon as a satisfactory result is reached and thus provides an optimal (in some sense) termination criterion. Additionally, it also provides the number of exact significant digits in the results. In the case of approximate methods, DSA allows the computation of an optimal step size for the numerical solution of differential equations and the computation of integrals. As in the previous cases, DSA also provides the number of exact significant digits in the results. For each of the three categories, simple but illustrative examples are presented.

2.2 Round-Off Error Propagation Induced by FP Computation

A numerical algorithm is an ordered sequence of v operations. For the sake of simplicity it is supposed that the considered algorithm provides a unique result $r \in \mathbb{R}$. When this algorithm is translated into computer and executed, FP arithmetic is used. The obtained result always contains an error resulting from round-off error propagation and is different from the exact result r . However, it is possible to estimate this error from the round-off error resulting from each FP operator.

2.2.1 Errors Due to FP Operators

Let us consider any value $x \in \mathbb{R}$ in normalized FP form: in this section lowercase letters are used for real numbers and uppercase letters are used for ‘machine numbers.’ The FP operations on machine numbers are denoted, respectively, by $\oplus, \ominus, \otimes, \oslash$. A real number x is then represented using radix b as

$$x = \varepsilon \cdot m \cdot b^e \quad \text{with} \quad \frac{1}{b} \leq m < 1, \quad (1)$$

where ε is the sign of x , m is an unlimited mantissa, b is the radix, and e is the integer exponent.

This real number x is represented on a computer working with $b = 2$ and a finite length of p bits for the mantissa as $X \in \mathbb{F}$, \mathbb{F} being the set of FP values which may be represented on a computer and expressed as

$$X = \varepsilon M \cdot 2^E, \quad (2)$$

where M is the limited mantissa encoded using p bits, including the hidden bit, and E is the exponent.

Then, the absolute round-off error resulting from each FP operator is

$$X - x = \varepsilon M \cdot 2^E - \varepsilon \cdot m \cdot b^e.$$

In what follows it is supposed that the two exponents e and E are identical, which is the case most of the time, except, e.g., if $x = 1.9999999$ and $X = 2.0000000$. So the difference $X - x$, being caused by rounding, is

$$X - x = \varepsilon 2^E (M - m),$$

with the finite mantissa M and the infinite mantissa m being identical up to the p th bit. Consequently,

- For the assignment operator, this round-off error can be expressed by equation (3):

$$X = x - \varepsilon \cdot 2^{E-p} \cdot \alpha. \quad (3)$$

For the rounding to nearest mode: $\alpha \in [-0.5, 0.5[$

For the rounding to zero mode: $\alpha \in [0, +1[$

For the rounding to $-\infty$ or to $+\infty$ mode: $\alpha \in] -1, +1[$

- For the addition operator \oplus

$$\begin{aligned} x_1 &\in \mathbb{R} & x_2 &\in \mathbb{R} & X_1 &\in \mathbb{F} & X_2 &\in \mathbb{F} \\ X_i &= x_i - \varepsilon_i 2^{E_i-p} \alpha_i & i &= 1, 2 \end{aligned} \quad (4)$$

$$X_1 \oplus X_2 = x_1 + x_2 - \varepsilon_1 2^{E_1-p} \alpha_1 - \varepsilon_2 2^{E_2-p} \alpha_2 - \varepsilon_3 2^{E_3-p} \alpha_3, \quad (5)$$

where E_3 , ε_3 , and α_3 are, respectively, the exponent, the sign, and the round-off error resulting from the FP addition.

- For the subtraction operator \ominus

$$X_1 \ominus X_2 = x_1 - x_2 - \varepsilon_1 2^{E_1-p} \alpha_1 + \varepsilon_2 2^{E_2-p} \alpha_2 - \varepsilon_3 2^{E_3-p} \alpha_3. \quad (6)$$

- For the multiplication operator \otimes

$$X_1 \otimes X_2 = x_1 x_2 - \varepsilon_1 2^{E_1-p} \alpha_1 x_2 - \varepsilon_2 2^{E_2-p} \alpha_2 x_1 + \varepsilon_1 \varepsilon_2 2^{E_1+E_2-2p} \alpha_1 \alpha_2 - \varepsilon_3 2^{E_3-p} \alpha_3. \quad (7)$$

In equation (7) the fourth term is of second order in 2^{-p} . When this term is neglected, the approximation to the first order of the round-off error resulting from the FP multiplication is expressed as equation (8):

$$X_1 \otimes X_2 \simeq x_1 x_2 - \varepsilon_1 2^{E_1-p} \alpha_1 x_2 - \varepsilon_2 2^{E_2-p} \alpha_2 x_1 - \varepsilon_3 2^{E_3-p} \alpha_3. \quad (8)$$

- For the division operator \oslash

In the same way as that for the multiplication, the approximation to the first order of the round-off error is expressed as equation (9):

$$X_1 \oslash X_2 \simeq \frac{x_1}{x_2} - \varepsilon_1 2^{E_1-p} \frac{\alpha_1}{x_2} + \varepsilon_2 2^{E_2-p} \alpha_2 \frac{x_1}{x_2^2}. \quad (9)$$

2.2.2 Error in a Computed Result

Starting from the equations in Section 2.2.1, the absolute round-off error on the computed result R of a code requiring v FP operations including assignments can be modeled to first order in 2^{-p} by equation. (10)

$$R = r + \sum_{j=1}^v g_j(d) 2^{-p} \alpha_j, \quad (10)$$

where $g_j(d)$ are quantities depending exclusively on the data and on the code but independent of the α_j 's and r and R are the exact result and the computed result respectively. This formula has been proved in [2, 3].

2.3 The CESTAC Method, a Stochastic Approach for Analyzing Round-Off Error Propagation

In the stochastic approach the basic idea is that during a run of a code round-off errors may be randomly positive or negative with various absolute values. Thus in equation (10), the coefficients α_j 's may be considered as independent random variables. The distribution law of α_j 's has been studied by several authors. First, Hamming [4] and Knuth [5] showed that the most realistic distribution of mantissas is

a logarithmic distribution. Then, on this basis, Feldstein and Goodman [6] proved that the round-off errors denoted by the α_j 's can be considered as random variables uniformly distributed on the intervals previously defined in Section 2.2.1 as soon as the number of bits p of the mantissa is greater than 10. Note that in practice $p \geq 24$. In this approach a computed result R can be considered as a random variable, and the accuracy of this result depends on the characteristics of this random variable, i.e., the mean value μ and the standard deviation σ . This means that the higher the value of $\frac{\sigma}{\mu}$, the lower the accuracy of R . But for estimating μ and σ it is necessary to obtain several samples of the distribution of R . Unfortunately, during the computation information on round-off errors is lost.

In consequence, how is it possible to obtain several samples of the computed result R ? The CESTAC method gives an easy answer to this question.

2.3.1 Basic Ideas of the Method

The CESTAC method was first developed by M. La Porte and J. Vignes [7–11] and was later generalized by the later in [12–19].

The basic idea of the method is to execute the same code N times in a synchronous manner so that round-off error propagation is different each time.

In doing this N samples are obtained using a random rounding mode.

2.3.2 The Random Rounding Mode

The idea of the random rounding mode is that each result $R \in \mathbb{F}$ of an FP operator (assignment, arithmetic operator), which is not an exact FP value, is always bounded by two FP values R^- and R^+ obtained, respectively, by rounding down and rounding up, each of them being representative of the exact result. The random rounding consists, for each FP operation or assignment, in choosing the result randomly and with an equal probability, as R^- or R^+ .

When a code is performed N times in a synchronous parallel way using this random rounding mode, N samples R_k , $k = 1, \dots, N$, are obtained of each computed result. From these samples, the accuracy of the mean value \bar{R} , considered to be the computed result, may be estimated as explained in the following section.

2.3.3 Modeling the CESTAC Method

From the formalization of the round-off error of the FP operators presented in Section 2.2, a probabilistic model of the round-off error on a computed result obtained with the random rounding mode has been proposed. This model is based on two hypotheses:

- Hyp. 1: The elementary round-off errors α_j of the FP operators are random, independent, uniformly distributed variables.
- Hyp. 2: The approximation of the first order in 2^{-p} is legitimate.

Hypothesis 2 means that the terms in 2^{-2p} , which appear in the expression of the round-off error of FP multiplications and FP divisions, have been neglected. Only the terms in 2^{-p} are considered.

It has been shown [2, 3] that if the two previous hypotheses hold, each sample R_k obtained by the CESTAC method may be modeled by a random variable Z defined by

$$Z \simeq r + \sum_{i=1}^v u_i(d) 2^{-p} z_i \quad z_i \in]-1, +1[, \quad (11)$$

where $u_i(d)$ are constants, v is the number of arithmetic operations, and z_i are the relative round-off errors on mantissas considered as independent, centered, and equidistributed variables.

Then

- $\bar{R} = E(Z) \simeq r$.
- The distribution of Z is a quasi-Gaussian distribution. Consequently to estimate the accuracy of \bar{R} it is legitimate to use Student's test which provides a confidence interval for \bar{R} and then to deduce from this interval the number of significant decimal digits $C_{\bar{R}}$ of \bar{R} , which is estimated by equation (12).

$$C_{\bar{R}} = \log_{10} \left(\frac{\sqrt{N} |\bar{R}|}{\sigma \tau_\eta} \right), \quad (12)$$

where

$$\bar{R} = \frac{1}{N} \sum_{k=1}^N R_k \quad \text{and} \quad \sigma^2 = \frac{1}{N-1} \sum_{k=1}^N (R_k - \bar{R})^2,$$

τ_η is value of Student's distribution for $N - 1$ degrees of freedom and a probability level $1 - \eta$.

Remark. The statistical property used here is the following. Let m be the unknown mean value of a Gaussian distribution. Then, if (x_k) , $k = 1, \dots, N$, are N measured values satisfying this distribution, \bar{x} is the mean value of the x_k , and σ is the empirical standard deviation defined as $\sigma = \sqrt{\frac{1}{N-1} \sum_{k=1}^N (x_k - \bar{x})^2}$, the variable $T = \frac{\bar{x}-m}{\sigma} \sqrt{N}$ satisfies a Student's distribution with $N - 1$ degrees of freedom. For example for $N = 3$ and $1 - \eta = 0.05$, i.e., a percentile of 0.975, Student's table for $N - 1 = 2$ degrees of freedom provides a value $\tau_\eta = 4.303$.

From a theoretical point of view, we may define a new number called a 'stochastic number,' which is a Gaussian variable defined by its mean value and its standard deviation. The corresponding operators for addition, subtraction, multiplication, and division define what is called stochastic arithmetic.

2.4 Stochastic Arithmetic

Stochastic arithmetic operates on stochastic numbers and is directly derived from operations on independent Gaussian random variables. Let us present here the main properties of this arithmetic, which are detailed in [20, 21].

From the granular computing point of view a stochastic number is a granule and the stochastic arithmetic is a tool for computing granules.

2.4.1 Definition of the Stochastic Operators

Definition 1. *Stochastic numbers (granules).*

The set of stochastic numbers denoted \mathbb{S} is the set of Gaussian random variables. Then an element $X \in \mathbb{S}$ is defined by $X = (m, \sigma)$, m being the mean value of X and σ being its standard deviation.

If $X \in \mathbb{S}$ and $X = (m, \sigma)$, then λ_η exists (depending only on η) such that

$$P(X \in [I_{\lambda_\eta, X}]) = 1 - \eta$$

$$I_{\lambda_\eta, X} = [m - \lambda_\eta \sigma, m + \lambda_\eta \sigma]. \quad (13)$$

$I_{\lambda_\eta, X}$ is the confidence interval of m with a probability $(1 - \eta)$. For $\eta = 0.05$, $\lambda_\eta = 1.96$. Then the number of significant digits on m is obtained by

$$C_{\lambda_\eta, X} = \log_{10} \left(\frac{|m|}{\lambda_\eta \sigma} \right). \quad (14)$$

Definition 2. *Stochastic zero.*

$X \in \mathbb{S}$ is a stochastic zero, denoted $\underline{0}$, if and only if

$$C_{\eta, X} \leq 0 \quad \text{or} \quad X = (0, 0).$$

Note that if mathematically $f(x) = 0$, then with stochastic arithmetic $F(X) = \underline{0}$.

Definition 3. *Stochastic operators (tools working on granules).*

The four elementary operations of stochastic arithmetic between two stochastic numbers $X_1 = (m_1, \sigma_1)$ and $X_2 = (m_2, \sigma_2)$, denoted by $s+$, $s-$, $s\times$, $\sqrt{\cdot}$, are defined by

$$\begin{aligned} X_1 \quad s+ \quad X_2 &\stackrel{\text{def}}{=} \left(m_1 + m_2, \sqrt{\sigma_1^2 + \sigma_2^2} \right), \\ X_1 \quad s- \quad X_2 &\stackrel{\text{def}}{=} \left(m_1 - m_2, \sqrt{\sigma_1^2 + \sigma_2^2} \right), \\ X_1 \quad s\times \quad X_2 &\stackrel{\text{def}}{=} \left(m_1 m_2, \sqrt{m_2^2 \sigma_1^2 + m_1^2 \sigma_2^2 + \sigma_1^2 \sigma_2^2} \right), \\ X_1 \quad \sqrt{\cdot} \quad X_2 &\stackrel{\text{def}}{=} \left(m_1/m_2, \sqrt{\left(\frac{\sigma_1}{m_2}\right)^2 + \left(\frac{m_1 \sigma_2}{m_2}\right)^2} \right), \quad \text{with } m_2 \neq 0. \end{aligned}$$

Remark. The first three formulas including stochastic multiplication are exact. However the last one is often simplified by considering only the first-order terms in $\frac{\sigma}{m}$, i.e., by suppressing the term $\sigma_1^2 \sigma_2^2$ [21], but this is acceptable only when this ratio is small, which is not always the case. The formula for the division is defined only to the first-order terms in $\frac{\sigma}{m}$.

Definition 4. *Stochastic equality.*

Equality between two stochastic numbers X_1 and X_2 is defined as follows: X_1 is stochastically equal to X_2 , denoted by $X_1 = X_2$, if and only if

$$X_1 = X_2 \Rightarrow |m_1 - m_2| \leq \lambda_\eta \sqrt{\sigma_1^2 + \sigma_2^2}. \quad (15)$$

Definition 5. *Order relations between stochastic numbers.*

- X_1 is stochastically greater than X_2 , denoted $X_1 > X_2$, if and only if

$$m_1 - m_2 > \lambda_\eta \sqrt{\sigma_1^2 + \sigma_2^2}. \quad (16)$$

- X_1 is stochastically greater than or equal to X_2 , denoted $X_1 \geq X_2$, if and only if

$$m_1 \geq m_2 \quad \text{or} \quad |m_1 - m_2| \leq \lambda_\eta \sqrt{\sigma_1^2 + \sigma_2^2}. \quad (17)$$

Based on these definitions the following properties of stochastic arithmetic have been proved [21, 22]:

1. $m_1 = m_2 \Rightarrow X_1 = X_2$.
2. $=$ is a reflexive and symmetric relation, but it is not a transitive relation.
3. $X_1 > X_2 \Rightarrow m_1 > m_2$.
4. $m_1 \geq m_2 \Rightarrow X_1 \geq X_2$.
5. $<$ is the opposite of $>$.
6. $>$ is a transitive relation.

7. \leq_s is a reflexive and symmetric relation, but is not a transitive relation.
 8. $\underline{0}$ is absorbent; i.e., $\forall X \in \mathbb{S}, \underline{0} \times X_s = \underline{0}$.

2.4.2 Some Algebraic Structures of Stochastic Arithmetic

As seen above stochastic numbers are Gaussian random variables with a known mean value and a known standard deviation.

It can be seen that algebraic structures close to those existing in the set of real numbers can be developed for stochastic numbers. But this is beyond the scope of this handbook.

As an example a numerical example of the algebraic solution of linear systems of equations with right-hand sides involving stochastic numbers is presented. The aim of this example is to show how a theory can be developed for a better understanding of the properties of the CESTAC method. In particular, it must be noticed that the signs of errors are unknown but when computing with these errors, operations are done with their signs. Consequently, as errors are represented in the theory by standard deviations of Gaussian variables, a sign must be introduced for them. This is done in the same way that intervals are extended to generalized intervals in which the bounds are not ordered. A stochastic number (m, σ) for which σ may be negative is called a generalized stochastic number.

For a detailed presentation of the theory, see [23–26].

The solution of a linear system with stochastic right-hand sides [24]. We shall use here the special symbol $\dot{+}$ for the arithmetic addition over standard deviations and the special symbol $*$ for the multiplication of standard deviations by scalars. These operations are different from the corresponding ones for numbers but the use of the same symbol $*$ for the multiplication of standard deviations or stochastic numbers by a scalar causes no confusion. The operations $\dot{+}$ and $*$ induce a special arithmetic on the set \mathbb{R}^+ . We consider a linear system $Ax = b$, such that A is a real $n \times n$ matrix and the right-hand side b is a vector of stochastic numbers. Then the solution x also consists of stochastic numbers, and in consequence, all arithmetic operations (additions and multiplications by scalars) in the expression Ax involve stochastic numbers; therefore, we shall write $A * x$ instead of Ax .

Problem.. Assume that $A = (a_{ij})$, $i, j = 1, \dots, n$, $a_{ij} \in \mathbb{R}$ is a real $n \times n$ matrix, and $B = (b, \tau)$ is a n -tuple of (generalized) stochastic numbers, such that $b, \tau \in \mathbb{R}^n$, $b = (b_1, \dots, b_n)$, and $\tau = (\tau_1, \dots, \tau_n)$. We look for a (generalized) stochastic vector $X = (x, \sigma)$, $x, \sigma \in \mathbb{R}^n$, i.e., an n -tuple of stochastic numbers, such that $A * X = B$.

Solution.. The i th equation of the system $A * X = B$ reads $a_{i1} * x_1 \dot{+} \dots \dot{+} a_{in} * x_n = b_i$. Obviously, $A * X = B$ reduces to a linear system $Ax = b$ for the vector $x = (x_1, \dots, x_n)$ of mean values and a system $A * \sigma = \tau$ for the standard deviations $\sigma = (\sigma_1, \dots, \sigma_n)$. If $A = (a_{ij})$ is non-singular, then $x = A^{-1}b$. We shall next concentrate on the solution of the system $A * \sigma = \tau$ for the standard deviations.

The i th equation of the system $A * \sigma = \tau$ reads $a_{i1} * \sigma_1 + \dots + a_{in} * \sigma_n = \tau_i$. It has been proved [21] that this is equivalent to

$$a_{i1}^2 \text{sign}(\sigma_1) \sigma_1^2 \dot{+} \dots \dot{+} a_{in}^2 \text{sign}(\sigma_n) \sigma_n^2 = \text{sign}(\tau_i) \tau_i^2, \quad i = 1, \dots, n,$$

with $\text{sign}(\sigma_j) = 1$ if $\sigma_j \geq 0$ and $\text{sign}(\sigma_j) = -1$ if $\sigma_j < 0$.

Setting $y_i = \text{sign}(\sigma_i) \sigma_i^2$ and $c_i = \text{sign}(\tau_i) \tau_i^2$, we obtain a linear $n \times n$ system $Dy = c$ for $y = (y_i)$, where $D = (a_{ij}^2)$ and $c = (c_i)$. If D is non-singular, we can solve the system $Dy = c$ for the vector y and then obtain the standard deviation vector σ by means of $\sigma_i = \text{sign}(y_i) \sqrt{|y_i|}$. Thus for the solution of the original problem it is necessary and sufficient that both matrices $A = (a_{ij})$ and $D = (a_{ij}^2)$ are non-singular.

Summarizing, to solve $A * X = B$ the following steps are performed:

1. Check the matrices $A = (a_{ij})$ and $D = (a_{ij}^2)$ for non-singularity.
2. Find the solution for mean values; i.e., solve the linear system $Ax = b$.

3. Find the solution $y = D^{-1}c$ of the linear system $Dy = c$, where $c = (c_i)$ and $c_i = \text{sign}(\tau_i)\tau_i^2$. Compute $\sigma = \text{sign}(y_i)\sqrt{|y_i|}$.
4. The solution of $A * X = B$ is $X = (x, \sigma)$.

Numerical Experiments

Numerical experiments, using imprecise stochastic data, have been performed to compare the theoretical results with numerical results obtained using the CESTAC method, implemented in the CADNA library (see Section 2.8). As an example, the two solutions obtained for a linear system are reported below.

Let $A = \{a_{ij}\}$ be a real matrix such that $a_{ij} = i$, if $i = j$ else $a_{ij} = 10^{-|i-j|}$, $i, j = 1, \dots, n$.

Assume that B is a stochastic vector such that the component B_i is a stochastic number with a mean value $b_i = \sum_{j=1}^n a_{ij}$ and a standard deviation for each component equal to $1 \cdot e - 4$. The centers x_i of the components of the solution are thus close to 1 and present no difficulty for their computation.

The theoretical standard deviations are obtained according to the method described in the previous section. First, matrix D is computed from the matrix A , then $Dy = c$ is solved, and then $\sigma_i = \text{sign}(y_i)\sqrt{|y_i|}$ is computed.

For a correct comparison of the solution provided by the CADNA library and the theoretical solution, accurate values for the standard deviations are obtained as follows. Twenty different vectors $b^{(k)}$, $k = 1, \dots, 20$, for the right-hand side have been randomly generated, and the corresponding twenty systems $A * X = B^{(k)}$ have been solved. For each component of $B^{(k)}$, the standard deviation of the $N = 3$ samples has been computed with the CADNA software and then the mean value of the standard deviations has been computed for each component and presented in Table 2.1.

As we can see in Table 2.1, the theoretical standard deviations and the computed values are very close.

To conclude this subsection we comment that the theoretical study of the properties of stochastic numbers allows us to obtain a rigorous abstract definition of stochastic numbers with respect to the operations of addition and multiplication by scalars. This theory also allows the solution of algebraic problems with stochastic numbers. Moreover, this provides a possibility of comparing algebraically obtained results with practical applications of stochastic numbers, such as those provided by the CESTAC method [27].

Remark. The authors are grateful to Professor S. Markov from the Bulgarian Academy of Sciences and to Professor J.L. Lamotte from University Pierre et Marie Curie (Paris, France) for their contribution to the above section.

Table 2.1 Theoretical and computed standard deviations

Component	Theoretical standard deviations, x''	Computed standard deviations
1	9.98e-05	10.4e-05
2	4.97e-05	4.06e-05
3	3.32e-05	3.21e-05
4	2.49e-05	2.02e-05
5	1.99e-05	1.81e-05
6	1.66e-05	1.50e-05
7	1.42e-05	1.54e-05
8	1.24e-05	1.02e-05
9	1.11e-05	0.778e-05
10	0.999e-05	0.806e-05

2.5 Validation and Implementation of the CESTAC Method

2.5.1 Validation and Reliability of the CESTAC Method

The theoretical validation of the CESTAC method is therefore established if and only if the two previous hypotheses hold. But its efficiency in scientific codes can be guaranteed only if its underlying hypotheses hold in practice:

- Concerning hypothesis 1, because of the use of the random rounding mode, round-off errors α_i are random variables; however, in practice they are not rigorously centered and in this case Student's test leads to a biased estimation of the computed result. It may be thought that the presence of a bias seriously jeopardizes the reliability of the CESTAC method. In fact it has been proved in [13] that it is the ratio q of the bias divided by the standard deviation σ which is the key of the reliability of equation (12).

It is shown in [13, 21] that a magnitude of q of several tens only induces an error less than one decimal significant digit on $C_{\bar{R}}$ computed with equation (12). This great robustness of equation (12) is due first to the use of the logarithm and second to the natural robustness of Student's test. Consequently in practice even if hypothesis 1 is not exactly satisfied, it is not a drawback for the reliability of equation (12).

- Concerning hypothesis 2, the approximation to first order only concerns multiplications and divisions, because in formulas (5) and (6) for round-off errors in additions or subtractions, the second-order terms, i.e., those in 2^{-2p} , do not exist. For the first-order approximation to be legitimate, it is shown in [2, 20] that if ε_1 and ε_2 are, respectively, the absolute round-off errors on the operands $X_1 \in \mathbb{F}$ and $X_2 \in \mathbb{F}$, the following condition must be verified:

$$\text{Max} \left[\left| \frac{\varepsilon_1}{X_1} \right|, \left| \frac{\varepsilon_2}{X_2} \right| \right] \ll 1. \quad (18)$$

Hence, the more accurate the computed results, the more legitimate the first-order approximation. However, if a computed result becomes non-significant, i.e., if its round-off error is of the same order of magnitude as itself, then the first-order approximation may not be legitimate.

In other words, with the use of the CESTAC method hypothesis 2 holds when

1. The operands of any multiplication are both significant.
2. The divisor of any division is significant.

As a consequence, validation of the CESTAC method requires during the run of the code to control steps (1) and (2). Indeed if (1) or (2) is not satisfied, this means that hypothesis 2 has been violated and then the results obtained with equation (12) must be considered as unreliable. This control is achieved with the concept of the computational zero described in Section 2.6. This control is in practice performed by the CADNA library, which is sketched in Section 2.8.

2.5.2 Implementation of the CESTAC Method

The two main features of the CESTAC method are as follows:

- The random rounding for each arithmetical operation, which consists in randomly choosing either the result rounded to up ρ^+ or the result rounded to down ρ^- .
- Performing the N runs of code.

To set these features in context we must consider the period pre-1988 when FP arithmetic was machine dependent and post-1988 when it was standardized by IEEE.

Asynchronous Implementation

Before 1988, FP arithmetic was highly computer dependent. Scientific computers as IBM, CDC, and CRAY worked with different rounding modes either with a chopping mode (rounding to zero) or with a rounding to the nearest mode. Sometimes, even on the same computer some arithmetic operations were performed with the chopping mode and others with the rounding to the nearest mode.

At this time an implementation which violates the hypotheses of the method has been used in a software named Prosolver [28]. As a consequence this flawed software has been the origin of some criticisms as in [29, 30], which have been erroneously attributed to the method. This implementation has also been used later in the Monte Carlo arithmetic (see [31–33]).

In this implementation, which is called ‘asynchronous implementation,’ the N runs of a code were performed independently. This means that the code was first run to completion and then it was run a second time and so on until the N th run. In addition, in the software Prosolver, the random rounding mode consisted in randomly adding ± 1 or 0 to the last bit of every FP operation result. This random rounding mode is unsatisfactory because, even when the result of an FP operation is an exact FP value, it is increased or decreased by one unit on the last position (ulp). The main criticisms of this implementation were that the random rounding used as defined before violates theorems about exact rounding, and when a computation is virulently unstable but in a way that almost always diverges to the same wrong destination, such a randomized recomputation almost always yields the same wrong result.

The correct implementation is now described in the following section.

Correct Synchronous Implementation

It is only since 1990 that the standard IEEE 754 FP arithmetic has been available to users. Around the same time scientific languages began to provide users with the capability of overloading operators. With IEEE 754 arithmetic and the overloading statements it is easy to implement the CESTAC method correctly.

- A correct random rounding mode

It was proposed in Section 2.3.2 to choose ρ^- or ρ^+ as result of FP operator. In practice we use the IEEE 754 rounding toward $+\infty$ and toward $-\infty$. Rounding occurs only when an arithmetic operation has a result that is not exact. Therefore no artificial round-off error is introduced in the computation. The choice of the rounding is at random with an equal probability for the $(N - 1)$ first samples, with the last one chosen as the opposite of the $(N - 1)$ th sample.

With this random rounding the theorems on exact rounding are respected.

- Synchronous runs

We have seen previously that to control the reliability of the CESTAC method it is absolutely necessary to detect during the run of the code the emergence of computational zeroes. To achieve this it suffices to use the synchronous implementation which consists of performing each FP operator N times with the random rounding mode before performing the next operation. Thus everything proceeds as if N identical codes were running simultaneously on N synchronized computers each using the random rounding mode. Thus for each numerical result we have N samples, from which with equation (12) the number of significant decimal digits of the mean value, considered as the computed result, is estimated. With this implementation a DSA may be defined, allowing during the run of the code to control dynamically the reliability of the CESTAC method.

Thus it is possible dynamically to

- control the round-off error propagation of each FP operation,
- detect a loss of accuracy during the computation,
- control the branching statements, and
- detect a violation of hypothesis 2 which guarantees the reliability of the method.

2.6 Discrete Stochastic Arithmetic (DSA)

The concept of the computational zero and the synchronous implementation of the CESTAC method leads to operations on N -tuples as referred to discrete stochastic numbers. Operation on these numbers is

also termed DSA. The salient properties of this arithmetic, which is detailed in [16, 17, 34], are presented here.

From the granular computing point of view, a discrete stochastic number is a granule and the DSA is a tool for computing granules.

2.6.1 Discrete Stochastic Arithmetic Operators

Definition 6. *Discrete stochastic numbers (granules).*

A discrete stochastic number is an N -tuple formed by the N samples provided by the synchronous implementation of the CESTAC method.

Definition 7. *Discrete stochastic arithmetic (tools working on granules).*

DSA operates on discrete stochastic numbers. The result of the four discrete stochastic operators is by definition the result of the corresponding arithmetic operation provided by the CESTAC method. Let X , Y , and Z be discrete stochastic numbers, and let \diamond be an FP arithmetic operator $\diamond \in [\oplus, \ominus, \otimes, \oslash]$, as defined in Section 2.2,

$$X = (X_1, \dots, X_N), \quad Y = (Y_1, \dots, Y_n), \quad Z = (Z_1, \dots, Z_N).$$

Then any of the four stochastic arithmetic operations $s+$, $s-$, $s\times$, $s\oslash$, denoted \diamond , is defined as

$$Z = X_s \diamond Y \Rightarrow Z = ((X_1 \diamond Y_1)^\pm, \dots, (X_N \diamond Y_N)^\pm), \quad (19)$$

where \pm means that the FP operation \diamond has been randomly performed with the rounding toward $+\infty$ or toward $-\infty$, as explained previously. Thus any discrete stochastic operator provides a result that is an N -tuple obtained from the corresponding FP operator operating on the components of the two operands the result of which is rounded at random toward $+\infty$ or $-\infty$.

Remark. To simplify the notations the ones for the discrete stochastic operators are chosen to be the same as those for the (continuous) stochastic operators.

Then with DSA it is straightforward using equation (12) to estimate the number of significant decimal digit of any result produced by a DSA operator.

Definition 8. *Discrete stochastic zero (computational zero) [15].*

Any discrete stochastic number $X = (X_1, X_2, \dots, X_N)$ is a discrete stochastic zero, also called computational zero, denoted $@.0$, if one of the two following condition holds:

1. $\forall i, X_i = 0, \quad i = 1, \dots, N.$
2. $C_{\bar{X}} \leq 0$, where $C_{\bar{X}}$ is obtained from equation (12).

2.6.2 Discrete Stochastic Relations (Tools Working on Granules)

From the concept of the discrete stochastic zero $@.0$, discrete stochastic relations can now be defined.

Let X , Y be discrete stochastic numbers, it is possible to define equality and order relations for these numbers. They are called discrete stochastic equality and discrete order relations and are defined as follows.

Definition 9. Discrete stochastic equality denoted by $s =$.

The discrete stochastic equality is defined by

$$X_s = Y \quad \text{if} \quad Z = X_s - Y = @.0.$$

Definition 10. Discrete stochastic inequalities denoted by $s >$ and $s \geq$.

These are defined by

$$\begin{aligned} X_s > Y &\quad \text{if} \quad \bar{X} > \bar{Y} \quad \text{and} \quad X_s - Y \neq @.0; \\ X_s \geq Y &\quad \text{if} \quad \bar{X} \geq \bar{Y} \quad \text{or} \quad X_s - Y = @.0. \end{aligned}$$

With this DSA it is possible during the execution of a code to follow the round-off error propagation, detect numerical instabilities, check branchings, and check hypotheses that guarantee the reliability of equation (12).

2.7 Taking into Account Data Errors

In real-life problems, data often come from measurements and thus contain errors issuing from sensors. Most of the time data errors may be considered as centered Gaussian random variables. It is then absolutely necessary to estimate the effect of these errors on the numerical results provided by DSA.

In a similar fashion to estimating equation (11), let us consider a finite sequence of v arithmetic operations, providing a single result r and requiring nd uncertain data d_i , $i = 1, \dots, nd$. Let δ_i be the data error on each d_i . These δ_i 's may be considered as Gaussian variables with a standard deviation of σ_i . It has been proved [3, 35] that when the previous finite sequence is performed with DSA, each data D_i , $i = 1, \dots, nd$, is defined by

$$D_i = d_i(1 + 2\theta\sigma_i). \quad (20)$$

θ being a random number uniformly distributed on $] -1, +1[$, then each N -tuple of the computer result R may be modeled by a Gaussian random centered variable

$$R \simeq r + \sum_{i=1}^{nd} v_i(d) 2^{-p} \delta_i + \sum_{i=1}^v g_i(d) 2^{-p} z_i, \quad (21)$$

$v_i(d)$ being quantities depending exclusively on the data and on the code. This formula is an extension of equation (11). Indeed the first quantity represents the error coming from uncertainties of data and the second represents the round-off error propagation. Then to estimate the number of significant decimal digits in the computed result \bar{R} it suffices to use equation (21). In this estimation both errors (uncertainties of data and round-off error) have been taken in account.

In the framework of granular computing each data item D_i is a granule elaborated from (20), which is an operand for the DSA operators.

2.8 The CADNA Library [36]

The CADNA library has been written in Fortran, C++, and ADA. It is presented in detail in [20]. It is the Fortran version that is described here. The CADNA library automatically implements DSA in any Fortran code. For CADNA Fortran and CADNA ADA, $N = 3$ has been chosen. But for CADNA C++, the value of N must be chosen by the user. Furthermore, the probability is here chosen to the classical level of $\eta = 0.95$. As seen in the beginning, in equation (12) for $N = 3$ and $1 - \eta = 0.05$

the value of Student's table for $N - 1 = 2$ degrees of freedom is $\tau_{\eta} = 4.303$. Thus the CADNA library enables a user to run a scientific code with DSA on a conventional computer without having to rewrite or even substantially modify the Fortran source code. A new 'stochastic number' type has been created, which is a triplet (because $N = 3$), each component being a sample provided by the random rounding.

All the Fortran arithmetic operators, $+, -, *, /$, have been overloaded, so that when such an operator is performed, the operands and the result are stochastic numbers. In the same way the relational operators such as $==, >, \geq, <, \leq$ have also been overloaded, satisfying the properties of the discrete stochastic relations.

Moreover, all the standard functions defined in Fortran 77 have also been overloaded. Similarly, the printing statement has been modified and gives the computer result written only with its exact number of significant digits, estimated by equation (12). Furthermore, in order to estimate the effect of data errors on a result provided by the computer, a special function has been created that allows the user to introduce uncertainties into these data. This function must always be used associated with assignment statements when data are not exact FP values.

The modifications that the user has to make to the Fortran source are mainly to change the declaration statements of real type to stochastic type, and the input–output statements. Thus, see [36], when a modified Fortran source combined with the CADNA library is run, it is as if ($N = 3$) identical codes were simultaneously run on N synchronized computers, each of them using the random rounding mode. So round-off error propagation can be analyzed step by step and then any numerical anomaly can be detected.

One major feature of the CADNA library is that this dynamical analysis is performed during the execution of code. As soon as a numerical anomaly is detected, a warning is written to a special CADNA file.

These warnings are divided into two categories: those concerning the reliability of the results provided and those concerning the numerical debugging of the code.

- Concerning the reliability of the results, the warnings are
 - unstable multiplication (the operands of the multiplication are computational zeroes),
 - unstable power (the operand of the power is a computational zero), and
 - unstable division (the divisor is a computational zero).
- Concerning the debugging of the code, the warnings are
 - instabilities in functions (*SIGN*, *MOD*, *DIM*, *LOG*, *SIN*, . . .),
 - computational zero detected in a branching, and
 - sudden loss of accuracy.

When a code has been instrumented with the CADNA library and run, the user must always consult the special CADNA file. If it is empty, this means that no anomaly has been detected and that the computed results provided by the code are reliable and that the number of significant decimal digits of each of them is correctly estimated up to 1.

If the special CADNA file contains warnings, the following two cases must be considered:

1. One or several warnings belonging to the first category appear in the file. This means that hypothesis 2 has been violated and so the results provided by the code must be considered as unreliable.
2. One or several warnings belonging to the second category appear. This means that instabilities have been detected. In this case, the user is able, with the use of the debugger, to identify the statement in which the anomaly has appeared. The user must then try to improve the stability of the code, for instance, by replacing unstable formulas by more stable ones.

With the special CADNA file the user knows the numerical behavior of the code and then may conclude about the reliability of results obtained.

2.9 The Use of the CADNA Library

In scientific computing, numerical methods are used for solving problems on a computer. These numerical methods can be classified in three categories: finite methods, iterative methods, and approximate methods.

2.9.1 Finite Methods

A method of this class consists of a finite ordered sequence of arithmetic operations and branchings depending on some criteria, e.g., elimination methods for linear systems and more general scientific computations that involve a succession of algebraic formulas. As shown earlier, when these methods are performed on a computer with the usual FP arithmetic, false results may be obtained without any warning. But when these methods are performed with DSA and using the CADNA library, numerical instabilities are detected and the result is provided with its accuracy. To illustrate this, consider the following example.

Example 1. This example has been proposed in [29, 30]. It is an adaptation from another example [37] and consists in computing the result of the following formula;

$$t = L - \frac{(M - N/(L - (M - N/z)/(L - (M - N/y)/z)))}{(L - (M - N/(L - (M - N/y)/z))/(L - (M - N/Z)/(L - (M - N/y)/z)))} \quad (22)$$

with

$$\begin{aligned} L &= a + b + c & M &= a(b + c) + bc & N &= abc \\ x &= \frac{b + c}{2} & y &= \frac{b.b + c.c}{b + c} & z &= L - \frac{M - N/x}{y} \\ a &= 3.10^8 & b &= 6 & c &= 5 \end{aligned}$$

The exact result is $t = (b^7 + c^7)/(b^6 + c^6)$; i.e., $t = 358061/62281 = 5.7491209197$.

With IEEE standard arithmetic and any of its rounding mode, the obtained result is always $t = 3.0 \times 10^8$ and of course the user is not informed that this result is false.

With the Prosolver software the result obtained is again $t = 3.0 \times 10^8$. So Prosolver has also failed.

With the CADNA library the result is the same, $t = 3.0 \times 10^8$, but the special CADNA file contains six ‘unstable division’ warnings. As explained previously, this means that hypothesis 2 has been violated. Thus the provided result is not reliable and must be considered as an incorrect result. CADNA has not failed.

Example 2. This system of linear equations has been proposed by J.H. Wilkinson and can be found in [14]. It is defined as follows: $W_n \cdot \mathcal{X} = \mathcal{B}$,

with $W_n = (w_{i,j})$, $i, j = 1, \dots, n$, with

$$\left\{ \begin{array}{ll} w_{i,i} = 1.0 \\ w_{i,j} = -1.0 & \text{for } i > j \\ w_{i,j} = 0.0 & \text{for } i < j \\ w_{i,n} = 1.0 & \text{for } i \in [1, n-1] \\ w_{n,n} = \alpha = 0.9 \end{array} \right. \quad (23)$$

In this system the diagonal and the n th column and are $(1, 1, \dots, \alpha)$, the elements of the upper triangular sub-matrix are null except the last column, and those of the lower triangular sub-matrix are -1 . The n elements of right-hand side \mathcal{B} are equal to 1.

It is easy to show that the exact solution of this system, which is not ill conditioned, is

$$\begin{cases} x_i^* = -2^{i-1} \frac{1-\alpha}{\Delta^*} & i = [1, n-1] \\ x_n^* = \frac{2^{n-1}}{\Delta^*} \end{cases} \quad (24)$$

$\Delta^* = 2^{n-1} - 1 + \alpha$, Δ^* being the determinant of the matrix.

This system has been solved using Gaussian elimination method with partial pivoting, first with the IEEE 754 standard double precision and then with the CADNA library, for $n \in [30, 35, 40, 45, 50]$.

The determinant computed with both the IEEE 754 standard and the CADNA library yields results correct to 15 significant decimal digits.

Concerning the solution X_i , $i = 1, n-1$, we find the following results which are detailed in Table 2.2.

- With the IEEE 754 standard some of the last digits are false, but of course the user is not informed of the failure.
- With the CADNA library, only the N decimal digits estimated to be exact up to 1 by the software are provided. It can be seen in Table 2.2 that these are in perfect agreement with the number of exact digits, N^* , obtained by comparing the CADNA solution to the exact solution x_i^* , $i = 1, n-1$.

The following example concerns a problem with uncertain data solved by the CADNA library. To perturb the data, CADNA uses a special function constructed according to formula (20).

Example 3. Study of the sensitivity of a determinant to the coefficients of the matrix.

Let us consider the determinant proposed in [38]:

$$\Delta = \begin{vmatrix} -73 & 78 & 24 \\ 92 & 66 & 25 \\ -80 & 37 & 10 \end{vmatrix}. \quad (25)$$

The exact value of this determinant is $\Delta = 1$.

When this determinant is computed with IEEE 754 FP arithmetic in double precision using different rounding modes, the results obtained are as follows:

- with the rounding to nearest mode $\Delta = 0.9999999999468869$,
- with the rounding to zero mode $\Delta = 0.999999999\underline{468865}$,
- with the rounding to $-\infty$ mode $\Delta = 0.999999999\underline{894979}$, and
- with the rounding to $+\infty$ mode $\Delta = 1.000000000747207$.

The underlined digits are false but obviously the user is not aware of this fact. When the determinant is computed with the CADNA library, the result is $\Delta = 1.000000000$. Note that the result is printed with only ten digits, which is the best accuracy which can be obtained.

Table 2.2 Accuracy of the solution of system (23) for different size n

n	Number of false last decimal digits (IEEE 754 standard)	Number of decimal digits, N (CADNA library)	Number of exact decimal digits, N^*
30	9	6	6
35	10	4	5
40	12	3	3
45	13	1	2
50	15	0	0

Table 2.3 Number of exact decimal digits of Δ as a function of ε

ε	10^{-15}	10^{-13}	10^{-11}	10^{-9}	10^{-7}	10^{-5}
\mathcal{N}	10	8	6	4	2	0

Suppose now that coefficients $a_{12} = 78$ and $a_{33} = 10$ of the matrix are uncertain data. This means that they both contain a relative error ε , which is taken here to be the same. In other words,

$$a_{12} \in [78 - 78\varepsilon, 78 + 78\varepsilon] \quad \text{and} \quad a_{33} \in [10 - 10\varepsilon, 10 + 10\varepsilon].$$

The CADNA library, as explained above, is an effective tool for estimating the influence of data uncertainties on the computed determinant.

Table 2.3 presents the number of exact decimal digits, \mathcal{N} , provided by CADNA in the computed determinant (25) as a function of ε , which determines the uncertainty of a_{12} and a_{33} .

From these results it clearly appears that if the magnitude of uncertainty in the coefficients is greater than or equal to 10^{-5} , then the determinant cannot be computed since the result obtained is not significant.

2.9.2 Iterative Methods

From the mathematical standpoint, these methods, starting from an initial point x_0 considered as an approximation of the solution to the problem to be solved, consist in computing a sequence x_1, x_2, \dots, x_k that is supposed to converge to the solution.

So, let us consider here an iterative sequence defined by

$$x_{k+1} = \varphi(x_k) \quad \mathbb{R}^m \xrightarrow{\varphi} \mathbb{R}^m.$$

If the method is convergent, then $\exists x : x = \lim_{k \rightarrow \infty} x_k$.

From the computational point of view, this limit cannot be reached, and consequently a termination criterion is used to stop the iterative process, such as

$$\text{if } \|X_k - X_{k-1}\| \leq \varepsilon \|X_k\| \text{ then stop} \quad X_k \in \mathbb{F}^m,$$

where ε is an arbitrary positive value. It is clear that this termination criterion is not satisfactory for two reasons.

If ε is too large then the sequence is broken off before a good approximation to the solution is reached. On the contrary if ε is too small, then many useless iterations are performed, without improving the accuracy of the solution because of round-off error propagation. Moreover each X_k has only a certain number of significant decimal digits. If the ε selected is less than the accuracy of X_k , this termination criterion is no longer meaningful. Two problems then arise.

1. How can the iterative process be stopped correctly?
2. What is the accuracy of the computed solution provided by the computer?

With the use of the CADNA library, thanks to the properties of DSA, it is possible to define new termination criteria, depending on the problem to be solved, which stop the iterative process as soon as a satisfactory computational solution is reached.

Indeed two categories of problems exist:

1. those for which there exists some function which is null at the solution of the problem. The solution of a linear or non-linear system or the search of an optimum for a constrained or non-constrained problem belong to this category;
2. those for which such a function does not exist. Such is the computation of the sum of a series.

For the first category the termination criterion is called ‘optimal termination criterion.’ It acts directly on functions which must be null at the solution of the problem. For example, from the mathematical standpoint if $x_s \in \mathbb{R}^m$ is the solution of a linear system then $A \cdot x_s - B = 0$. The optimal termination criterion consists in stopping the iterative process on the k th iterations if and only if $A * X_k - B = @.0$ (@.0 being the computational zero).

For the second category the usual termination criterion is replaced by

$$\text{if } \|X_k - X_{k-1}\| = @.0 \quad \text{then stop.}$$

With this termination criterion the arbitrary value ε is eliminated.

Example 4. Jacobi’s iterations.

To illustrate the fact that the choice of ε in the classical termination criterion may be difficult and the efficiency of the optimal termination criterion, let us consider the following linear system $AX = B$ of dimension 25 with

$$\begin{aligned} a_{ij} &= \frac{1}{i+j-1} \quad \text{for } i, j = 1, \dots, n \text{ and } i \neq j \\ a_{ii} &= 1. + \sum_{j=1}^{i-1} a_{ij} + \sum_{j=i+1}^n a_{ij} \quad \text{for } i = 1, \dots, n \\ b_i &= \sum_{j=1}^n a_{ij} z_j \quad \text{for } i = 1, \dots, n, \quad \text{with } z_j = 3^{j-1} \times 2^{-10}, \quad j = 1, \dots, n. \end{aligned} \tag{26}$$

As the diagonal of A is dominant Jacobi’s iterations are always convergent.

The exact solution is $x_j = 3^{j-1} \times 2^{-10}$.

System (26) was first solved using standard IEEE 754 double-precision floating-point arithmetic (DPFP) with several values of ε and then with the CADNA double-precision DSA. The results are the following:

- With DPFP and the classical termination criterion with $\varepsilon = 10^{-4}$, the last unknown x_{25} has been computed with the maximum accuracy (15 decimal digits) and the accuracy decreases from component to component until it reaches 3 decimal digits on the first component x_1 .
- With $\varepsilon < 10^{-4}$, the test is never satisfied and the iterations stop on a predefined arbitrary maximum number of iterations. Hence a great number of useless iterations are computed, without improving the accuracy of the solution. For example with $\varepsilon = 10^{-5}$, the process is stopped after 10,000 iterations and the accuracy is identical to the one obtained with $\varepsilon = 10^{-4}$ (see Table 2.4).
- On the contrary with $\varepsilon = 10^{-3}$, the process is stopped too soon (419 iterations) and x_1 and x_{25} are obtained with, respectively, 2 and 14 decimal digits.
- Moreover in all cases with IEEE 754 FP arithmetic the number of significant decimal digits of each unknown cannot be obtained.
- On the contrary with the use of the CADNA library and the optimal termination criterion defined above the process is stopped as soon as a satisfactory solution is obtained, 459 iterations in this case, and the number of exact decimal digits, up to one, of each component is provided.

The number N of decimal digits thus obtained for system (26) with the initialization

$$x_i^0 = 15$$

are given in Table 2.4.

Table 2.4 Number of exact decimal digits in the solution of system (26) with the CADNA library

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	23	25
N	3	4	4	5	6	6	6	7	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15	15	15

In fact, as shown in [39] the optimal termination criterion which consists in testing the residual and the usual criterion which consists in testing the difference between two iterates are closely connected in the case of Jacobi's method because

$$X_{k+1} - X_k = D^{-1}(B - A X_k),$$

matrix D being the diagonal of A .

This is perfectly verified with the CADNA library. In fact when the termination criterion is the stochastic equality of two successive vector iterates, the process is stopped at the 460th iteration and the accuracy of the solution is also the same as the one reported in Table 2.4.

2.9.3 Approximate Methods

From the mathematical standpoint, these methods provide only an approximation of the solution. This category contains, e.g., numerical computation of derivatives, numerical integration, and numerical solution of differential or partial differential equations.

When these methods are run on a computer, they always provide a solution containing an error e_g , which is a combination of the method error e_m inherent in the employed method and the error due to the propagation of round-off errors called computation error e_c .

It is well known that the method error e_m is an increasing function of the discrete step size h . On the contrary the computation error e_c is an increasing function of the inverse $\frac{1}{h}$ of the step size. This means that e_m and e_c act in the opposite way and consequently the global error e_g is a function which has a minimum for some value of h . Thus the best approximation of the solution that can be obtained on a computer corresponds to an optimal discrete step size h^* , such that $de_g/dh = 0$. Obviously, it is impossible to establish a general methodology to estimate h^* , because the method error e_m is specific to the method. Yet most of the time for a specific method, e_m can be estimated. Furthermore, e_c can be estimated using the CADNA library. Then in many cases, it is possible to estimate h^* [17].

To illustrate this, let us consider the following example, which is a simple solution of a differential equation using Euler's method.

Example 5. *The differential equation is*

$$y' = e^x y + xy - (x + 1)e^{-x} - 1. \quad (27)$$

With the initial condition $y(0) = 1$, the exact solution is $y(x) = e^{-x}$.

The computation of the optimal step size for each interval $[x_k, x_{k+1}]$ requires three phases:

1. *the estimation of the round-off error e_c ,*
2. *the evaluation of the truncation error (method error) e_m , and*
3. *the computation of the optimal step size.*

The estimation of the round-off error e_c is obtained using the CADNA library. Indeed a special function called nb-significant digits(x) of this library returns the number of significant digits in a stochastic argument x. This number is an integer n obtained from equation (12) rounded down. The estimation of the round-off error is then computed by equation (28):

$$e_c = 10^{-(n+1)}. \quad (28)$$

The estimation of the truncation error e_m at each step for Euler's method is well known and is given by

$$e_m = 2|y_1 - y_2|. \quad (29)$$

Table 2.5 Solution of equation (27) with different step size

x	Exact solution y^*	$h = 10^{-1}$	$h = 10^{-3}$	$h = 10^{-6}$	Optimal h
0.0	1.0	1.0	1.0	1.0	1.0
0.1	0.905	0.900	0.905	0.905	0.905
0.2	0.819	0.809	0.819	0.818	0.819
0.3	0.741	0.726	0.741	0.740	0.741
0.4	0.670	0.649	0.670	0.670	0.670
0.5	0.607	0.578	0.606	0.606	0.607
0.6	0.549	0.511	0.548	0.550	0.548
0.7	0.497	0.447	0.496	0.499	0.496
0.8	0.449	0.384	0.448	0.453	0.447
0.9	0.407	0.320	0.405	0.413	0.404
1.0	0.368	0.250	0.366	0.380	0.366

y_1 is the value of $y(x_k + h_k)$ integrated over the interval $[x_k, x_k + h_k]$ with step size h_k , while y_2 is the value of $y(x_k + h_k)$ integrated over the same interval $[x_k, x_k + h_k]$ with step size $\frac{1}{2}h_k$. Of course, e_m may not be less than e_c because it is also computed with the same computer.

The optimal step size h^* can then be obtained with a simple minimization method.

The results obtained in single precision using CADNA are presented in Table 2.5.

From the results of Table 2.5 it clearly appears that if the step size is too large ($h = 0.1$) or too small ($h = 10^{-6}$), only one or two decimal digits are obtained in the solution, while with the optimal step size the solution is computed with two or three exact digits.

Here, a very simple method to estimate the optimal step size has been presented, but more sophisticated methods have also been developed in [27].

2.10 Can the CADNA Library Fail?

To answer to this question, imagine a computation such that only one or two rounding errors are the dominant contribution to the final error. This is the case in example 1 and example 2. Concerning example 1 which has been specially created to jeopardize the stochastic approach of FP computation, it can be shown experimentally that as the number of samples N increases the percentage of failure decreases. This percentage which is presented in Table 2.6 is in total agreement with the approximation of the mean value and standard deviation of an unknown Gaussian distribution by those of empirical values which is used in equation (12).

Concerning example 2, during the Gaussian elimination there is no round-off error propagation except for the last pivot $a_{n,n}$, because all the other results are integer values which are exact FP values.

It is exactly the same thing for the computation of the n elements of the right-hand side \mathcal{B} which are also exact FP values.

The values of the last pivot $a_{n,n}$ and b_n are

$$a_{n,n} = \alpha + 2^{n-1} - 1 \quad \text{and} \quad b_n = 2^{n-1}.$$

Table 2.6 Percentage of failure as a function of the number N of samples in example 1

N	3	4	5	7	10
%	10	5	3	1	0

Table 2.7 Percentage of failures as a function of n and N in example 2

n	$N = 3$	$N = 4$	$N = 5$	$N = 6$	$N = 7$
5	5	3	2	0	0
25	10	5	3	1	0
45	10	6	4	3	0

The larger the n , the closer x_n is to 1.

Furthermore, $x_n = b_n/a_{n,n}$ and $x_{n-1} = a_{n-1,n+1}(1 - x_n)$ because $a_{n-1,n+1} = a_{n-1,n}$.

With the CADNA library if a particular combination of random roundings makes the N elements of x_n equal, then the round-off error on x_n has vanished and the resulting $x_{n-1}, x_{n-2}, \dots, x_1$ are false values and CADNA does not detect the failure.

Table 2.7 presents the percentages of failures with respect to the dimension n and the number of samples N .

Tables 2.6 and 2.7 show that for computations in which only one rounding error is the dominant contribution to the final error, N must be greater than 3 so that there is no failure. Then the choice of $N = 3$ has to be explained. Indeed this is because for normal computing, several rounding errors contribute to the final error. The CADNA library uses $N = 3$ and a probability of 95% for estimating the number of significant decimal digits. However it has been shown that if the user accepts an error of one unit in the number of significant decimal digits, then the probability of estimating it up to 1 is 99.94%.

2.11 Conclusion

In this chapter the CESTAC method, which is a stochastic method for estimating the error propagation from both the FP arithmetic and the uncertainties of data issuing from sensors, has been presented. With this method the number (up to one) of significant digits on a computed numerical result can be evaluated. However this type of method was incorrectly implemented by S.G. Popovitch with his Prosolver software. In this software the N runs are not synchronized and thus the control of numerical anomalies cannot be performed at the level of each elementary operators. Thus many numerical instabilities will not be detected. It is for this reason that examples to expose the weakness of this software were proposed in [29] and [30]. Later using the ideas developed for the CESTAC method, the Monte Carlo method and the software Wonglediff were also proposed in [31–33] with the same drawbacks as those of Prosolver.

Indeed to be effective, the stochastic method requires that an eventual anomaly or instability is checked at the level of each elementary operation, i.e., an arithmetic operation, an order relation, or a branching. This requires that the N samples representing the result of an operation are obtained synchronously and not by running the same code N times in sequence. In other words, these methods are reliable if and only if they are implemented in the scope of granular computing and follow the model of DSA.

The theory of stochastic arithmetic, which is proposed in this chapter, provides a model for computation on approximate data. In this sense it aims at the same target as interval arithmetic except that the operands and operators are different. In the scope of granular computing the granules of stochastic arithmetic are independent Gaussian variables and the tools are the classical operators on Gaussian functions. These operators induce many algebraic structures and some of them have been presented.

The theory of DSA provides a model in which granules are composed of an N -tuple of N samples of the same mathematical result of an arithmetical operator implemented in FP arithmetic. These samples differ from each other because the data are imprecise and because of different rounding. The operator working on these granules is an FP operator corresponding to the exact arithmetical operator which is performed N times in a synchronous way with random rounding. Thus the result is also a granule. This granule is called a discrete stochastic number. It has been shown that the DSA operating on discrete stochastic numbers has many properties (but not all) of real numbers; in particular, the notion of stochastic zero has been defined.

The CADNA library implements DSA and is able during the run of a code to analyze the effect of uncertainties of the data and of round-off error propagation on the result of each arithmetical operation. Thus any anomaly can be detected at this level. When such an anomaly is detected, a warning is written in a special file provided for the user. Hence, because of its correct implementation in the scope of granular computing the CADNA library does not fail when tested with the previously cited examples. This library has been successfully used for solving many problems belonging to the three categories of numerical methods.

In the field of linear algebra it has been used for the solution of linear systems using Gaussian elimination, GMRES [40], Orthomin(k) [41], and CGS [39] algorithms. It has enabled the optimization of collocation algorithms [27] and quadrature algorithms [42, 43]. It has also been used for checking the reliability of numerical methods in most fields of research in applied numerical mathematics: geology [44, 45], acoustics [46], solid mechanics [47], engine combustion [48], and atomic physics [49]. In all cases the CADNA library has always been successful.

Moreover, many future developments and applications of the CESTAC method, DSA, and CADNA are now possible particularly in the production of self-validated libraries requiring no programming effort in every domain of numerical analysis.

References

- [1] J.G. Rokne. Interval arithmetic and interval analysis: an introduction. In: *Granular Computing: An Emerging Paradigm*. Physica-Verlag GmbH, Heidelberg, Germany, 2001, pp. 1–22.
- [2] J.M. Chesneaux. Study of the computing accuracy by using a probabilistic approach. In: C. Ullrich (ed.), *Contributions to Computer Arithmetic and Self-validating Methods*, IMACS, NJ, 1990, pp. 19–30.
- [3] J.M. Chesneaux. *Etude théorique et implémentation en ADA de la méthode CESTAC*. Thesis. Paris VI University, Paris, 1988.
- [4] R.W. Hamming. On the distribution of numbers. *Bell Syst. Tech. J.* 49 (1970) 1609–1625.
- [5] D. Knuth, *The Art of Computer Programming* 2. Addison-Wesley, Reading, MA, 1969.
- [6] A. Feldstein and R. Goodman. Convergence estimates for the distribution of trailing digits. *J. ACM*. 23 (1976) 287–297.
- [7] M. La Porte and J. Vignes. Evaluation statistique des erreurs numériques sur ordinateur. *Proc. Canadian Comp. Conf.* (1972) 414201–414213.
- [8] M. La Porte and J. Vignes. Etude statistique des erreurs dans l’arithmétique des ordinateurs, application au contrôle des résultats d’algorithmes numériques. *Numer. Math.* 23 (1974) 63–72.
- [9] M. La Porte and J. Vignes. Méthode numérique de détection de la singularité d’une matrice. *Numer. Math.* 23 (1974) 73–82.
- [10] M. La Porte and J. Vignes. Evaluation de l’incertitude sur la solution d’un système linéaire. *Numer. Math.* 24 (1975) 39–47.
- [11] J. Vignes and M. La Porte. Error analysis in computing. In: *Information Processing* 74, North-Holland, Amsterdam, 1974, pp. 610–614.
- [12] P. Bois, and J. Vignes. An algorithm for automatic round-off error analysis in discrete linear transforms. *Intern. J. Comput. Math.* 12 (1982) 161–171.
- [13] J.M. Chesneaux and J. Vignes. Sur la robustesse de la méthode CESTAC. *C.R. Acad. Sci. Paris* 307 (1988) 855–860.
- [14] J. Vignes. New methods for evaluating the validity of the results of mathematical computations. *Math. Comput. Simul.* 20 (4) (1978) 227–248.
- [15] J. Vignes. Zéro mathématique et zéro informatique. *C.A. Acad. Sci., Paris* 303 (1) (1986) 997–1000; *La Vie des Sciences*, 4, (1) (1987) 1–13.
- [16] J. Vignes. Discrete stochastic arithmetic for validating results of numerical software. *Numer. Algorithms* 37 (2004) 377–390.
- [17] J. Vignes. A stochastic arithmetic for reliable scientific computation. *Math. Comput. Simul.* 35 (1993) 233–261.
- [18] J. Vignes. Review on stochastic approach to round-off error analysis and its applications. *Math. Comput. Simul.* 30 (6) (1988) 481–491.
- [19] J. Vignes and R. Alt. An efficient stochastic method for round-off error analysis, In: *Accurate Scientific Computations, L.N.C.S 235*, Springer-Verlag, New York, 1985, pp. 183–205.
- [20] J.M. Chesneaux. *L’Arithmétique stochastique et le logiciel CADNA*. Habilitation à diriger les recherches. Université Pierre et Marie Curie, Paris, 1995.

- [21] J.M. Chesneaux and J. Vignes. Les fondements de l'arithmétique stochastique. *C.R. Acad. Sci., Paris* 315 (1992) 1435–1440.
- [22] J.M. Chesneaux. The equality relation in scientific computing. *Numer. Algorithms* 7 (1994) 129–143.
- [23] R. Alt, and S. Markov. On the algebraic properties of stochastic arithmetic, comparison to interval arithmetic. In: W. Kraemer and J. Wolff von Gudenberg (eds), *Scientific Computing, Validated Numerics, Interval Methods*. Kluwer, Dordrecht, 2001, pp. 331–341.
- [24] R. Alt, J.L. Lamotte and S. Markov. On the numerical solution to linear problems using stochastic arithmetic. In: Proceedings of the 2006 ACM Symposium on Applied Computing, Dijon France, (2006), pp. 1655–1659.
- [25] S. Markov, R. Alt, and J.-L. Lamotte. Stochastic arithmetic: S-spaces and some applications, *Numer. Algorithms* 37 (1–4) (2004) 275–284.
- [26] S. Markov and R. Alt. Stochastic arithmetic: addition and multiplication by scalars. *Appl. Numer. Math.* 50 (2004) 475–488.
- [27] R. Alt and J. Vignes. Validation of results of collocation methods for ODEs with the CADNA library. *Appl. Numer. Math.* 20 (1996) 1–21.
- [28] S.G. Popovitch. *Prosolver, La Commande Electronique, France*. Ashton-Tate, Torrance, CA, 1987.
- [29] W. Kahan. The improbability of probabilistic error analyses. In: *UCB Statistics Colloquium*. Evans Hall, University of California, Berkeley, 1996. <http://www.ca.berkeley.edu/wkahan/improber.ps>.
- [30] W. Kahan. How futile are mindless assessments of round-off in floating point computation. *Householder Symposium XVI*, 2005, <http://www.cs.berkeley.edu/wkahan/Mindless.pdf>.
- [31] D.S. Parker. *Monte Carlo Arithmetic: Exploiting Randomness in Floating Point Arithmetic. Report of computer science department, UCLA, Los Angeles*, March, 30, 1997.
- [32] D.S. Parker, B. Pierce, and D.R. Eggert. Monte Carlo arithmetic: how to gamble with floating point and win. *Comput. Sci. Eng.* (2000) 58–68.
- [33] P.R. Eggert, and D.S. Parker. Perturbing and evaluating numerical programs without recompilation – the wonglediff way. *Softw. Pract. Exp.* 35 (2005) 313–322.
- [34] J. Vignes. A stochastic approach to the analysis of round-off error propagation: a survey of the CESTAC method. In: *Proceedings of the 2nd Real Numbers and Computer Conference*, Marseille, France, 1996, pp. 233–251.
- [35] M. Pichat and J. Vignes. Ingénierie du contrôle de la précision des calculs sur ordinateur. *Technip*, Paris (1993).
- [36] Cadna user's guide, <http://www.lip6.fr/cadna>.
- [37] M. Daumas, and J.M. Muller. *Qualité des Calculs sur Ordinateur*. Masson, Paris, 1997.
- [38] J.R. Westlake. *Handbook of Numerical Matrix Inversion and Solution of Linear Equations*. Wiley. New York, 1968.
- [39] J.M. Chesneaux, and A. Matos. Breakdown and near breakdown control in the CGS algorithm using stochastic arithmetic. *Numer. Algorithms* 11 (1996) 99–116.
- [40] F. Toutounian. The use of the CADNA library for validating the numerical results of the hybrid GMRES algorithm. *Appl. Numer. Math.* 23 (1997) 275–289.
- [41] F. Toutounian. The stable $A^T A$ -orthogonal s -step Orthomin(k) algorithm with the CADNA library. *Numer. Algorithms* 17 (1998) 105–119.
- [42] F. Jézéquel and J.M. Chesneaux. Computation of an infinite integral using Romberg's method. *Numer. Algorithms* 36 (2004) 265–283.
- [43] F. Jézéquel. Dynamical control of converging sequences computation. *Appl. Numer. Math.* 50 (2004) 147–164.
- [44] F. Delay and J.-L. Lamotte. Numerical simulations of geological reservoirs: improving their conditioning through the use of entropy. *Math. Comput. Simul.* 52 (2000) 311–320.
- [45] J.-L. Lamotte and F. Delay. On the stability of the 2D interpolation algorithms with uncertain data. *Math. Comput. Simul.* 43 (1997) 183–190.
- [46] J.M. Chesneaux, and A. Wirgin. Reflection from a corrugated surface revisited. *J. Acoust. Soc.* 96(2 pt. 1) (1993) 1116–1129.
- [47] N.C. Albertsen, J.-M. Chesneaux, S. Christiansen, and A. Wirgin. Evaluation of round-off error by interval and stochastic arithmetic methods in a numerical application of the Rayleigh theory to the study of scattering from an uneven boundary. In: G. Cohen (ed), *Proceedings of the Third International Conference on the Mathematical and Numerical Aspects of Wave Propagation*, SIAM, Philadelphia, 1995, pp. 338–346.
- [48] S. Guilain, and J. Vignes. Validation of numerical software results. Application to the computation of apparent heat release in direct injection diesel engine. *Math. Comput. Simul.* 37 (1994) 73–92.
- [49] N.S. Scott, F. Jézéquel, C. Denis, and J.M. Chesneaux. Numerical 'health check' for scientific codes: the CADNA approach. *Comput. Phys. Commun.* 176 (8) (2007) 507–521.

3

Fundamentals of Interval Analysis and Linkages to Fuzzy Set Theory

Weldon A. Lodwick

3.1 Introduction

The granular computing of interest to this chapter processes entities (granules) whose representation is other than real numbers. Processing with real numbers leads to determinism and mathematical analysis. The granules of interest to this chapter are intervals and fuzzy sets. The point of departure for this chapter is interval analysis. It is noted that parts of what is presented can be found in [1–4].

An interval $[a, b]$ on the real-number line, with the usual meaning of the order relation \leq , is the set of all real numbers $\{x : a \leq x \leq b\}$. The next section develops a natural definition of arithmetic for intervals, represented as pairs of real numbers (their endpoints), that follow from elementary properties of the relation \leq . However, intervals on the real line may be viewed as having a dual nature, both as a *set* (of real numbers) and as a new kind of *number* represented by pairs of real numbers. Interval arithmetic derived from both these two natures of the granules (a new number and a set) leads to a relatively new type of mathematical analysis called interval analysis [5]. The point of view of intervals as a set is also discussed in [6].

The logic of interval analysis that follows is one of certain containment. The sum of two intervals certainly contains the sums of all pairs of real numbers, one from each of the intervals. We can also compute intersections and unions of intervals. For a given interval $[a, b]$ and a given real number x , the statement $x \in [a, b]$ is either true or false. Moreover, for two intervals A_1 and A_2 , if we know that $x \in A_1$ and $x \in A_2$, then we also know that $x \in A_1 \cap A_2$. In interval arithmetic, and the interval analysis developed from it, a measure of possibility or probability is not assigned to parts of an interval. A number x either is in an interval A or is not.

The introduction of a distribution that represents the possible (probable) spread of uncertainty within an interval, and using level sets, integrals, or other measures, connects interval arithmetic to the other granule of interest – fuzzy sets. Computing with intervals, as parts of the total support, whether finite or infinite, of possibility or probability distributions, can produce intervals representing enclosures of mappings of input intervals. It is a separate problem to assign possibility or probability measures to the interval results, according to assumptions about measure on the input intervals, and the general theories of possibility or probability distributions.

The certainty offered by interval methods refers only to certainty in the sense of knowledge about solutions to mathematical equations. For example, using interval computation we may find that the range of values of a real-valued function f of a single real variable x is contained in an interval $[c, d]$ when $x \in [a, b]$. We can denote this by $f([a, b]) \subseteq [c, d]$. Suppose f is continuous, and we are interested in finding an interval containing a zero of f , which is a solution to the equation $f(x) = 0$. If $0 \notin [c, d]$, which can be tested using $0 < c$ or $d < 0$, then it is known that there is no solution in $[a, b]$. On the other hand, if $0 \in [c, d]$, then it is *possible* that f has a zero in $[a, b]$. It is not certain because it is in the nature of interval computation that it cannot generally find exact ranges of values. Thus we may have $0 \in [c, d]$ but $0 \notin f([a, b])$. By using continuous contraction mappings or other techniques of interval analysis, we may be able to *prove* that there is a solution in $[a, b]$.

Interval analysis and fuzzy set theory, as active fields of research and application, are relatively new mathematical disciplines, receiving the impetus that defined them as separate fields of study in 1959 and 1965, respectively, with R.E. Moore's technical reports on interval analysis and his Ph.D. thesis (see [7–10]) and L. Zadeh's seminal papers on fuzzy set theory (see [11–13]). The connection between interval analysis and possibility theory is evident in the mathematics of uncertainty. The first to recognize the potential of interval analysis in dealing with fuzzy set and possibility theory seems to have been D. Dubois and H. Prade (see [14,15]). The theory of interval analysis models, among other things, the uncertainty arising from numerical computation which can be considered as a source of ambiguity. Fuzzy set theory and possibility theory model, among other things, the uncertainty of vagueness and ambiguity arising from the transitional nature of entities and a lack of information respectively.

Interval analysis developed as part of the then emergent field of numerical analysis initially had three directions of interest to this chapter:

1. **Computational error analysis** (automatically computed error including rounding).
2. **Verified computing** which R.E. Moore called *range arithmetic* dealt with guaranteed enclosures (including rounding) of the minimum and maximum of a continuous function over interval domains. Later Aberth [16] developed an interval arithmetic he called range arithmetic, which is different than how R.E. Moore used the phrase (which he used only once).
3. The derivation of the underlying algebraic structure of floating-point numbers called **computer algebra**.

Fuzzy sets have developed two directions of interest to this chapter:

1. **Fuzzy interval analysis** (see [17–21]), and
2. **Possibility theory** (see [22, 23]).

Although the two fields can be thought of as having come from a common root, interval analysis and fuzzy set theory are independent fields whose cross fertilization has been a relatively recent phenomenon (see [2, 3, 18, 19]). All interval, fuzzy/possibilistic and interval-valued probabilistic analyses that follow are over sets of real numbers, i.e., real-valued intervals and real-valued distributions (in the case of fuzzy membership functions or possibilistic distributions). Moreover, when the word ‘box’ is used in the context of intervals it is understood to be in \mathbb{R} if the box refers to an interval $[a, b]$ and in \mathbb{R}^n if the box is a rectangular n -dimensional hyperrectangle $[a_1, b_1] \times \cdots \times [a_n, b_n], a_i, b_i \in \mathbb{R} \quad i = 1, \dots, n$.

3.2 The Central Issues

The underlying mathematical theory from which interval and distribution analysis arise is set functions particularized to intervals and fuzzy sets and associated upper and lower approximations of the resultants. Therefore, the central issues common to both interval and fuzzy analyses must include the *interval/fuzzy extension principles*, *interval/fuzzy arithmetic*, and *enclosure/verification*.

Extension principles of R.E. Moore [9] and L. Zadeh [11] are directly related to an earlier development and more general set function theory (see, e.g., [24, 25]). A treatment of set-valued functions is [26].

Of course, set-valued functions extend real-valued functions to functions on intervals and fuzzy sets. The extension principle used in interval analysis is called the *united extension* (see [9,10]). In fuzzy set theory it is called simply the *extension principle* (see [11]). Since arithmetic operations are continuous real-valued functions (excluding division by zero), the extension principles may be used (and have been) to define interval and fuzzy arithmetic.

Enclosure, for this exposition, means approximations that produce upper and lower values (interval or functional envelope, depending on the context) to the theoretical solution which lies between the upper and lower values or functions. Efficient methods to compute upper and lower approximations are desired and necessary in practice. When enclosure is part of mathematical problem solving, it is called *verification*, formally defined in what follows.

The point of view that the extension principle is an important thread which can be used to relate and understand the various principles of uncertainty that is of interest to this chapter (interval, fuzzy, and possibility) leads to a direct relationship between the associated arithmetic. Moreover, upper and lower approximation pairs for fuzzy sets allow for simpler computation using *min/max* arithmetic and lead to enclosures with careful implementation. Arithmetic is a gateway to mathematical analysis. The importance of this is that in the computational methods requisite in doing mathematics (arithmetic and analysis) with various types of uncertainty, the underlying approach to computing is the extension principle.

This chapter, therefore, considers three associated themes:

1. **Extension principles,**
2. **Arithmetic** (derived from rules on interval endpoints and extension principles), and
3. **Enclosure and verification.**

3.2.1 Extension Principles

The extension principle is key because it defines how real-valued expressions are represented in the context of intervals and fuzzy sets. One can view the extension principle as one of the main unifying concepts between interval analysis and fuzzy set theory. Moreover, the extension principle is used here to define how to do arithmetic on intervals, fuzzy sets, and, more generally, distributions which will not be discussed (see [4] for an extended discussion). Arithmetic can also be defined using rules on interval endpoints. Both the extension and the rules on interval endpoint approaches are discussed.

All extension principles associated with intervals and fuzzy sets may be thought of coming from set-valued mappings or graphs. Generally, an extension principle defines how to obtain functions whose domains are sets. It is clear how to accomplish this for real numbers. It is more complex for sets since how to obtain resultant well-defined entities must be defined.

Set-valued maps have a very long history in mathematics. Relatively recently, Strother's 1952 Ph.D. thesis [24] and two papers [25, 27] define the *united extension* for set-valued functions for domains possessing specific topological structures. R.E. Moore applied Strother's united extension to intervals. In doing so, he had to show that the topological structures on intervals were one of those that Strother developed. Having done this, Moore retains the name *united extension* as the extension principle particularized to intervals. In fact, Strother is a coauthor of the technical report that first uses the set-valued extension principle on intervals. That is, Moore's united extension (the interval extension principle) is a set-valued function whose domain is the set of intervals and of course the range is an interval for those underlying functions that are continuous.

Zadeh's extension principle (see [11]) is also the way functions of fuzzy sets are derived from real-valued functions. It expresses, in essence, how to compute with fuzzy sets. That is, Zadeh's extension principle can be thought of as a set-valued function where the domain elements are fuzzy sets and of course the range values are fuzzy sets for the appropriate maps, called *membership functions*, which are defined below. The extension principle was generalized and made more specific to what are now called *fuzzy numbers* or *fuzzy intervals* by various researchers beginning with H. Nguyen [28].

3.2.2 Arithmetic

Interval arithmetic is central to fuzzy arithmetic and can be derived axiomatically or from Moore's *united extension*. Of special interest is the latter approach especially in deriving a constrained interval arithmetic (see Section 3.3.2.2) which will have implications for fuzzy arithmetic. There are two direct precursors of Moore's development of interval arithmetic in 1959, M. Warmus (1956 – [29]) and T. Sunaga (1958 – [30]). Moore's initial work references and extends in significant ways Sunaga's work in that he develops computational methods, incorporates computer rounding, develops for the first time automatic numerical error analysis (gets the computer to calculate round-off, numerical truncation, and numerical method error estimations), and extends interval arithmetic to interval analysis. Analysis on intervals, since they are sets, require set-valued functions, limits, integration, and differentiation theory. This is done via the *united extension* (see [5]).

The rules of interval arithmetic as articulated by Warmus [29], Sunaga [30], and Moore [7] are as follows. It is noted that Warmus' notation is different but the operations are the same.

1. Addition:

$$[a, b] + [c, d] = [a + c, b + d] \quad (1)$$

2. Subtraction:

$$[a, b] - [c, d] = [a - d, b - c] \quad (2)$$

3. Multiplication:

$$[a, b] \times [c, d] = [\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}] \quad (3)$$

4. Division:

$$[a, b] \div [c, d] = [a, b] \times [1/d, 1/c] \quad \text{where } 0 \notin [c, d]. \quad (4)$$

There is an extended interval arithmetic that incorporates the case where $0 \in [c, d]$ for division (see [31, 32]). Moreover, there are a variety of ways to approach interval arithmetic, e.g., see [33–36]. In fuzzy arithmetic, the axioms of interval arithmetic apply to each α -cut of a fuzzy set membership function as long as the entity is a *fuzzy number* or *fuzzy interval*. An interval arithmetic base on axioms 1–4 above is called here *interval arithmetic* or *traditional interval arithmetic* when there is a need to distinguish it from what is developed in the sequel using a set-valued approach to interval arithmetic that is called *constraint interval arithmetic*.

The implementation of interval arithmetic to the computer in which the concern is to account for all errors (numerical and truncation error) is called *rounded interval arithmetic* (see [7, 8]). U. Kulisch in [37] studied rounded interval arithmetic and uncovered the resultant algebraic structure, called a *ringoid*, with W.L. Miranker (see [38, 39]). While specialized extended languages (Pascal-XSC and C-XSC) and chips were developed for interval and rounded interval data types incorporating the ideas set out by Moore, Kulisch, and Miranker (among other researchers), the most successful rounded interval tool is undoubtedly *INTLAB*, a software package that runs in conjunction with *MATLAB* with embedded interval arithmetic, rounded interval arithmetic, and some interval analytic methods, in particular computational linear algebraic methods (downloadable from www.ti3.tu-harburg.de/~rump/intlab).

3.2.3 Enclosure and Verification

The approach of Archimedes ([40, 41]) to the computation of the circumference of a circle using outer circumscribed and inner inscribed regular polygons whose perimeter is a straightforward calculation is an enclosure and verification method, perhaps the first one. The essential part of enclosure and verification

is that a solution is mathematically provable to exist (perhaps is unique) and lies between the *computed* upper and lower bound values (real numbers for our purposes). That is, verification guarantees that the solution exists (and perhaps is unique) in a mathematical sense. Enclosure is the computed upper and lower bound containing the solution. Verification in the case of Archimedes computation of the circumference of a circle is the geometrical fact (theorem) that the perimeter of the circumscribed regular polygon is greater than the circumference of a circle and that the inscribed regular polygon has a perimeter less than that of the circumference of a circle. Moreover, the two outer and inner partitions converge to the ‘perimeter’ of the circle (one from above and the other from below), which Archimedes took as obvious by construction. Often, to verify the existence of solutions in mathematical analysis, fixed-point theorems (contractive mapping theorems for example) are used. These theorems are employed from the point of view of verification of hypotheses so that interval analysis will mathematically calculate guaranteed bounds on the Lipschitz constant, e.g., accounting for *all* numerical and truncation errors which if less than 1 means that the mapping is contractive and hence a *computed* solution will exist.

The methods to compute upper and lower bounds in a mathematically correct way on a computer must account for numerical and computer truncation error. This is one of the core research areas of interval mathematics. One of the primary applications of interval analysis is to enclosure methods for verification and as will be seen the equivalent for fuzzy sets and possibility theory is the computation of functional envelopes (interval-valued probability [42]). Interval verification methods obtain interval enclosures containing the solution(s) within its bounds. In interval analysis, verification means that existence is mathematically verified and guaranteed bounds on solutions are given. When possible and/or relevant, uniqueness is mathematically determined. Thus, verification in the context of a computational process that uses intervals for a given problem means that a solution, say x , is verified (mathematically) to **exist** and the computed solution is returned with lower and upper **bounds**, a and b , such that the solution, shown to exist, is guaranteed to lie between the provided bounds, i.e., $a \leq x \leq b$. **Uniqueness** is determined when possible or desirable.

Although, not often thought of in these terms, possibility/necessity pairs when carefully constructed enclose a resultant distribution of the solution to a problem and are functional enclosures. Verification in the context of distributions is understood to be the construction of lower and upper functions, $g(x)$ and $h(x)$, to a given function $f(x)$, such that $g(x) \leq f(x) \leq h(x)$. We wish to do this not only when x is a real number or vector, but also when x is a (vectors of) distribution such as random variables, intervals, fuzzy sets, and/or possibilities. When $f(x)$ is a complex expression, this is an especially difficult problem.

3.3 Interval Analysis

Intervals are sets, and they are a (new type of) number. This dual role is exploited in the arithmetic and analysis. Professor A. Neumaier [43, p. 1] states, ‘Interval arithmetic is an elegant tool for practical work with inequalities, approximate numbers, error bounds, and more generally with certain convex and bounded sets.’ And he goes on to say that intervals arise naturally in

1. Physical measurements
2. Truncation error – the representation of an infinite process by a finite one
 - (a) Representation of numbers by finite expansions
 - (b) Finite representation of limits and iterations
3. Numerical approximations
4. Verification of monotonicity and convexity
5. Verification of the hypotheses of fixed-point theorems – the contraction mapping theorem or Brouwer’s fixed-point theorem for example
6. Sensitivity analysis, especially as it is applied to robotics
7. Tolerance problems

Interval arithmetic and analytic methods have been used to solve an impressive array of problems given that these methods capture error (modeling, round-off, and truncation) so that rigorous accounting of error together with the contraction mapping theorem or Brouwer’s fixed-point theorem allow for computer

verification of existence, uniqueness, and enclosure. In particular, W. Tucker [44], using interval analysis, solved a long outstanding problem (Smale's fourteenth conjecture [45]) by showing that the Lorenz equations do possess a strange attractor.

Professor B. Davies [46, p. 1352] observes,

Controlled numerical calculations are also playing an essential role as intrinsic parts of papers in various areas of pure mathematics. In some areas of non-linear PDE, rigorous computer-assisted proofs of the existence of solutions have been provided. . . . These use *interval arithmetic* to control the rounding errors in calculations that are conceptually completely conventional.

Another long-standing problem, Kepler's conjecture about the densest arrangement of spheres in space, was solved by T. Hales [47] using interval arithmetic. There were ten problems posed by Nick Trefethen in January/February, 2002 *SIAM News*, each of which had a real-number solution and the objective was to obtain a ten-digit solution to each of the problems. The book [48] documents not only the correct solutions but the analysis behind the problems. One of the authors, S. Wagon, in a personal communication indicated that

[i]ntervals were extremely useful in several spots. In Problem 2 intervals could be used to solve it by using smaller and smaller starting interval until success is reached. In Problem 4 intervals were used in designing an optimization algorithm to solve it by subdividing. Moreover, for Problems 2, 4, 7 and 9, intervals yield proofs that the digits are correct.

Chapter 4 of [48] contains an exposition of interval optimization.

Robust stability analysis for robots performed with the aid of a computer that is mathematically verifiable uses interval analysis methods (see [49–51]).

There are excellent introductions to interval analysis beginning with R.E. Moore's book [5] (also see other texts listed in the references). A more recent introduction can be found in [52] and downloaded from http://www.eng.mu.edu/corlissg/PARA04/READ_ME.html. Moreover, there are introductions that can be downloaded from the interval analysis Web site (<http://www.cs.utep.edu/interval-comp>).

3.3.1 Interval Extension Principle

Moore's three technical reports [7–9] recognized that the extension principle is a key concept. Interval arithmetic, rounded interval arithmetic, and computing range of functions can be derived from interval extensions. At issue is how to compute ranges of set-valued functions. This requires continuity and compactness over interval functions, which in turn needs well-defined extension principles.

Moore in [9] uses for the first time in an explicit way the extension principle for intervals called the *united extension*, which particularizes [24, 25] set-valued extensions to sets that are intervals. If $f : X \rightarrow Y$ is an arbitrary mapping from an arbitrary set X into an arbitrary set Y , the united extension of f to $S(X)$, denoted F , is defined as follows (see [9]):

$$F : S(X) \rightarrow S(Y), \text{ where}$$

$$F(A) = \{f(a) \mid \forall a \in A, A \in S(X)\}, \text{ in particular}$$

$$F(\{x\}) = \{f(x) \mid x \in \{x\}, \{x\} \in S(X)\}. \text{ Thus,}$$

$$F(A) = \bigcap_{a \in A} \{f(a)\}.$$

This definition, as we shall see, is quite similar to the fuzzy extension principle of Zadeh, where the union is replace by the supremum which is a fuzzy union.

Theorem 1. *Let X and Y be compact Hausdorff spaces and $f : X \rightarrow Y$ continuous. Then the united extension of f , F , is continuous. Moreover, F is closed (see [25]).*

The results of interest associated with the united extension for intervals are the following (see [5]):

1. **Isotone property:** A mapping f from partially ordered set (X, r_X) into another (Y, r_Y) , where r_X and r_Y are relations, is called *isotone* if $x \ r_X \ y$ implies $f(x) \ r_Y \ f(y)$. In particular, the united extension is isotone with respect to intervals and the relation \subseteq . That is, for $A, B \in S([X])$, $A \subseteq B$, then $F(A) \subseteq F(B)$.
2. **The Knaster–Tarski theorem (1927):** An isotone mapping of a complete lattice into itself has at least one fixed point.

Recall that $(S([\mathbb{R}]), \subseteq)$ is a complete lattice. Considering the united extension $F : S([\mathbb{R}]) \rightarrow S([\mathbb{R}])$, the Knaster–Tarski theorem implies that F has at least one fixed ‘point’ (set) in $S([\mathbb{R}])$, which may be the empty set. However, this result has an important numerical consequence. Consider the sequence $\{X_n\}$ in $S(X)$ defined by

$$X_{n+1} = F(X_n).$$

Since

$$X_1 \subseteq F(X_0) \subseteq X_0,$$

by induction,

$$X_{n+1} \subseteq X_n.$$

Considering

$$Y = \bigcap_{n=0}^{\infty} X_n,$$

the following is true (see [9]). If $x = f(x)$ is any fixed point of f in X , then $x \in X_n$ for all $n = 0, 1, 2, \dots$ and so $x \in Y$ and

$$x \in F(Y) \subseteq Y.$$

Thus X_n , Y , and $F(Y)$ contain all the fixed points f in X . If Y and/or $F(Y)$ is empty, then there are no fixed points of f in X . Newton’s method is a fixed-point method, so that the above theorem pertains to a large class of problems. Moreover, these enclosures lead to computational validated solutions when implemented on a computer with rounded arithmetic.

3.3.2 Interval Arithmetic

Interval arithmetic was defined by R.C. Young [53] in 1931, P.S. Dwyer [54] in 1951, M. Warmus [29] in 1956, and then independently by T. Sunaga [30] in 1958. Moore [7, 8] extends interval arithmetic to rounded interval arithmetic thereby allowing interval arithmetic to be useful in computational mathematics. There are two approaches to interval arithmetic. The first is the traditional interval arithmetic obtained by application of rules (1)–(4) to interval endpoints. The second is the approach that considers an interval as a set and uses the united extension on the arithmetic operations as functions on sets. As will be seen, interval arithmetic derived from the direct application of the united extension has the complexity of global optimization. The traditional approach is simple in its application, but for expressions involving non-trivial computations requires the exponential complexity of partitioning the domain to obtain realistic bounds. That is, the direct extension principle interval arithmetic models the problem of interval arithmetic as a global optimization problem obtaining an intuitive algebra with not only additive/multiplicative identities but additive multiplicative inverses as well at the cost of the complexity of

global optimization. The traditional approach to interval arithmetic obtains a simple and direct approximation from the beginning and adds the exponential complexity of *n-dimensional* partitioning to obtain reasonable bounds as a second step.

There is an interval arithmetic and associated semantics that allows for ‘intervals’ $[a, b]$ for which $a > b$ [55, 56]. This arithmetic is related to directed interval arithmetic (see Section 3.2.2.3) and has some interesting applications to fuzzy control (see [57, 58]).

The basic rules associated with interval arithmetic are (1), (2), (3), and (4). They are more fully developed in [5]. There are various properties associated with the traditional approach to interval arithmetic, which are different from those of real numbers and that of the constraint interval arithmetic. In particular, there is only the subdistributive property. Thus, from [59] we have for intervals X , Y , and Z

1. $X + (Y + Z) = (X + Y) + Z$ – the associative law for addition.
2. $X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$ – the associative law for multiplication.
3. $X + Y = Y + X$ – the commutative law for addition.
4. $X \cdot Y = Y \cdot X$ – the commutative law for multiplication.
5. $[0, 0] + X = X + [0, 0] = X$ – additive identity.
6. $[1, 1] \cdot X = X \cdot [1, 1] = X$ – multiplicative identity.
7. $X \cdot (Y + Z) \subseteq X \cdot Y + X \cdot Z$ – the subdistributive property.

Example 2. [59, p. 13] points out that

$$[1, 2](1 - 1) = [1, 2](0) = 0,$$

whereas

$$[1, 2](1) + [1, 2](-1) = [-1, 1].$$

Moore’s [7] implementation of [30] (neither Moore nor Sunaga was aware of Warmus’ earlier work [29]) has

$$X \circ Y = \{z | z = x \circ y, x \in X, y \in Y, \circ \in \{+, -, \times, \div\}\}.$$

That is, Moore applies the united extension for distinct (independent) intervals X and Y . However, Moore abandons this united extension definition and develops associated rules, assuming independence of all intervals, since it generates rules (1), (2), (3), and (4). These rules lead to a simplification of the operations since one does not have to account for multiple occurrences, while at the same time it leads to overestimation in the presence of dependencies that is severe at times. From the beginning, Moore was aware of the problems of overestimation associated with multiple occurrences of the same variable in an expression. Thus, it is apparent that from the axiomatic approach, $X - X$ is **never** 0 unless X is a real number (a zero width interval). Moreover, $X \div X$ is **never** 1 unless X is a real number (a zero width interval).

3.3.2.1 Interval Arithmetic from Intervals Considered as Pairs of Numbers: Traditional Interval Arithmetic

The traditional approach to interval arithmetic considers all instantiations of variables as independent. That is, Warmus, Sunaga, and Moore’s approach to interval arithmetic is one that considers the same variable that appears more than once in an expression as being independent variables. While axiomatic interval arithmetic is quite simple to implement, it leads to overestimations.

Example 3. Consider

$$f(x) = x(x - 1), x \in [0, 1].$$

Using the traditional interval analysis approach,

$$[0, 1]([0, 1] - 1) = [0, 1][-1, 0] = [-1, 0]. \quad (5)$$

However, the smallest interval containing $f(x) = x(x - 1)$ is $[-0.25, 0]$. Traditional interval arithmetic leads to (5) because the two instantiations of the variable x are taken as independent when in reality they are dependent. The united extension $F(x)$, which is

$$F(x) = \{y \mid y = f(x) = x(x - 1), x \in [0, 1]\} = [-0.25, 0],$$

was not used.

If the calculation were $x(y - 1)$ for $x \in [0, 1]$, $y \in [0, 1]$, then the smallest interval containing $x(y - 1)$, its united extension, is $[-1, 0]$. Note also that the subdistributive property does not use the united extension in computing $X \cdot Y + X \cdot Z$ but instead considers $X \cdot Y + W \cdot Z$ where $W = X$. Partitioning the interval variables (which are repeated) will lead to closer approximation to the united extension. That is, take the example given above and partition the interval in which x lies.

Example 4. Consider $x(x - 1)$ again, but $x \in [0, 0.5] \cup [0.5, 1]$. This yields

$$[0, 0.5] ([0, 0.5] - 1) \cup [0.5, 1]([0.5, 1] - 1) \quad (6)$$

$$= [0, 0.5][-1, -0.5] \cup [0.5, 1][-0.5, 0] \quad (7)$$

$$= [-0.5, 0] \cup [-0.5, 0] \quad (8)$$

$$= [-0.5, 0],$$

which has an overestimation of 0.25 compared with an overestimation of 0.5 when the full interval $[0, 1]$ was used.

In fact, for operations that are continuous functions, a reduction in width leads to estimations that are closer to the united extension and in the limit, to the exact united extension value (see [5, 10, 59]). There are other approaches which find ways to reduce the overestimation arising from the traditional approach that have proved to be extremely useful such as centered, mean value, and slope forms (see [43, 59, 60–63]).

3.3.2.2 Interval Arithmetic from the United Extension: Constraint Interval Arithmetic

The power of the traditional approach to interval arithmetic is that it is simple to apply. Its complexity is at most four times that of real-valued arithmetic (per partition). However, the traditional approach to interval arithmetic leads to overestimations in general because it takes every instantiation of the same variable independently. The united extension when applied to sets of real numbers is global optimization (as will be seen below). On the other hand, simple notions such as

$$X - X = 0 \text{ and} \quad (9)$$

$$X \div X = 1, 0 \notin X \quad (10)$$

are desirable properties and can be maintained if the united extension is used to define interval arithmetic [64]. In the context of fuzzy arithmetic (which uses interval arithmetic), Klir [65] looked at fuzzy arithmetic which was constrained to account for (9) and (10), though from a case-based approach. What is given next was developed in [64] independently of [65] and is more general. It develops interval arithmetic from first principles, the united extension, rather than traditional interval arithmetic or case based.

It is known that applying interval arithmetic to the union of intervals of decreasing width yield tighter bounds on the result that converge to the united extension interval result [10] in the limit. Of course,

for *n-dimensional* problems, ‘intervals’ are rectangular parallelepipeds (boxes), and as the diameters of these boxes approach zero, the union of the result approaches the correct bound for the expression. Partitioning each of the sides of the *n-dimensional* box in half has complexity of $O(2^n)$. Theorems proving convergence to the exact bound of the expression and the rates associated with subdividing intervals can be found in [43, 59, 60–63]. What is proposed here is to redefine interval numbers in such a way that dependencies are explicitly kept. The ensuing arithmetic will be called *constraint interval arithmetic*. This new arithmetic is the derivation of arithmetic directly from the united extension of [24].

An interval number is redefined (also see [64]) into an equivalent form next as the *graph* of a function of one variable and two constants (inputs, coefficients, or parameters).

Definition 5. An interval number $[\underline{x}, \bar{x}]$ (or interval for short) is the **graph** of the real single-valued function $X^I(\lambda_x)$, where

$$X^I(\lambda_x) = \lambda_x \underline{x} + (1 - \lambda_x) \bar{x}, \quad 0 \leq \lambda_x \leq 1. \quad (11)$$

Strictly speaking, in (11), since the numbers \underline{x} and \bar{x} are known (inputs), they are **coefficients**, whereas λ_x is varying, although constrained between 0 and 1, hence the name ‘constraint interval arithmetic.’ Note that (11) defines a *set* representation explicitly and the ensuing arithmetic is developed on sets of numbers. The algebraic operations are defined as follows:

$$\begin{aligned} Z &= X \circ Y \\ &= \{z \mid z = x \circ y, \text{ for all } x \in X^I(\lambda_x), y \in Y^I(\lambda_y), 0 \leq \lambda_x, \lambda_y \leq 1\} \\ &= \{z \mid z = (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}) \circ (\lambda_y \underline{y} + (1 - \lambda_y) \bar{y}), 0 \leq \lambda_x \leq 1, 0 \leq \lambda_y \leq 1\} \\ &= [\underline{z}, \bar{z}], \end{aligned} \quad (12)$$

$$\text{where } \underline{z} = \min \{z\}, \bar{z} = \max \{z\}, \text{ and } \circ \in \{+, -, \times, \div\}. \quad (13)$$

It is clear from (13) that constraint interval arithmetic is a global optimization problem. However, when the operations use the same interval variable, no exceptions need be made as in [65]. We only use (13) and obtain

$$\begin{aligned} Z &= X \circ X \\ &= \{z \mid z = (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}) \circ (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}), \\ &\quad 0 \leq \lambda_x, \lambda_x \leq 1\} \\ &= \{z \mid z = (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}) \circ (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}), \\ &\quad 0 \leq \lambda_x \leq 1\} \\ &= [\underline{z}, \bar{z}]. \end{aligned} \quad (14)$$

This results in the following properties:

1. Addition of the same interval variable:

$$\begin{aligned} X + X &= \{z \mid z = (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}) + (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}), 0 \leq \lambda_x \leq 1\} \\ &= \{z \mid z = 2(\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}), 0 \leq \lambda_x \leq 1\} = [2\underline{x}, 2\bar{x}]. \end{aligned}$$

2. Subtraction of the same interval variable:

$$X - X = \{z \mid z = (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}) - (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}), 0 \leq \lambda_x \leq 1\} = 0.$$

3. Division of the same interval variable, $0 \notin X$:

$$X \div X = \{z \mid z = (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}) \div (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}), 0 \leq \lambda_x \leq 1\} = 1.$$

4. Multiplication of the same interval variable with $\underline{x} < \bar{x}$:

$$\begin{aligned} X \times X &= \{z \mid z = (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}) \times (\lambda_x \underline{x} + (1 - \lambda_x) \bar{x}), 0 \leq \lambda_x \leq 1\} \\ &= \{z \mid z = (\lambda_x^2 \underline{x}^2 + 2(1 - \lambda_x)\underline{x}\lambda_x \bar{x} + (1 - \lambda_x)^2 \bar{x}^2), 0 \leq \lambda_x \leq 1\} \\ &= [\min\{\underline{x}^2, \bar{x}^2, 0\}, \max\{\underline{x}^2, \bar{x}^2, 0\}] = [\underline{z}, \bar{z}]. \end{aligned}$$

To verify that this is the interval solution, note that as a function of the single variable λ_x , the production $X \times X$ is

$$f(\lambda_x) = (\bar{x} - \underline{x})^2 \lambda_x^2 + 2\underline{x}(\bar{x} - \underline{x})\lambda_x + \underline{x}^2,$$

which has a critical point at

$$\lambda_x = -\frac{\underline{x}}{\bar{x} - \underline{x}}.$$

Thus,

$$\begin{aligned} \underline{z} &= \min\{f(0), f(1), f(-\frac{\underline{x}}{\bar{x} - \underline{x}})\} = \min\{\underline{x}^2, \bar{x}^2, 0\} = 0, \\ \bar{z} &= \max\{f(0), f(1), f(-\frac{\underline{x}}{\bar{x} - \underline{x}})\} = \max\{\underline{x}^2, \bar{x}^2, 0\} = \max\{\underline{x}^2, \bar{x}^2\}. \end{aligned}$$

Of course, if $\underline{x} = \bar{x}$, then $X \times X = x^2$.

5. $X(Y + Z) = XY + XZ$.

Constraint interval arithmetic is the complete implementation of the united extension, and it gives an algebra which possesses an additive inverse, a multiplicative inverse, and the distributive law.

3.3.2.3 Specialized Interval Arithmetic

Various interval arithmetic approaches have been developed in addition to the axiomatic and united extension approaches. Different representations of intervals were created and include the development of range arithmetic (see [16, pp. 13–25]) and rational arithmetic (see [66]). These purport to simplify operations and/or result in more accuracy. Another issue addressed by researcher was how to extend interval arithmetic, called *extended interval arithmetic*, to handle unbounded intervals that result from a division by zero (see [32, 67–71]). The general space of improper intervals, which includes extended interval arithmetic, called *directed interval arithmetic* was developed subsequently. However, previously M. Warmus [29, 72] had considered this space and its arithmetic. G. Alefeld states (of improper intervals),

These intervals are interpreted as intervals with negative width. The point intervals $[a, a]$ are no longer minimal elements with respect to the ordering \subseteq . All the structures of $I(\mathbb{R})$ are carried over to $I(\mathbb{R}) \cup I(\mathbb{R})$ and a completion through two improper elements p and $-p$ is achieved. In this manner the division by an interval $A = [\underline{a}, \bar{a}]$ with $\underline{a} \leq 0 \leq \bar{a}$, $\underline{a} \neq \bar{a}$, can also be defined ([73, p. 8]).

This approach was studied by [55, 56]. E.D. Popova [74] states,

Directed interval arithmetic is obtained as an extension of the set of normal intervals by improper intervals and a corresponding extension of the definitions of the interval arithmetic operations. The corresponding extended interval arithmetic structure possesses group properties with respect to addition and multiplication operations and a number of other advantages.

Generalized interval arithmetic (and its more recent generalization, *affine arithmetic* (see [36])) dealt with the problem of reducing overestimation that characterizes the axiomatic approach to interval arithmetic. *Triplex arithmetic* [35] is a way to carry more information about the uncertainty beyond the bounds that are represented by the endpoints of the interval (the endpoints of the support if it is a distribution) by

keeping track of a *main value* within the interval in addition to its endpoints. According to [35], triplex arithmetic started out as a project initiated in 1966 at the University of Karlsruhe to develop a compiler and demonstrate its usefulness for solution to problems in numerical analysis. Three-valued set theory has also been studied by Klaua [75] and Jahn [76]. What is presented here is a synopsis of [35]. Its generalization, *quantile arithmetic* (see [33, 77]), is a way to carry more information about the uncertainty bound in a probabilistic and statistically faithful way than triplex arithmetic. While it is more complex as will be seen, it does have a well-defined probabilistic and statistical semantics. In fact, triplex arithmetic can be represented by quantile arithmetic. In particular, quantile arithmetic approximates distributions whose support is an interval (which can be infinite for extended interval arithmetic) whose value lies between the given lower and upper bounds and whose error at each arithmetic operation is independent. In [33, 77], a three-point arithmetic is used to approximate a discrete distribution, although there is nothing to prevent, using a finer approximation except computational time considerations. In triplex arithmetic, a main value is carried. The uncertainty within a given interval and the manner in which this uncertainty propagates within the interval, when a function or expression is applied, is not a part of interval analysis. The problem of where the uncertainty lies in the resultant of a function or expression is especially problematic when the uncertainty has a large support, and the bulk of the uncertainty is amassed around a single value; that is, it has a narrow dispersion and a long tail.

The *ellipsoidal arithmetic* of [34] is based on approximating enclosing affine transformations of ellipsoids that are again contained in an ellipsoid. The focus of the [34] is to enclose solutions to dynamical system models where the wrapping effect associated with interval (hyperboxes) enclosures may severely hamper their usefulness since boxes parallel to the axes are not the optimal geometric shape to minimized bounds (also see [78]). A second focus of [34] is to enclose confidence limits. In [79] how to compute the ‘tightest’ ellipsoid enclosure of the intersection of two ellipsoids is shown, which is the underlying basis of the approximations developed in [34]. It is clear that computing with ellipsoids is not simple. Therefore, an approximation which is simple is necessary if the method is to be useful. While the sum and product of ellipses are not found explicitly worked out in [34], it is implicit. Enclosing the sum is straightforward. The difference, product, and quotient need approximations.

Variable-precision interval arithmetic ([80, 81], and more recently [82, 83]) was developed to enclose solutions to computational mathematical problems requiring more precision that afforded by usual floating-point arithmetic (single and double precision for example). A problem in this category is wind-shear (vortex) modeling (see [84]). There is a specialized interval arithmetic that has been developed both in software (see [81]) and in hardware (see [82]).

3.3.3 Comparison between Traditional and Constraint Interval Arithmetic

The traditional approach to interval arithmetic considers an interval as a *number* (like a complex number, an interval has two components), whereas constraint interval arithmetic considers an interval as a *set*. In considering an interval as a number, interval arithmetic defines the operations using rules on interval endpoints. Interval arithmetic is simple and straightforward since it is defined via real-number operations which, on a sequential machine, is no less than twice and at most four times more complex than the corresponding real-number operations. What ensues is an arithmetic that does not have additive nor multiplicative inverses and is subdistributive resulting in overestimation of computations. Exponential complexity arises when trying to reduce overestimations.

Considering an interval as a set leads to an arithmetic defined via global optimization of the united extension expressing the continuous function of the arithmetic operations. Thus, constraint interval arithmetic requires a procedure. The complexity is explicit at the onset and potentially NP-hard. Nevertheless, the algebraic structure of constraint interval arithmetic possesses additive and multiplicative inverses as well as distributive.

3.3.4 Enclosure and Verification

Enclosure and verification are approaches to computational mathematical problem solving in which solutions are returned with automatically computed bounds. This is what is mean by enclosure. If the

enclosure is non-empty, a check of existence (and uniqueness if possible) is mathematically carried out on the machine. This is what is mean by verification. There are three different approaches to enclosure and verification methods of interest:

1. **Range of a function methods** compute an upper bound to the maximum and a lower bound to the minimum of a continuous function by using rounded interval arithmetic (see [60, 85, 63], for example).
2. **Epsilon inflation methods** (see [86] for example) compute an approximate solution, inflate the approximate to form an interval, and compute the range according to (1) above.
3. **Defect correction methods** (see [87] for example) compute an approximate inverse to the problem. If the approximate inverse composed with the given function is contractive, then iterative methods are guaranteed to converge and mathematically correct error bounds on the solution can be computed on a digital machine.

The naive (most often non-useful) approach to compute the range of a rational function is to replace every algebraic operation by interval arithmetic operation. This works in theory for continuous function when one takes unions of smaller and smaller boxes whose diameters go to zero. However, this approach is computationally complex. Authors have found excellent and efficient ways to implement, using a variety of theorems along with intelligent methods (see [60, 62]).

The meaning of enclosure and verification in the context of interval analysis is discussed next.

Definition 6. By the *enclosure* of a set of real numbers (real vectors) Y is meant a set of real numbers (real vectors) X such that $Y \subseteq X$. In this case X encloses Y . The set X is called the *enclosing set*.

Enclosure makes sense when Y is an unknown but for which bounds on its values are sought. For example, the set Y could be the set of solutions to a mathematical problem. In the case of interval analysis over \mathbb{R}^n , the enclosing set X is a computed box. Typically, algorithms return a real-number (vector) approximation \tilde{x} as a computed value of the unknown solution y with no sense of the quality of the solution, i.e., the error bounds. The idea of enclosure is to provide mathematically valid computed error bounds, $Y \subseteq X = [\underline{x}, \bar{x}]$, on the set of solutions Y . If the approximation to the solution is $\tilde{x} = \frac{\underline{x} + \bar{x}}{2}$, the maximal error is guaranteed to be $\text{error}_{\max} = \frac{\bar{x} - \underline{x}}{2}$. If we are dealing with functions, there are only two pertinent cases.

- The first is the enclosure of the range of a function in a box; that is, $Y = \{f(x) \mid x \in \text{domain}\} \subseteq X$, where X is a box.
- The second case of function is pointwise enclosure; that is, $[g(x), h(x)]$ encloses the function $f(x)$ pointwise if $g(x) \leq f(x) \leq h(x) \forall x \in \text{domain}$. That is, at each point in the domain, $f(x)$ is enclosed by a box. This is the function envelope of f .

Researchers do not give a definition to ‘enclosure methods,’ since the word itself, ‘enclosure,’ seems to denote its definition. In fact, Alefeld [85] states,

In this paper we do not try to give a precise definition of what we mean by an *enclosure method*. Instead we first recall that the four basic interval operations allow to include the range of values of rational functions. Using more appropriate tools also the range of more general functions can be included. Since all enclosures methods for solution of equations which are based on interval arithmetic tools are finally enclosures methods for the range of some function we concentrate ourselves on methods for the inclusion of the range of function.

There is an intimate relation between enclosure and inclusion of the range of functions. However, enclosure for this study is more general than that to which Alefeld limits himself in [85], since we deal with epsilon inflation and defect correction methods in addition to finding the range of a function. Nevertheless, when the inclusion of the range of a function is computed, it is an important class of enclosure methods (see [60, 62, 63, 85] for example).

The concept of verification for this study is restricted to the context of computed solutions to problems in continuous mathematics. Verification is defined next.

Definition 7. *Verification of solutions to a problem in continuous mathematics in \mathbb{R}^n is the construction of a box X that encloses the solutions of the problem in a given domain where, for $X \neq \emptyset$, at least one solution exists, and for $X = \emptyset$, no solution exists in the given domain of the problem.*

Thus verification includes the *existence* of solutions and the *computability* of enclosures. In particular, when the construction of the verified solution is carried out on a computer, the enclosures are mathematically valid enclosures whose endpoints are floating-point numbers. That is, the construction must take into account round-off errors and of course inherent and truncation errors. The literature often uses ‘validation’ to mean what we have defined as ‘verification.’ Methods that compute enclosures and verified uniqueness are called *E-methods* by [86], and these methods are applied to solutions to fixed-point problems, $f(x) = x$. The authors of [86] also develop methods to solve linear equations by *E-methods*. Many authors, in the context of verifying solutions to equations, use the word ‘proof’ (see, e.g., section 2 of [88]). While the mathematical verification of existence (and perhaps uniqueness) is a type of proof, for this chapter, the mathematical verification of the hypotheses of an existing theorem (say Brouwer’s fixed-point theorem) will simply be called *verification*. Along these lines, [88] states on p. 3,

A powerful aspect of interval computations is tied to the Brouwer fixed point theorem.

Theorem A (Brouwer fixed point theorem – see any elementary text on Real Analysis or [43 page 200]) *Let D be a convex and compact subset of \mathbb{R}^n with $\text{int}(D) \neq \emptyset$. Then every continuous mapping $G : D \rightarrow D$ has at least one fixed point $x^* \in D$, that is, a point with $x^* = G(x^*)$.*

The Brouwer fixed point theory combined with interval arithmetic enables numerical computations to prove existence of solutions to linear and non-linear systems. The simplest context in which this can be explained is the one-dimensional interval Newton method.

Suppose $f : x = [\underline{x}, \bar{x}] \rightarrow \mathbb{R}$ has continuous first derivative on x , $\check{x} \in x$, and $f'(x)$ is a set that contains the range of f' over x (such as when f' is evaluated at x with interval arithmetic). Then the operator

$$N(f; x, \check{x}) = \check{x} - f(\check{x})/f'(x) \quad (15)$$

is termed the *univariate interval Newton method*. . . Applying the Brouwer fixed point theorem in the context of the univariate interval Newton method leads to:

Theorem B *If $N(f; x, \check{x}) \subset x$, then there exists a unique solution to $f(x) = 0$ in x .*

Existence in Theorem B follows from *Miranda’s theorem*, a corollary of the Brouwer fixed point theorem.

We next turn our attention to three types of verifications that occur in practice. These are (1) enclosure of the range of a function or global optimization, (2) epsilon inflation, and (3) defect correction.

3.3.4.1 Enclosure of the Range of a Function

The enclosure of the range of a function using interval arithmetic assumes a function is continuous so that as long as rounded interval arithmetic is used the enclosure (and the existence) is mathematically guaranteed to be correct. Uniqueness can also be verified mathematically on a computer using methods outlined in [43, 60]. More recent methods to compute verified ranges of functions can be found in [61, 62].

3.3.4.2 Epsilon Inflation

Epsilon inflation methods are approaches for the verification of solutions to the problem $f(x) = 0$, using two steps: (1) apply a usual numerical method to solve $f(x) = 0$ to obtain an approximate solution \hat{x} , and (2) inflate \hat{x} to obtain an approximate interval $\hat{X} = [\hat{x} - \epsilon, \hat{x} + \epsilon]$ and apply interval methods using rounded interval arithmetic (e.g., interval Newton's method) to obtain an enclosure. Günter Mayer [89] outlines how to solve problems via *E-methods*, using *epsilon-inflation* techniques to solve $f(x) = 0$ (see p. 98 of [89]), where the function is assumed to be continuous over its domain of definition. The idea is to solve the problem on a closed and bounded subset of its domain, using the following steps:

1. Transform the problem into an equivalent fixed-point problem, $f(x) = 0 \Leftrightarrow g(x) = x$.
2. Solve the fixed point for an approximate solution \tilde{x} using a known algorithm. That is, $g(\tilde{x}) \approx \tilde{x}$.
3. Identify an interval function enclosure to the fixed-point representation of the problem

$$g(x) \in [G](x) \quad \forall x \in [x] \quad \text{where } [x] \text{ is in the domain of both } g \text{ and } [G].$$

For example,

$$[G](x) = [\min_{y \in [x]} \underline{G}(y), \max_{y \in [x]} \overline{G}(y)].$$

4. Verify

$$[G](x) \subseteq \text{interior}[x]$$

by doing the following:

- (a) $[x]^0 := [\tilde{x}, \tilde{x}]$
- (b) $k = -1$
- (c) **repeat**
 - (i) $k := k + 1$
 - (ii) choose $[x]_\epsilon^k$ such that $[x]^k \subseteq \text{interior}([x]_\epsilon^k)$ – this is the *epsilon inflation*
 - (iii) $[x]^{k+1} := [G]([x]_\epsilon^k)$
- (d) **until** $[x]^{k+1} \subseteq \text{interior}([x]_\epsilon^k)$ or $k > k_{\max}$.

There are a variety of ways to pick the epsilon inflation. In particular [90] uses the following:

$$[x]_\epsilon = (1 + \epsilon)[x] - \epsilon[x] + [-\eta, \eta],$$

where η is the smallest floating-point number (machine epsilon). Another approach is as follows:

$$\begin{aligned} [y] &= [\underline{y}, \overline{y}] := (1 + \epsilon)[x] - \epsilon[x] \\ [x]_\epsilon &:= [\text{pred}(\underline{y}), \text{succ}(\overline{y})], \end{aligned}$$

where $\text{pred}(\underline{y})$ denotes the next floating-point number below \underline{y} (round down) and $\text{succ}(\overline{y})$ denotes the next floating-point number above \overline{y} (round up). The value $\epsilon = \overline{0}.1$ has been used as an initial guess.

3.3.4.3 Defect Correction

Defect correction methods [87] solve the fixed-point problem $f(x) = x$ by computing an approximate inverse in such a way that the approximate inverse acting on the original operator is contractive. This approach is then used in conjunction with verification (see [86]), for example, when they are used in conjunction with epsilon inflation and/or range enclosure outlined above. The general defect method as stated by [87, p. 3] is

Solve

$$Fz = y, \tag{16}$$

where $F : D \subset E \rightarrow \hat{D} \subset \hat{E}$ is a bijective, continuous, generally non-linear operator; E and \hat{E} are Banach spaces. The domain and range are defined appropriately so that for every $\tilde{y} \in \hat{D}$ there exists exactly one solution of $Fz = \tilde{y}$. The (unique) solution to (16) is denoted z^* . Assume that (16) cannot be solved directly but the *defect* (also called the residual in other contexts)

$$d(\tilde{z}) := F\tilde{z} - y \quad (17)$$

may be evaluated for ‘approximate solutions’ $\tilde{z} \in D$. Further assume that the *approximate problem*

$$\tilde{F}\tilde{z} = \tilde{y} \quad (18)$$

can be readily solved for $\tilde{y} \in \hat{D}$. That is, we can evaluate the solution operator \tilde{G} of (18). $\tilde{G} : \hat{D} \rightarrow D$ is an *approximate inverse* of F such that (in some approximate sense)

$$\tilde{G}F\tilde{z} = \tilde{z} \quad \text{for } \tilde{z} \in D \quad (19)$$

and

$$F\tilde{G}\tilde{y} = \tilde{y} \quad \text{for } \tilde{y} \in \hat{D}. \quad (20)$$

Assume that an approximation $\tilde{z} \in D$ to z^* is known and the defect $d(\tilde{z})$ (17) has been computed. There are, in general, two ways to compute another (hopefully better) approximation \bar{z} to \tilde{z} by solving (18).

1. Compute a change Δz in (18) with the right-hand side being the defect, $d(\tilde{z})$, and then use Δz as a correction for \tilde{z} . That is,

$$\begin{aligned} \bar{z} &:= \tilde{z} - \Delta z = \tilde{z} - [\tilde{G}(y + d(\tilde{z})) - \tilde{G}y] \\ \bar{z} &:= \tilde{z} - \tilde{G}F\tilde{z} + \tilde{G}y. \end{aligned} \quad (21)$$

This assumes that the approximate inverse, \tilde{G} , is linear; that is, $\tilde{G}(y + d(\tilde{z})) = \tilde{G}y + \tilde{G}(F\tilde{z} - y) = \tilde{G}F\tilde{z}$.

2. Use the known approximate solution \tilde{z} in (18) to compute \tilde{y} . Now change this value by the defect to obtain $\bar{y} = \tilde{y} - d(\tilde{z})$. Use the approximate inverse and solve using \bar{y} . That is,

$$\begin{aligned} \bar{y} &:= \tilde{y} - d(\tilde{z}) = \tilde{y} - (F\tilde{z} - y) \\ &= \tilde{y} - F\tilde{G}\tilde{y} + y, \end{aligned}$$

since $\tilde{y} = F\tilde{z}$, so that from (19) $\tilde{G}\tilde{y} = \tilde{G}F\tilde{z} = \tilde{z}$; that is, $\tilde{z} = F\tilde{G}\tilde{y}$. Now, the new approximation to \bar{z} becomes

$$\bar{z} = \tilde{G}\bar{y} = \tilde{G}[(\tilde{F} - F)\tilde{z} + y], \quad (22)$$

where again, we must assume that the inverse operator \tilde{G} is linear.

The success of the defect correction steps (21) or (22) depends on the *contractivity* of the operators

$$(I - \tilde{G}F) : D \rightarrow D$$

or

$$(I - F\tilde{G}) : \hat{D} \rightarrow \hat{D}$$

respectively since (21) implies

$$\bar{z} - z^* = (I - \tilde{G}F)\bar{z} - (I - \tilde{G}F)z^*,$$

while (22) implies

$$\bar{y} - y^* = (I - F\tilde{G})\bar{y} - (I - F\tilde{G})y^*.$$

The associated iterative algorithm (see [91]) is

Defect correction 1 (21)

$$z_{k+1} = z_k - \tilde{G}Fz_k + \tilde{G}y \quad (23)$$

Defect correction 2 (22)

$$\begin{aligned} y_{k+1} &= y_k - F\tilde{G}y_k + y \\ z_k &= \tilde{G}y_k \\ z_{k+1} &= \tilde{G}[(\tilde{F} - F)z_k + y]. \end{aligned} \quad (24)$$

3.4 Fuzzy Set Theory

Fuzzy set and possibility theory were defined and developed by L. Zadeh beginning with [11] and subsequently [12, 13]. The idea was to mathematize and develop analytical tools to solve problems whose uncertainty was more amble in scope than probability theory. Classical mathematical sets, e.g., a set A , have the property that an element either $x \in A$ or $x \notin A$ but not both. There are no other possibilities for classical sets which are also called *crisp* sets. An interval is a classical set. L. Zadeh's idea was to relax this 'all or nothing' membership in a set to allow for grades of belonging to a set. When grades of belonging are used, a fuzzy set ensues. To each fuzzy set \tilde{A} , L. Zadeh associated a real-valued function $\mu_{\tilde{A}}(x)$, called a *membership function*, for all x in the domain of interest, the universe Ω , whose range is in the interval $[0, 1]$ that describes, quantifies the degree to which x belongs to \tilde{A} . For example, if \tilde{A} is the fuzzy set 'middle-aged person' then a 15-year-old has a membership value of zero, while a 35-year-old might have a membership value of one and a 40-year-old might have a membership value of one-half. That is, a fuzzy set is a set for which membership in the set is defined by its membership function $\mu_{\tilde{A}}(x) : \Omega \rightarrow [0, 1]$, where a value of zero means that an element does not belong to the set \tilde{A} with certainty and a value of one means that the element belongs to the set \tilde{A} with certainty. Intermediate values indicate the degree to which an element belongs to the set. Using this definition, a classical (so-called crisp) set A is a set whose membership function has a range that is binary; that is, $\mu_A(x) : \Omega \rightarrow \{0, 1\}$, where $\mu_A(x) = 0$ means that $x \notin A$ and $\mu_A(x) = 1$ means $x \in A$. This membership function for a crisp set A is, of course, the characteristic function. So a fuzzy set can be thought of as being one which has a generalized characteristic function that admits values in $[0, 1]$ and not just two values $\{0, 1\}$ and is uniquely defined by its membership function. A fuzzy set is a (crisp) set in \mathbb{R}^2 . This follows from the mathematical definition of function as a set of ordered pairs (graph):

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x))\} \subseteq \{(-\infty, \infty) \times [0, 1]\}. \quad (25)$$

Some of the earliest people to recognize the relationship between interval analysis and fuzzy set theory were H. Nguyen [28], implicitly, Dubois and Prade [14,15], and Kaufmann and Gupta [92], explicitly. In particular, [17–19, 21] deal specifically with interval analysis and its relationship with fuzzy set theory. In [19] it is shown that,

set-inclusive monotonicity, as given by R.E. Moore (see [5, 59]), holds for fuzzy quantities. That is, for fuzzy sets \tilde{A} and \tilde{B} ,

$$\tilde{A} \subseteq \tilde{B} \Rightarrow f(\tilde{A}) \subseteq f(\tilde{B}).$$

This crucial result just reminds us that when the operands become more imprecise, the precision of the result cannot but diminish. Due to its close relationship to interval analysis, the calculus of fuzzy quantities is clearly pessimistic about precision, since $f(A_1, A_2)$ is the largest fuzzy set in the sense of fuzzy set inclusion, that is,

$$\tilde{A} \subseteq \tilde{B} \Leftrightarrow \mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x), \forall x.$$

Much has been written about fuzzy sets that can be found in standard textbooks (see, e.g., [23]) and will not be repeated here. We present only the ideas that are pertinent to the areas in the interfaces between interval and fuzzy analysis of interest. Given that the primary interest is in the relationships between real-valued interval and fuzzy analysis, we restrict our fuzzy sets to a real-valued universe $\Omega \subseteq \mathbb{R}$ whose membership functions are *fuzzy numbers* or *fuzzy intervals*, defined next.

Definition 8. A *modal value* of a membership function is a domain value at which the membership function is 1. A fuzzy set with at least one modal value is called **normal**. The **support** of a membership function is the closure of $\{x \mid \mu_{\tilde{A}}(x) > 0\}$.

Definition 9. A *fuzzy interval* is a fuzzy set whose domain is a subset of the reals and whose membership function is upper semicontinuous, normal, and has bounded support. A *fuzzy number* is a fuzzy interval with a unique modal value.

Remark 10. The α -cuts of fuzzy intervals are closed intervals of real numbers for all $\alpha \in (0,1]$. The difference between a fuzzy number and a fuzzy interval is that the modal value for a fuzzy number is just 1 point, whereas for a fuzzy interval, the modal value can be a non-zero width interval. The fact that we have bounded intervals at each α -cut means that fuzzy arithmetic can be defined by interval arithmetic on each α -cut. In fact, when dealing with fuzzy intervals, the operations and analysis are interval operations and analyses on α -cuts.

There is a more recent development of what are called *gradual numbers* (see [20, 21]). In the context of a fuzzy interval (all fuzzy numbers are fuzzy intervals) \tilde{A} with membership function $\mu_{\tilde{A}}(x)$, the idea is to define a gradual number by the inverse of two functions, one is the inverse of the membership function restricted to $(-\infty, m^-]$, i.e., the inverse of the function

$$\mu_{\tilde{A}}^-(x) = \mu_{\tilde{A}}(x), x \in (-\infty, m^-],$$

where $[m^-, m^+]$ is the set of modal values. (Since we are restricting ourselves to fuzzy intervals over the reals, this set is non-empty.) The second function is the inverse of the membership function restricted to $[m^+, \infty)$, i.e., the inverse function

$$\mu_{\tilde{A}}^+(x) = \mu_{\tilde{A}}(x), x \in [m^+, \infty).$$

These inverses are well defined for fuzzy intervals for which $\mu_{\tilde{A}}^-(x)$ ($\mu_{\tilde{A}}^+(x)$) is continuous and strictly increasing (decreasing). These two (inverse) functions

$$(\mu_{\tilde{A}}^-)^{-1}(\alpha) : (0, 1] \rightarrow \mathbb{R} \text{ and} \quad (26)$$

$$(\mu_{\tilde{A}}^+)^{-1}(\alpha) : (0, 1] \rightarrow \mathbb{R} \quad (27)$$

define the gradual numbers in the context of real fuzzy intervals, which is our interest.

Definition 11. A *gradual real number* \tilde{r} is defined by an assignment $A_{\tilde{r}}$ from $(0, 1]$ to \mathbb{R} (see [21]).

Functions $(\mu_{\tilde{A}}^-)^{-1}(\alpha)$ (26), and $(\mu_{\tilde{A}}^+)^{-1}(\alpha)$ (27) are special cases of this definition and the fuzzy sets that describe fuzzy intervals are the ones of interest to this chapter.

3.4.1 Fuzzy Extension Principle

Fuzzy extension principles show how to transform real-valued functions into functions of fuzzy sets. The meaning of arithmetic depends directly on the extension principle in force since arithmetic operations are (continuous) functions over the reals assuming that division by zero is not allowed and over the extended reals [31] when division by zero is allowed. The fuzzy arithmetic coming from Zadeh's extension principle [11] and its relationship to interval analysis has an extensive development (see, e.g., [56, 92]). Moreover, there is an intimate interrelationship between the extension principle being used and the analysis that ensues. For example, in optimization, the way one extends union and intersection via *t-norms* and *t-conorms* will determine the constraint sets so that it captures the way trade-offs among decisions are made.

The extension principle within the context of fuzzy set theory was first proposed, developed, and defined in [11, 13].

Definition 12. (*Extension principle – L. Zadeh*) Given a real-valued function $f : X \rightarrow Y$, the function over fuzzy sets $f : S(X) \rightarrow S(Y)$ is given by

$$\mu_{f(\tilde{A})}(y) = \sup\{\mu_{\tilde{A}}(x) \mid y = f(x)\}$$

for all fuzzy subsets \tilde{A} of $S(X)$ (the set of all fuzzy sets of X). This definition leads to fuzzy arithmetic as we know it. Moreover, it is one of the main mechanisms requisite to perform fuzzy interval analysis.

Various researchers have dealt with the issue of the extension principle and amplified its applicability. H. Nguyen [28] pointed out, in his 1978 paper, that a fuzzy set needs to be defined to be what Dubois and Prade later called a fuzzy interval in order that

$$[f(A, B)]_\alpha = f(A_\alpha, B_\alpha),$$

where the function f is assumed to be continuous. In particular A_α and B_α need to be compact (i.e., closed/bounded intervals) for each α -cut. Thus, H. Nguyen defined a fuzzy number as one whose membership function is upper semicontinuous and whose support was compact. In this case, the α -cuts generated are closed and bounded (compact) sets, i.e., real-valued intervals. This is a well-known result in real analysis.

R. Yager [93] pointed out that by looking at functions as graphs (in the Euclidean plane), the extension principle could be extended to include all graphs, thus allowing for analysis of what he calls ‘non-deterministic’ mappings, i.e., graphs that are not functions. Now, ‘non-determinism’ as is used by Yager can be considered as point-to-set mappings. Thus, Yager implicitly restores the extension principle to a more general setting of point-to-set mappings.

J. Ramik [94] points out that we can restore L. Zadeh's extension principle to its more general setting of set-to-set mappings explicitly. In fact, a fuzzy mapping is indeed a set-to-set mapping. He defines the image of a fuzzy set-to-set mapping as being the set of α 's generated by the function on the α -cuts of domain.

Lastly, T.Y. Lin's paper [95] is concerned with determining the function space in which the fuzzy set generated by the extension principle ‘lives.’ That is, the extension principle generates the resultant membership function in the range space. Suppose one is interested in stable controls, then one way to extend is generate resultant (range space) membership functions that are continuous. The (ϵ/δ) definition of continuous function essentially states that small perturbations in the input, i.e., domain, cause small perturbations in the output, i.e., range, which is one way to view the definition of stability. T.Y. Lin points out conditions that are necessary in order that range membership function have some desired characteristics (such as continuity or smoothness).

What these extension principles express is how to define functions over fuzzy sets so that the resulting range has various properties of interest, defining what may be done in the space of where the extension sends the fuzzy set via the function as dictated by the extension principle itself.

3.4.2 Fuzzy Arithmetic

Fuzzy arithmetic was, like interval arithmetic, derived from the extension principle of Zadeh [11]. S. Nahmias [96] defined fuzzy arithmetic via a convolution:

1. **Addition:** $\mu_{Z=x+y}(z) = \sup_x \min\{\mu_X(x), \mu_Y(z-x)\}$, where $z = x + y$.
2. **Subtraction:** $\mu_{Z=x+y}(z) = \sup_x \min\{\mu_X(x), \mu_Y(x-z)\}$, where $z = x - y$.
3. **Multiplication:** $\mu_{Z=x \times y}(z) = \sup_x \min\{\mu_X(x), \mu_Y(z/x)\}$, where $z = x \times y$.
4. **Division:** $\mu_{Z=x \div y}(z) = \sup_x \min\{\mu_X(x), \mu_Y(x/z)\}$, where $z = x \div y$.

The above definition was how arithmetic of fuzzy entities was originally conceived. When the extension principle of Zadeh [11] was applied to 1–4 above assuming the fuzzy entities involved were non-interactive (independent), what has come to be known as fuzzy arithmetic ensued. That is, much as what occurred to interval arithmetic occurred to fuzzy arithmetic – its roots in the extension principle were eliminated when given non-interaction (independence) and the axioms for arithmetic (using [28] and requiring membership functions to be upper/lower semicontinuous) ensued. Thus, fuzzy arithmetic became interval arithmetic on α -cuts (see [14, 15]).

3.4.2.1 Traditional Fuzzy Arithmetic

The fuzzy arithmetic that has been developed by [92] is taken as the standard approach where a more recent approach is found in [56]. What is needed is the fact that a fuzzy interval is uniquely determined by its α -cuts, $\tilde{A} = \bigcup_{\alpha \in (0,1]} [\mu_{\tilde{A}}^-(\alpha), \mu_{\tilde{A}}^+(\alpha)]$, where $\mu_{\tilde{A}}^-(\alpha)$ and $\mu_{\tilde{A}}^+(\alpha)$ are the left and right endpoints of the α -cuts of the fuzzy set \tilde{A} . In particular, for fuzzy intervals we have

$$\tilde{A} + \tilde{B} = \bigcup_{\alpha \in (0,1]} \{[\mu_{\tilde{A}}^-(\alpha), \mu_{\tilde{A}}^+(\alpha)] + [\mu_{\tilde{B}}^-(\alpha), \mu_{\tilde{B}}^+(\alpha)]\}, \quad (28)$$

$$\tilde{A} - \tilde{B} = \bigcup_{\alpha \in (0,1]} \{[\mu_{\tilde{A}}^-(\alpha), \mu_{\tilde{A}}^+(\alpha)] - [\mu_{\tilde{B}}^-(\alpha), \mu_{\tilde{B}}^+(\alpha)]\}, \quad (29)$$

$$\tilde{A} \times \tilde{B} = \bigcup_{\alpha \in (0,1]} \{[\mu_{\tilde{A}}^-(\alpha), \mu_{\tilde{A}}^+(\alpha)] \times [\mu_{\tilde{B}}^-(\alpha), \mu_{\tilde{B}}^+(\alpha)]\}, \quad (30)$$

$$\tilde{A} \div \tilde{B} = \bigcup_{\alpha \in (0,1]} \{[\mu_{\tilde{A}}^-(\alpha), \mu_{\tilde{A}}^+(\alpha)] \div [\mu_{\tilde{B}}^-(\alpha), \mu_{\tilde{B}}^+(\alpha)]\}. \quad (31)$$

For fuzzy sets whose membership functions are semicontinuous,

$$(\tilde{A} * \tilde{B})_\alpha = (\tilde{A})_\alpha * (\tilde{B})_\alpha, * \in \{+, -, \times, \div\}.$$

Computer implementation of (28)–(31) can be found in [97]. This program uses INTLAB (another downloadable system that has interval data types and runs in conjunction with MATLAB) to handle the fuzzy arithmetic on α -cuts.

3.4.2.2 Case-Based Fuzzy Arithmetic

Klir [65] notices, as Moore before him, that if (28)–(31) are used, overestimations will occur. Moreover, when this approach is used, $\tilde{A} - \tilde{A} \neq 0$ and $\tilde{A} \div \tilde{A} \neq 1$. Klir's idea for fuzzy arithmetic with requisite constraints is to do fuzzy arithmetic with constraints dictated by the context of the problem. That is, Klir defines exceptions to obtain $\tilde{A} - \tilde{A} = 0$ and $\tilde{A} \div \tilde{A} = 1$.

3.4.2.3 Constraint Fuzzy Arithmetic

Klir's approach to fuzzy arithmetic [65] requires an a priori knowledge (via cases) of which variables are identically the same. Constraint fuzzy arithmetic [64] carries this information in the parameters; that is, it performs (28), (29), (30), and (31) using parameter λ_x that identifies the variable. The resulting fuzzy arithmetic derived from constraint interval arithmetic on α -cuts is essentially fuzzy arithmetic with requisite constraints of Klir without cases.

3.4.2.4 Fuzzy Arithmetic Using Gradual Numbers

The implementation of [21] as a way to perform fuzzy arithmetic uses (26) and (27) in the following way:

$$\tilde{A} * \tilde{B} = \begin{cases} (\mu_{\tilde{A} * \tilde{B}}^-)^{-1}(\alpha) = \min\{(\mu_{\tilde{A}}^-)^{-1}(\alpha) * (\mu_{\tilde{B}}^-)^{-1}(\alpha), (\mu_{\tilde{A}}^-)^{-1}(\alpha) * (\mu_{\tilde{B}}^+)^{-1}(\alpha), \\ \quad (\mu_{\tilde{A}}^+)^{-1}(\alpha) + (\mu_{\tilde{B}}^-)^{-1}(\alpha)(\mu_{\tilde{A}}^+)^{-1}(\alpha) * (\mu_{\tilde{B}}^+)^{-1}(\alpha)\} \\ (\mu_{\tilde{A} * \tilde{B}}^+)^{-1}(\alpha) = \max\{(\mu_{\tilde{A}}^-)^{-1}(\alpha) * (\mu_{\tilde{B}}^-)^{-1}(\alpha), (\mu_{\tilde{A}}^-)^{-1}(\alpha) * (\mu_{\tilde{B}}^+)^{-1}(\alpha), \\ \quad (\mu_{\tilde{A}}^+)^{-1}(\alpha) + (\mu_{\tilde{B}}^-)^{-1}(\alpha)(\mu_{\tilde{A}}^+)^{-1}(\alpha) * (\mu_{\tilde{B}}^+)^{-1}(\alpha)\} \end{cases}$$

for $* \in \{+, -, \times, \div\}$.

3.5 Historical Context of Interval Analysis

The context in which fuzzy set theory arose is quite well known, whereas for interval analysis this is perhaps not as clear since its precursors go back at least to Archimedes and more recently Burkhill [98] in 1924. L. Zadeh is recognized as the primary impetus in the creation and development of fuzzy set theory. R.E. Moore, T. Sunaga, and M. Warmus have played a similar role in interval analysis. While there are five known direct and clear precursors to Moore's version of interval arithmetic and interval analysis beginning in 1924 (see [29, 30, 53, 54, 98]), Moore worked out rounded computer arithmetic and fully developed the mathematical analysis of intervals, called interval analysis.

Interval analysis has an early history in Archimedes' computation of circumference of a circle [40]. However as developed by R.E. Moore (see [7–9]), interval analysis arose from the attempt to compute error bounds of numerical solutions on a finite-state machine that accounted for all numerical and truncation error, including round-off error, automatically (by the computer itself). This leads in a natural way to the investigation of computations with intervals as the entity, data type, that enabled automatic error analysis.

R.E. Moore and his colleagues are responsible for developing the early theory, extensions, vision and wide applications of interval analysis, and the actual implementation of these ideas to computers. The major contributions that Moore made have to include at least the following:

1. He recognized how to use intervals in computational mathematics, now called *numerical analysis*.
2. He extended and implemented the arithmetic of intervals to computers.
3. His work was influential in creating IEEE standards for accessing computer's rounding processes, which is a necessary step in obtaining computer-generated validated computations (see [39]).
4. He developed the *analysis* associated with intervals where, as will be seen, functions of intervals, called the *united extension*, play a key role in achieving this. Citing but one major achievement in this area, he showed that Taylor series methods for solving differential equations are not only more tractable but more accurate (see [59]).
5. He was the first to recognize the usefulness of interval analysis for use in computer *verification* methods especially for solutions to non-linear equations using interval Newton's method in which the method includes verification of existence and uniqueness of solution(s).

3.6 Conclusion

This chapter presented the main themes pertinent to mathematical analysis associated with granules that are intervals and fuzzy interval sets. The intimate relationship between interval analysis and fuzzy interval analysis was shown. The central role that the extension principle plays in both the arithmetic and the resulting analysis were discussed. Lastly, the role of interval analysis in the areas of enclosure and verification was highlighted. The reader who is interested in the role of enclosure and verification methods with respect to fuzzy sets, possibility theory, and probability theory is directed to [4, 42, 99], where, among other things, enclosure and verification methods applied to risk analysis as well as optimization under uncertainty are developed.

References

- [1] R.E. Moore and W.A. Lodwick. Interval analysis and fuzzy set theory. *Fuzzy Sets Syst.* 135(1) (2003) 5–9.
- [2] W.A. Lodwick and K.D. Jamison (eds). Special issue on the interfaces between fuzzy set theory and interval analysis. *Fuzzy Sets Syst.* 135(1) (April 2003).
- [3] W.A. Lodwick (ed.). Special issue on linkages between interval analysis and fuzzy set theory. *Reliab. Comput.* 9(2) (April 2003).
- [4] W.A. Lodwick. Interval and fuzzy analysis: A unified approach, *Adv. Imag. Electron. Phys.* 148 (2007) 75–192.
- [5] R.E. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1966.
- [6] J.D. Pryce and G.F. Corliss. Interval arithmetic with containment sets. *Computing*, 78(3) (2006) 25–276.
- [7] R.E. Moore. *Automatic Error Analysis in Digital Computation*. Technical Report LMSD-48421. Lockheed Missile and Space Division, Sunnyvale, CA, 1959. See http://interval.louisiana.edu/Moores_early_papers/bibliography.html.
- [8] R.E. Moore and C.T. Yang. *Interval Analysis I*. Technical Report LMSD285875. Lockheed Missiles and Space Division, Sunnyvale, CA, 1959.
- [9] R.E. Moore, W. Strother, and C.T. Yang. *Interval Integrals*. Technical Report LMSD703073. Lockheed Missiles and Space Division, Sunnyvale, CA, 1960.
- [10] R.E. Moore. *Interval Arithmetic and Automatic Error Analysis in Digital Computing*. Ph.D. Thesis. Stanford University, Stanford, CA. Published as Applied Mathematics and Statistics Laboratories Technical Report No. 25, November 15, 1962. See http://interval.louisiana.edu/Moores_early_papers/bibliography.html.
- [11] L.A. Zadeh. Fuzzy sets. *Inf. Control* 8 (1965) 338–353.
- [12] L.A. Zadeh. Probability measures of fuzzy events. *J. Math. Anal. Appl.* 23 (1968) 421–427.
- [13] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Inf. Sci. Pt I* 8 (1975) 199–249; *Part II* 8 (1975) 301–357; *Part III* 9 (1975) 43–80.
- [14] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, 1980.
- [15] D. Dubois and H. Prade. Additions of interactive fuzzy numbers. *IEEE Trans. Autom. Control* 26(4) (1981) 926–936.
- [16] O. Aberth. *Precise Numerical Analysis*. William C. Brown, Dubuque, IO, 1988.
- [17] D. Dubois and H. Prade. Evidence theory and interval analysis. In: *Second IFSA Congress*, Tokyo, July 20–25, 1987, pp. 502–505.
- [18] D. Dubois and H. Prade. Random sets and fuzzy interval analysis. *Fuzzy Sets Syst.* 42 (1991) 87–101.
- [19] D. Dubois, E. Kerre, R. Mesiar, and H. Prade. Chapter 10: Fuzzy interval analysis. In: D. Dubois and H. Prade (eds). *Fundamentals of Fuzzy Sets*. Kluwer Academic Press, Dordrecht, 2000.
- [20] D. Dubois and H. Prade. Fuzzy elements in a fuzzy set. In: *Proceedings of the 10th International Fuzzy System Association (IFSA) Congress*, Beijing 2005, pp. 55–60.
- [21] J. Fortin, D. Dubois, and H. Fargier. Gradual numbers and their application to fuzzy interval analysis, *IEEE Trans. Fuzzy Syst.* (2008, in press).
- [22] D. Dubois and H. Prade. *Possibility Theory an Approach to Computerized Processing of Uncertainty*. Plenum Press, New York, 1988.
- [23] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic*. Prentice Hall, Upper Saddle River, NJ, 1995.
- [24] W. Strother. *Continuity for Multi-Valued Functions and Some Applications to Topology*. Ph.D. Thesis. Tulane University, New Orleans, LA, 1952.
- [25] W. Strother. Fixed points, fixed sets, and m-retracts. *Duke Math. J.* 22(4) (1955) 551–556.
- [26] J.-P. Audin and H. Frankowska. *Set-Valued Analysis*. Birkhäuser, Boston, 1990.
- [27] W. Strother. Continuous multi-valued functions. *Boletim da Sociedade de Matematica de São Paulo* 10 (1958) 87–120.

- [28] H.T. Nguyen. A note on the extension principle for fuzzy sets. *J. Math. Anal. Appl.* 64 (1978) 369–380.
- [29] M. Warmus. Calculus of approximations. *Bull. Acad. Pol. Sci. Cl. III* (4), (1956) 253–259. See <http://www.cs.utep.edu/interval-comp/early.html>.
- [30] T. Sunaga. Theory of an interval algebra and its application to numerical analysis. *RAAG Mem.* 2 (1958) 547–564. See <http://www.cs.utep.edu/interval-comp/early.html>.
- [31] E.R. Hansen. A generalized interval arithmetic. In: K. Nickel (ed.), *Interval Mathematics, Lecture Notes in Computer Science* 29. Springer-Verlag, New York, 1975, pp. 7–18.
- [32] W.M. Kahan. *A More Complete Interval Arithmetic*. Lecture Notes for a Summer Course at University of Michigan, Ann Arbor, MI, 1968.
- [33] M.A.H. Dempster. An application of quantile arithmetic to the distribution problem in stochastic linear programming. *Bull. Inst. Math. Appl.* 10 (1974) 186–194.
- [34] A. Neumaier. The wrapping effect, ellipsoid arithmetic, stability and confidence regions. *Comput. Suppl.* 9 (1993) 175–190.
- [35] K. Nickel. Triplex-Algol and its applications. In: E.R. Hansen (ed.), *Topics in Interval Analysis*. Oxford Press, New York, 1969, pp. 10–24.
- [36] J. Stolfi, M.V.A. Andrade, J.L.D. Comba, and R. Van Iwaarden. Affine arithmetic: a correlation-sensitive variant of interval arithmetic. See <http://www.dcc.unicamp.br/~stolfi/EXPORT/projects/affine-arith>, accessed January 17, 2008.
- [37] U. Kulisch. An axiomatic approach to rounded computations. *Numer. Math.* 18 (1971) 1–17.
- [38] U. Kulisch and W.L. Miranker. *Computer Arithmetic in Theory and Practice*. Academic Press, New York, 1981.
- [39] U. Kulisch and W.L. Miranker. The arithmetic of the digital computer: a new approach. *SIAM Rev.* 28(1) (1986) 1–40.
- [40] Archimedes of Siracusa. On the measurement of the circle. In: T.L. Heath (ed.), *The Works of Archimedes*. Cambridge University Press, Cambridge, 1897; Dover edition, 1953, pp. 91–98.
- [41] G.M. Phillips. Archimedes the numerical analyst. *Am. Math. Mon.* (1981) 165–169.
- [42] W.A. Lodwick and K.D. Jamison. Interval-valued probability in the analysis of problems that contain a mixture of fuzzy, probabilistic and interval uncertainty. In: K. Demirli and A. Akgunduz (eds), *2006 Conference of the North American Fuzzy Information Processing Society*, June 3–6, 2006, Montréal, Canada, paper 327137.
- [43] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge Press, Cambridge, 1990.
- [44] W. Tucker. A rigorous ODE solver and Smale’s 14th problem. *Found. Comput. Math.* 2 (2002) 53–117.
- [45] S. Smale. Mathematical problems for the next century. *Math. Intell.* 20(2) (1998) 7–15.
- [46] B. Davies. Whither mathematics? *Not. AMS* 52(11) (December 2005) 1350–1356.
- [47] T.C. Hales. Cannonballs and honeycombs. *Not. Am. Math. Soc.* 47 (2000) 440–449.
- [48] F. Bornemann, D. Laurie, S. Wagon, and J. Waldvogel. *The SIAM 100-Digit Challenge: A Study in High-Accuracy Numerical Computing*. SIAM, Philadelphia, 2004.
- [49] D. Daney, Y. Papegay, and A. Neumaier. Interval methods for certification of the kinematic calibration of parallel robots. In: *IEEE International Conference on Robotics and Automation*, New Orleans, LA, April 2004, pp. 191–198.
- [50] L. Jaulin. Path planning using intervals and graphs. *Reliab. Comput.* 7(1) (2001) 1–15.
- [51] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer, New York, 2001.
- [52] G.F. Corliss. Tutorial on validated scientific computing using interval analysis. In: *PARA’04 Workshop on State-of-the-Art Computing*. Technical University of Denmark, Denmark, June 20–23, 2004. See http://www.eng.mu.edu/corliss/PARA04/READ_ME.html.
- [53] R.C. Young. The algebra of many-valued quantities. *Math. Ann. Band* 104 (1931) 260–290.
- [54] P.S. Dwyer. *Linear Computations*, Wiley, New York, 1951.
- [55] E. Gardeñes, H. Mielgo, and A. Trepot. Modal intervals: reasons and ground semantics. In: K. Nickel (ed.), *Interval Mathematics 1985: Proceedings of the International Symposium*, Freiburg i. Br., Federal Republic of Germany, September 23–26, 1985, pp. 27–35.
- [56] M. Hanss. *Applied Fuzzy Arithmetic*. Springer-Verlag, Berlin, 2005.
- [57] J. Bondia, A. Sala, and M. Sáinz. Modal fuzzy quantities and applications to control. In: K. Demirli and A. Akgunduz (eds), *2006 Conference of the North American Fuzzy Information Processing Society*, June 3–6, 2006, Montréal, Canada, 2006, paper 327134.
- [58] M.A. Sáinz. Modal intervals. *Reliab. Comput.* 7(2) (2001) 77–111.
- [59] R.E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.
- [60] E.R. Hansen. *Global Optimization Using Interval Arithmetic*. Marcel Dekker, New York, 1992.
- [61] R.B. Kearfott. *Rigorous Global Search: Continuous Problem*. Kluwer Academic Publishers, Boston, 1996.
- [62] A. Neumaier. Complete search in continuous global optimization and constraint satisfaction. In: A. Iserles (ed.), *Acta Numerica 2004*. Cambridge University Press, Cambridge, 2004, pp. 271–369.

- [63] H. Ratschek and J. Rokne. *New Computer Methods for Global Optimization*. Horwood, Chichester, England, 1988.
- [64] W.A. Lodwick. *Constrained Interval Arithmetic*. CCM Report 138, February 1999.
- [65] G.J. Klir. Fuzzy arithmetic with requisite constraints. *Fuzzy Sets Syst.* 91(2) (1997) 165–175.
- [66] P. Korenerup and D.W. Matula. Finite precision rational arithmetic: an arithmetic unit. *IEEE Trans. Comput.* 32(4) (1983) 378–388.
- [67] E.R. Hansen. Interval forms of Newton's method. *Computing* 20 (1978) 153–163.
- [68] G.W. Walster. The extended real interval system (personal copy from the author, 1998).
- [69] H.-J. Ortolff. *Eine Verallgemeinerung der Intervallarithmetik*. Gesellschaft fuer Mathematik und Datenverarbeitung, Bonn, Germany, 1969 Vol. 11, pp. 1–71.
- [70] E. Kaucher. *Über metrische und algebraische Eigenschaften einger beim numerischen Rechnen auftretender Räume*. Ph.D. Thesis. University of Karlsruhe, Karlsruhe, Germany, 1973.
- [71] E. Kaucher. Interval analysis in the extended space $I\mathbb{R}$. *Comput. Suppl.* 2 (1980) 33–49.
- [72] M. Warmus. Approximations and inequalities in the calculus of approximations. Classification of approximate numbers. *Bull. Acad. Pol. Sci. Cl. IX* (4) (1961) 241–245. See <http://www.cs.utep.edu/interval-comp/early.html>.
- [73] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, New York, 1983.
- [74] E.D. Popova. <http://www.math.bas.bg/~epopova/directed.html>, accessed January 17, 2008.
- [75] D. Klaaua. Partielle Mengen und Zahlen. *Mitber. Dt. Akad. Wiss.* 11 (1969) 585–599.
- [76] K. Jahn. The importance of 3-valued notions for interval mathematics. In: K.E. Nickel (ed.) *Interval Mathematics 1980*. Academic Press, New York, 1980, pp. 75–98.
- [77] M.A.H. Dempster. Distributions in interval and linear programming. In: E.R. Hansen (ed.), *Topics in Interval Analysis*. Oxford Press, New York, 1969, pp. 107–127.
- [78] K.G. Guderley and C.L. Keller. A basic theorem in the computation of ellipsoidal error bounds. *Numer. Math.* 19(3) (1972) 218–229.
- [79] W.M. Kahan. Circumscribing an ellipsoid about the intersection of two ellipsoids. *Can. Math. Bull.* 11(3) (1968) 437–441.
- [80] R.E. Moore. Computing to arbitrary accuracy. In: C. Bresinski and U. Kulisch (eds), *Computational and Applied Mathematics I: Algorithms and Theory*. North-Holland, Amsterdam, 1992, pp. 327–336.
- [81] J.S. Ely. The VPI software package for variable precision interval arithmetic. *Interval Comput.* 2(2) (1993) 135–153.
- [82] M.J. Schulte and E.E. Swartzlander, Jr. A family of variable-precision interval processors. *IEEE Trans. Comput.* 49(5) (2000) 387–397.
- [83] N. Revol and F. Rouillier. Motivations for an arbitrary precision interval arithmetic and the MPFI library. *Reliab. Comput.* 11(4) (2005) 275–290.
- [84] J.S. Ely and G.R. Baker. High-precision calculations of vortex sheet motion. *J. Comput. Phys.* 111 (1993) 275–281.
- [85] G. Alefeld. Enclosure methods. In: C. Ullrich (ed.), *Computer Arithmetic and Self-Validating Numerical Methods*. Academic Press, Boston, 1990, pp. 55–72.
- [86] E. Kaucher and S.M. Rump. E-methods for fixed point equations $f(x) = x$. *Computing* 28(1) (1982) 31–42.
- [87] K. Böhmer, P. Hemker, and H.J. Stetter. The defect correction approach. *Comput. Suppl.* 5 (1984) 1–32.
- [88] R.B. Kearfott. Interval computations: introduction, uses, and resources. *Euromath. Bull.* 2(1) (1996) 95–112.
- [89] G. Mayer. Success in epsilon-inflation. In: G. Alefeld and B. Lang (eds), *Scientific Computing and Validated Numerics: Proceedings of the International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics SCAN-95*, Wuppertal, Germany, September 26–29, 1995, Akademie Verlag, Berlin, May 1996, pp. 98–104.
- [90] G. Mayer. Epsilon-inflation in verification algorithms. *J. Comput. Appl. Math.* 60 (1995) 147–169.
- [91] H.J. Stetter. The defect correction principle and discretization methods. *Numer. Math.* 29 (1978) 425–443.
- [92] A. Kaufmann and M.M. Gupta. *Introduction to Fuzzy Arithmetic—Theory and Applications*. Van Nostrand Reinhold, New York, 1985.
- [93] R.R. Yager. A characterization of the extension principle. *Fuzzy Sets Syst.* 18 (1986) 205–217.
- [94] J. Ramik. Extension principle in fuzzy optimization. *Fuzzy Sets Syst.* 19 (1986) 29–35.
- [95] T.Y. Lin. A function theoretical view of fuzzy sets: new extension principle. In: D. Filev and H. Ying (eds.), *Proceedings of the 2005 North American Fuzzy Information Processing Society Annual Conference: Computing for Real World Applications*, Ann Arbor, MI, 2005.
- [96] S. Nahmias. Fuzzy variable. *Fuzzy Sets Syst.* 1 (1978) 97–110.
- [97] M. Anile, S. Deodato, and G. Privitera. Implementing fuzzy arithmetic. *Fuzzy Sets Syst.* 72 (2) (1995) 239–250.
- [98] J.C. Burkill. Functions of intervals. *Proc. Lond. Math. Soc.* 22 (1924) 375–446.

- [99] K.D. Jamison and W.A. Lodwick. Interval-valued probability in the analysis of problems containing a mixture of fuzzy, possibilistic, probabilistic and interval uncertainty. *Fuzzy Sets Syst.*, 2008, in press.
- [100] J.J. Buckley. A generalized extension principle. *Fuzzy Sets Syst.* 33 (1989) 241–242.
- [101] A. Deif. *Sensitivity Analysis in Linear Systems*. Springer-Verlag, New York, 1986.
- [102] D. Dubois, S. Moral, and H. Prade. Semantics for possibility theory based on likelihoods. *J. Math. Anal. Appl.* 205 (1997) 359–380.
- [103] D. Dubois and H. Prade. *Le Flou, Mécédonka?* Technical Report, C.E.R.T.-D.E.R.A., Toulouse, France, Avril 1977.
- [104] D. Dubois and H. Prade. *Fuzzy Algebra, Analysis, Logics*. Technical Report, N⁰TR-EE 78-13, Purdue University, March 1978.
- [105] D. Dubois and H. Prade. Operations on fuzzy numbers. *Int. J. Syst. Sci.* 9(6) (1978) 613–626.
- [106] D. Dubois and H. Prade. Fuzzy real algebra: some results. *Fuzzy Sets Syst.* 2(4) (1979) 327–348.
- [107] D. Dubois and H. Prade. *Fuzzy Numbers: An Overview*. Technical Report No. 219. L.S.I., University of Paul Sabatier, Toulouse, France. Also In: J.C. Bezdek (ed.), Chapter 1 of *Analysis of Fuzzy Information, Volume 1, Mathematics and Logic*. CRC Press, Boca Raton, FL, 1987.
- [108] D. Dubois and H. Prade. Special Issue on fuzzy numbers. *Fuzzy Sets Syst.* 24(3) (December 1987).
- [109] D. Dubois and H. Prade (eds). *Fundamentals of Fuzzy Sets*. Kluwer Academic Press, Dordrecht, 2000.
- [110] R. Fullér and T. Keresztfalvi. On generalization of Nguyen's theorem. *Fuzzy Sets Syst.* 41 (1990) 371–374.
- [111] E.R. Hansen (ed.). *Topics in Interval Analysis*. Oxford Press, New York, 1969.
- [112] E.R. Hansen. Publications Related to Early Interval Work of R. E. Moore, August 13, 2001. See http://interval.louisiana.edu/Moores_early_papers/bibliography.html.
- [113] T. Hickey, Q. Ju, and M.H. van Emden. Interval arithmetic: from principles to implementation. *J. ACM* 48(5) (2001) 1038–1068.
- [114] S. Kaplan. On the method of discrete probability distributions in risk and reliability calculations—applications to seismic risk assessment. *J. Risk* 1(3) (1981) 189–196.
- [115] G.J. Klir. Chapter 1: The role of constrained fuzzy arithmetic in engineering. In: B. Ayyub and M.M. Gupta (eds), *Uncertainty Analysis in Engineering and Sciences: Fuzzy Logic, Statistics, and Neural Network Approach*. Kluwer Academic Publishers, Dordrecht, pp. 1–19. 1998.
- [116] W.A. Lodwick. Constraint propagation, relational arithmetic in AI systems and mathematical programs. *Ann. Oper. Res.* 21 (1989) 143–148.
- [117] W.A. Lodwick. Analysis of structure in fuzzy linear programs. *Fuzzy Sets Syst.* 38 (1990) 15–26.
- [118] M. Mizumoto and K. Tanaka. The four operations of arithmetic on fuzzy numbers. *Syst. Comput. Control* 7(5) (1976) 73–81.
- [119] R.E. Moore. The automatic analysis and control of error in digital computing based on the use of interval numbers. In: L.B. Rall (ed.), *Error in Digital Computation*. John Wiley and Sons, New York, 1965, vol. I, Chapter 2, pp. 61–130.
- [120] R.E. Moore. The dawning. *Reliab. Comput.* 5 (1999) 423–424.
- [121] B. Russell. Vagueness. *Aust. J. Phil.* 1 (1924) 84–92.
- [122] J.R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Comput. Geom. Theory Appl.* 22(1–3) (2002) 21–74.
- [123] J.A. Tupper. *Graphing Equations with Generalized Interval Arithmetic*. Ph.D. Thesis. University of Toronto, Ontario, 1996.
- [124] R.R. Yager. A procedure for ordering fuzzy subsets of the unit interval. *Inf. Sci.* 24 (1981) 143–161.

4

Interval Methods for Non-Linear Equation Solving Applications

Courtney Ryan Gwaltney, Youdong Lin, Luke David Simoni,
and Mark Allen Stadtherr

4.1 Overview

A problem encountered frequently in virtually any field of science, engineering, or applied mathematics is the solution of systems of non-linear algebraic equations. There are many applications in which such systems may have multiple solutions, a single solution, or no solution, with the number of solutions often unknown *a priori*. Can *all* solutions be found? If there are no solutions, can this be verified? These are questions that are difficult or impossible to answer using conventional local methods for equation solving. However, methods based on interval analysis are available that can answer these questions, and do so with mathematical and computational rigor. Such methods are based on the processing of granules in the form of intervals and can thus be regarded as one facet of granular computing [1]. The remainder of this chapter is organized as follows: in the next section, a brief summary of interval arithmetic is provided, and some of the key concepts used in interval methods for equation solving are reviewed. In subsequent sections, we focus on specific application areas, namely the modeling of phase equilibrium (Section 4.3), transition-state analysis (Section 4.4), and ecological modeling (Section 4.5).

4.2 Background

4.2.1 Interval Arithmetic

Interval arithmetic in its modern form was introduced by Moore [2] and is based on arithmetic conducted on closed sets of real numbers. A real interval X is defined as the set of real numbers between (and including) given upper and lower bounds. That is, $X = [\underline{X}, \overline{X}] = \{x \in \mathbb{R} \mid \underline{X} \leq x \leq \overline{X}\}$. Here an underline is used to indicate the lower bound of an interval, while an overline is used to indicate the upper bound. Unless indicated otherwise, uppercase quantities are intervals and lowercase quantities or uppercase quantities with an underline or overline are real numbers. An interval vector $\mathbf{X} = (X_1, X_2, \dots, X_n)^T$ has n interval components and can be interpreted geometrically as an n -dimensional rectangular convex polytope or ‘box.’ Similarly, an $n \times m$ interval matrix A has interval elements A_{ij} , $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$.

Interval arithmetic is an extension of real arithmetic. For a real arithmetic operation $\text{op} \in \{+, -, \times, \div\}$, the corresponding interval operation on intervals X and Y is defined by

$$X \text{ op } Y = \{x \text{ op } y \mid x \in X, y \in Y\}. \quad (1)$$

That is, the result of an interval arithmetic operation on X and Y is an interval enclosing the range of results obtainable by performing the operation with any number in X and any number in Y . Interval extensions of the elementary functions (\sin , \cos , \exp , \log , etc.) can be defined similarly and computed using interval arithmetic operations on the appropriate series expansions. For dealing with exceptions, such as division by an interval containing zero, extended models for interval arithmetic are available, often based on the extended real system $\mathbb{R}^* = \mathbb{R} \cup \{-\infty, +\infty\}$. The concept of containment sets (csets) provides a valuable framework for constructing models for interval arithmetic with consistent handling of exceptions [3, 4]. When machine computations using intervals are performed, rounding errors must be handled correctly in order to ensure that the result is a rigorous enclosure. Since computers can represent only a finite set of real numbers (machine numbers), the results of floating-point arithmetic operations that compute the endpoints of an interval must be determined using a directed (outward) rounding, instead of the standard round-to-nearest procedure. Through the use of interval arithmetic with directed outward rounding, as opposed to floating-point arithmetic, any potential rounding error problems are avoided. Several good introductions to interval analysis, including interval arithmetic and other aspects of computing with intervals, are available [3, 5–8]. Implementations of interval arithmetic and elementary functions are readily available for a variety of programming environments, including INTLIB [9, 10] for Fortran 77, INTERVAL_ARITHMETIC [11] for Fortran 90, PROFIL/BIAS [12] and FILIB++ [13] for C++, and INTLAB [14] for Matlab. Recent compilers from Sun Microsystems provide direct support for interval arithmetic and an interval data type.

For an arbitrary function $f(x)$, the *interval extension*, denoted by $F(X)$, encloses all possible values of $f(x)$ for $x \in X$. That is, $F(X) \supseteq \{f(x) \mid x \in X\}$ encloses the range of $f(x)$ over X . It is often computed by substituting the given interval X into the function $f(x)$ and then evaluating the function using interval arithmetic. This ‘natural’ interval extension may be wider than the actual range of function values, although it always includes the actual range. The potential overestimation of the function range is due to the ‘dependency’ problem of interval arithmetic, which may arise when a variable occurs more than once in a function expression. While a variable may take on any value within its interval, it must take on the *same* value each time it occurs in an expression. However, this type of dependency is not recognized when the natural interval extension is computed. In effect, when the natural interval extension is used, the range computed for the function is the range that would occur if each instance of a particular variable were allowed to take on a different value in its interval range. For the case in which $f(x)$ is a single-use expression, i.e., an expression in which each variable occurs only once, the use of interval arithmetic will always yield the true function range, not an overestimation. For cases in which obtaining a single-use expression is not possible, there are several other approaches that can be used to tighten interval extensions [3, 5, 7, 8, 15], including the use of monotonicity [16, 17] and the use of Taylor models [18, 19].

4.2.2 Equation-Solving Techniques

There are many ways intervals may be used in non-linear equation solving. No attempt is made to systematically survey all such methods here. Instead, we highlight some of the key concepts used in many interval methods for non-linear equation solving. Many of these concepts can be described in terms of contraction operators, or contractors [5]. Contractors may either reduce the size of or completely eliminate the region in which solutions to the equation system of interest are being sought. Consider the non-linear equation solving problem $\mathbf{f}(x) = \mathbf{0}$, for which real roots are sought in an initial interval $\mathbf{X}^{(0)}$. Interval-based strategies exist for contracting, eliminating, or dividing $\mathbf{X}^{(0)}$. Reliable methods for locating all solutions to an equation system are formed by combining these strategies. For a comprehensive treatment of these techniques, several sources are available, including monographs by Neumaier [8], Kearfott [7], Jaulin *et al.* [5], and Hansen and Walster [3].

4.2.2.1 Function Range Testing

Consider a search for solutions of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ in an interval \mathbf{X} . If an interval extension of $\mathbf{f}(\mathbf{x})$ over \mathbf{X} does not contain zero, i.e., $\mathbf{0} \notin \mathbf{F}(\mathbf{X})$, then the range of $\mathbf{f}(\mathbf{x})$ over \mathbf{X} does not contain zero, and it is not possible for \mathbf{X} to contain a solution of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. Thus, \mathbf{X} can be eliminated from the search space. The use of interval extensions for function range testing is one simple way an interval can be eliminated as not containing any roots. This is commonly used in non-linear equation solving methods prior to use of the contraction methods discussed below. A method that makes more extensive use of function range testing was developed by Yamamura [20] on the basis of use of linear combinations of the component functions of $\mathbf{f}(\mathbf{x})$. An approach for forming the linear combinations based on the inverse of the midpoint of the interval extension of the Jacobian of $\mathbf{f}(\mathbf{x})$ was shown to be very effective.

4.2.2.2 Constraint Propagation

Most constraint propagation strategies for non-linear equation solving are based on the concepts of hull consistency, box consistency, or some combination or variation thereof. These are strategies for contracting (shrinking) intervals in which roots are sought, or possibly eliminating them entirely.

The hull consistency form of constraint propagation is based on an interval extension of fixed-point iteration [3, 21]. Consider a single equation and variable $f(x) = 0$, and let it be reformulated into the fixed-point form $x = g(x)$. If X is the search interval, then any roots of $f(x) = 0$ must be in the interval $\tilde{X} = G(X)$. It may be possible to shrink the search interval by taking the intersection of X and \tilde{X} , i.e., $X \leftarrow X \cap \tilde{X}$. If this results in a contraction of X , then the process may be repeated. Furthermore, if $X \cap \tilde{X} = \emptyset$, then the current search interval can be eliminated entirely as containing no solutions. If there are different ways to obtain the function $g(x)$, then the process can be repeated using these alternative fixed-point forms. For systems of equations, hull consistency can be applied to one equation at a time and one variable at a time (holding other variables constant at their interval values in the current search space). In this way, contractions in one component of the search space can be propagated readily to other components. Clearly, there are many possible strategies for organizing this process.

Another type of constraint propagation strategy is known as box consistency [3, 5]. In this case, all but one of the variables in an equation system are set to their interval values in the current search space. Now there are one or more equations involving only the remaining variable, say x_j . These constraints can be used to contract X_j , the current search range for x_j . There are various ways to do this, including univariate interval-Newton iteration [22] and methods [3] for direct calculation of new bounds for x_j . This procedure can be repeated using a combination of any equation and any variable in the equation system. Again, this provides a way for contractions in one component of the search space to be propagated to other components. Box consistency and hull consistency tests can also be easily combined [3, 23]. A variety of software packages are available that apply constraint propagation techniques, often in combination with other interval-based methods, to solve systems of equations. These include RealPaver [24], Numerica [25], and ICOS [26].

4.2.2.3 Krawczyk and Interval-Newton

The Krawczyk and interval-Newton methods are contraction strategies that have been widely used in the solution of non-linear equation systems. They also provide a test for the existence of a unique solution in a given interval. Both are generally applied in connection with some bisection or other tessellation scheme [7], thus resulting in a sequence of subintervals to be tested. Let $\mathbf{X}^{(k)}$ indicate an interval in this sequence. Using Krawczyk or interval-Newton method, it is possible to contract $\mathbf{X}^{(k)}$, or even eliminate it, and also to determine if a unique solution to $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ exists in $\mathbf{X}^{(k)}$.

In the Krawczyk method, the interval $\mathbf{K}^{(k)}$ is computed from

$$\mathbf{K}^{(k)} = \mathbf{K}(\mathbf{X}^{(k)}, \mathbf{x}^{(k)}) = \mathbf{x}^{(k)} - Y^{(k)}\mathbf{f}(\mathbf{x}^{(k)}) + (I - Y^{(k)}F'(\mathbf{X}^{(k)}))(\mathbf{X}^{(k)} - \mathbf{x}^{(k)}). \quad (2)$$

Here, $F'(\mathbf{X}^{(k)})$ indicates an interval extension of the Jacobian matrix of $\mathbf{f}(\mathbf{x})$, but could be any Lipschitz matrix. Also, $\mathbf{x}^{(k)}$ is an arbitrary point in $\mathbf{X}^{(k)}$, and $Y^{(k)}$ is a real preconditioning matrix. The properties

of this method have been widely studied [3, 5, 7, 8, 27–29]. Any roots of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ in $\mathbf{X}^{(k)}$ will also be in $\mathbf{K}^{(k)}$, thus giving the contraction scheme $\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} \cap \mathbf{K}^{(k)}$. It follows that if $\mathbf{X}^{(k)} \cap \mathbf{K}^{(k)} = \emptyset$, then $\mathbf{X}^{(k)}$ contains no roots and can be eliminated. An additional property is that if $\mathbf{K}^{(k)}$ is in the interior of $\mathbf{X}^{(k)}$, then there is a unique root in $\mathbf{X}^{(k)}$. If $\mathbf{X}^{(k)}$ cannot be eliminated or sufficiently contracted, or cannot be shown to contain a unique root, then it is bisected, and the procedure is repeated on each resulting interval. Several improvements to the basic Krawczyk method have been suggested, including a bicentered method [30], a boundary-based method [30, 31], and a componentwise version of the algorithm [32].

In the interval-Newton method, the interval $\mathbf{N}^{(k)} = \mathbf{N}(\mathbf{X}^{(k)}, \mathbf{x}^{(k)})$ is determined from the linear interval equation system

$$Y^{(k)} F'(\mathbf{X}^{(k)}) (\mathbf{N}^{(k)} - \mathbf{x}^{(k)}) = -Y^{(k)} \mathbf{f}(\mathbf{x}^{(k)}). \quad (3)$$

This method has also been widely studied [3, 5, 7, 8, 33, 34] and has properties similar to the Krawczyk method. Any roots in $\mathbf{X}^{(k)}$ are also in $\mathbf{N}^{(k)}$, so the contraction $\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} \cap \mathbf{N}^{(k)}$ can be used. If $\mathbf{X}^{(k)} \cap \mathbf{N}^{(k)} = \emptyset$, then $\mathbf{X}^{(k)}$ can be eliminated. Furthermore, if $\mathbf{N}^{(k)}$ is in the interior of $\mathbf{X}^{(k)}$, there is a unique root in $\mathbf{X}^{(k)}$. In this case, the interval-Newton procedure can be repeated and converges quadratically to a narrow enclosure of the root. Alternatively, an approximation of the root can be found using a standard point-Newton algorithm starting from any point in $\mathbf{X}^{(k)}$. Again, if $\mathbf{X}^{(k)}$ cannot be eliminated or sufficiently shrunk, or cannot be shown to contain a unique root, it is bisected. $\mathbf{N}^{(k)}$ can be obtained from the linear interval equation system (3) in various ways. However, an interval Gauss–Seidel procedure [35] is widely used. In this case, $\mathbf{N}^{(k)}$ is never obtained explicitly, since after each component $N_i^{(k)}$ is computed, it is intersected with $X_i^{(k)}$, and the result is then used in computing subsequent components of $\mathbf{N}^{(k)}$. For a fixed preconditioning matrix, the enclosure provided by the interval-Newton method using Gauss–Seidel is at least as good as that provided by the Krawczyk method [8, 35]. Nevertheless, the Krawczyk method appears attractive, because it is not necessary to bound the solution of a system of linear interval equations. However, in practice the interval Gauss–Seidel procedure is a very simple and effective way to deal with the linear equation system. Overall, interval-Newton with Gauss–Seidel is regarded as computationally more efficient than the Krawczyk method [3, 36].

There are many variations on the interval-Newton method, corresponding to different choices of the real point $\mathbf{x}^{(k)}$ and preconditioning matrix $Y^{(k)}$, different strategies for choosing the bisection coordinate, and different ways to bound the solution of equation (3). The real point $\mathbf{x}^{(k)}$ is typically taken to be the midpoint of $\mathbf{X}^{(k)}$, and the preconditioning matrix $Y^{(k)}$ is often taken to be either the inverse of the midpoint of $F'(\mathbf{X}^{(k)})$ or the inverse of the Jacobian evaluated at the midpoint of $\mathbf{X}^{(k)}$. However, these choices are not necessarily optimal [37]. For example, several alternative preconditioning strategies are given by Kearfott *et al.* [38]. Gau and Stadtherr [39] combined one of these methods, a pivoting preconditioner, with a standard inverse midpoint scheme and were able to obtain significant performance gains compared with the use of the inverse midpoint preconditioner alone. For the choice of the real point $\mathbf{x}^{(k)}$, one alternative strategy is to use an approximation to a root of the equation system, perhaps obtained using a local equation solver. Gau and Stadtherr [39] suggested a real-point selection scheme that seeks to minimize the width of the intersection between $X_i^{(k)}$ and $N_i^{(k)}$. Several bisection or other box-splitting strategies have been studied [3, 7, 29, 40]. The maximum smear heuristic [40], in which bisection is done on the coordinate whose range corresponds to the maximum width in the function range, is often, but not always, an effective choice. For bounding the solution of equation (3) there are many possible approaches, though, as noted above, the preconditioned interval Gauss–Seidel approach is typically quite effective. One alternative, described by Lin and Stadtherr [41, 42], uses a linear programming strategy along with a real-point selection scheme to provide sharp enclosures of the solution $\mathbf{N}^{(k)}$ to equation (3). Although, in general, sharply bounding the solution set of a linear interval equation system is NP-hard, for the special case of interval-Newton, this linear programming approach can efficiently provide exact (within roundout) bounds. Finally, it should be noted that a slope matrix can be used in equations (2) and (3) instead of a Lipschitz matrix. In this case, the test for enclosure of a unique root is no longer applicable, unless some type of compound algorithm is used [43].

In implementing the interval-Newton method, values of $\mathbf{f}(\mathbf{x})$ are computed using interval arithmetic to bound rounding errors. Thus, in effect, $\mathbf{f}(\mathbf{x})$ is interval valued. In general, the interval-Newton method can

be used to enclose the solution set of any interval-valued function. For example, consider the problem $\mathbf{f}(\mathbf{x}, p) = \mathbf{0}$, where p is some parameter. If the value of p is uncertain but is known to be in the interval P , then we have the interval-valued function $\mathbf{F}(\mathbf{x}, P)$ and the problem is to enclose the solution set of $\mathbf{F}(\mathbf{x}, P) = \mathbf{0}$. This solution set is defined by $\mathcal{S} = \{\mathbf{x} \mid \mathbf{f}(\mathbf{x}, p) = \mathbf{0}, p \in P\}$. An interval enclosure of \mathcal{S} can be found readily using the interval-Newton method, though generally, due to bounding of rounding errors, it will not be the smallest possible interval enclosure. However, since \mathcal{S} is often not an interval, even its tightest interval enclosure may still represent a significant overestimation. To more closely approximate \mathcal{S} , one can divide P into subintervals, obtain an interval enclosure of the solution set over each subinterval, and then take the union of the results.

Implementation of interval methods for non-linear equation solving typically employs a combination of one or more of the concepts outlined above [21, 23, 44], perhaps also in connection with some manipulation of the equation system to be solved [20, 45, 46]. Often function range testing and constraint propagation techniques are first used to contract intervals, as these methods have low computational overhead. Then, more costly interval-Newton steps can be applied to the contracted intervals to obtain final solution enclosures. In most such equation-solving algorithms, the intervals can be treated as independent granules of data. Thus, parallel implementations of interval methods are generally apparent, though must be done with proper attention to load balancing in order to be most effective [47–49].

In the subsequent sections, we will look at some specific applications of interval methods for non-linear equation solving. The core steps in the algorithm used to solve these problems can be outlined as follows: for a given $\mathbf{X}^{(k)}$, (1) apply function range test; if $\mathbf{X}^{(k)}$ is not eliminated, then (2) apply hull consistency (this is done on a problem specific basis); if $\mathbf{X}^{(k)}$ is not eliminated, then (3) apply interval-Newton, using either the hybrid preconditioning technique of Gau and Stadtherr [39] or the linear programming method of Lin and Stadtherr [42]; if $\mathbf{X}^{(k)}$ is not eliminated, or a unique root in $\mathbf{X}^{(k)}$ not identified, then (4) bisect $\mathbf{X}^{(k)}$. This is only one possible way to implement an interval method for non-linear equation solving applications. However, it has proved to be effective on a wide variety of problems, some of which are discussed below. The applications considered next are purely equation-solving problems. However, since many optimization problems can easily be converted into an equivalent system of equations, the techniques described above are also often applied to problems requiring global optimization, typically in connection with some branch-and-bound procedure.

4.3 Modeling of Liquid–Liquid Phase Equilibrium

The modeling of phase behavior is a rich source of problems in which interval methods can play an important role, by ensuring that correct results are reliably obtained [50, 51]. Of interest is the development and use of models for predicting the number, type (liquid, vapor, or solid), and composition of the phases present at equilibrium for mixtures of chemical components at specified conditions. In model development, parameter estimation problems arise, which typically require solution of a non-convex optimization problem. Unfortunately, it is not uncommon to find that literature values for parameters are actually locally, but not globally, optimal [52]. Use of parameters that are not globally optimal may result in rejection of a model that would otherwise be accepted if globally optimal parameters were used. For the case of vapor–liquid equilibrium modeling, Gau and Stadtherr [52, 53] have used an interval method to guarantee that the globally optimal parameters are found. After models are developed, they are used to compute the phase equilibrium for mixtures of interest. This is another global optimization problem, the global minimization of the total Gibbs energy in the case of specified temperature and pressure. Again, it is not uncommon to find literature solutions that are only locally optimal, and thus do not represent stable equilibrium states [51]. For the phase stability and equilibrium problems, and for related phase behavior calculations, there have been a number of successful applications of interval methods to the underlying equation-solving and optimization problems [50, 51, 54–67].

In this section, we will focus on the problem of parameter estimation in the modeling of liquid–liquid equilibrium. This can be formulated as a non-linear equation solving problem involving only two equations and variables. However, the number of solutions to this system is unknown *a priori*, and it is not uncommon to see incorrect solutions reported in the literature.

4.3.1 Problem Formulation

Consider liquid–liquid equilibrium in a two-component system at fixed temperature and pressure. For this case, the necessary and sufficient condition for equilibrium is that the total Gibbs energy be at a global minimum. The first-order optimality conditions on the Gibbs energy lead to the equal activity conditions,

$$a_i^I = a_i^{II}, \quad i = 1, 2, \quad (4)$$

stating the equality of activities of each component (1 and 2) in each phase (I and II). This is a necessary, but not sufficient, condition for equilibrium. Given an activity coefficient model ($a_i = \gamma_i x_i$), expressed in terms of observable component mole fractions x_1 and $x_2 = 1 - x_1$, and activity coefficients γ_1 and γ_2 expressed in terms of composition and two binary parameters θ_{12} and θ_{21} , then the equal activity conditions can be expressed as

$$x_i^I \gamma_i^I (x_1^I, x_2^I, \theta_{12}, \theta_{21}) = x_i^{II} \gamma_i^{II} (x_1^{II}, x_2^{II}, \theta_{12}, \theta_{21}), \quad i = 1, 2. \quad (5)$$

Experimental measurements of the compositions of both phases are available. Thus, in equation (5), the values of x_1^I , x_2^I , x_1^{II} , and x_2^{II} are fixed. This results in a system of two equations in the two parameters θ_{12} and θ_{21} . This provides a widely used approach for parameter estimation in activity coefficient models for liquid–liquid equilibrium [68, 69], as generally it is possible to use physical grounds to reject all but one solution to equation (5). Parameter solutions are generally sought using local methods with multistart. A curve-following approach can also be used [70], but its reliability is step-size dependent and is not guaranteed. In this section, we will use an interval-Newton approach, as outlined at the end of Section 4.2, to determine reliably all solutions to equation (5) for the case in which the Non-Random Two-Liquid (NRTL) activity coefficient model is used.

In the NRTL model, the activity coefficients for use in equation (5) are given by

$$\ln \gamma_1 = x_2^2 \left[\tau_{21} \left(\frac{G_{21}}{x_1 + x_2 G_{21}} \right)^2 + \frac{\tau_{12} G_{12}}{(x_2 + x_1 G_{12})^2} \right] \quad (6)$$

$$\ln \gamma_2 = x_1^2 \left[\tau_{12} \left(\frac{G_{12}}{x_2 + x_1 G_{12}} \right)^2 + \frac{\tau_{21} G_{21}}{(x_1 + x_2 G_{21})^2} \right], \quad (7)$$

where

$$\tau_{12} = \frac{g_{12} - g_{22}}{RT} = \frac{\Delta g_{12}}{RT}$$

$$\tau_{21} = \frac{g_{21} - g_{11}}{RT} = \frac{\Delta g_{21}}{RT}$$

$$G_{12} = \exp(-\alpha_{12} \tau_{12})$$

$$G_{21} = \exp(-\alpha_{21} \tau_{21}).$$

Here, g_{ij} is an energy parameter characteristic of the $i-j$ interaction, and the parameter $\alpha = \alpha_{12} = \alpha_{21}$ is related to the non-randomness in the mixture. The non-randomness parameter α is frequently taken to be fixed when modeling liquid–liquid equilibrium. The binary parameters that must be determined from experimental data are then $\theta_{12} = \Delta g_{12}$ and $\theta_{21} = \Delta g_{21}$.

Table 4.1 Comparison of NRTL parameter estimates for the mixture of *n*-butanol and water ($\alpha = 0.4$, $T = 363$ K)^a

Solution	Reference [71]		Interval method	
	τ_{12}	τ_{21}	τ_{12}	τ_{21}
1	0.0075	3.8021	0.0075	3.8021
2	10.182	3.8034	10.178	3.8034
3	-73.824	-15.822		

^aThe parameter estimates are obtained by Heidemann and Mandhane [71] and by the use of an interval method.

4.3.2 *n*-Butanol and Water

Consider a mixture of *n*-butanol (component 1) and water (component 2) at $T = 363$ K and atmospheric pressure. Liquid–liquid phase equilibrium is observed experimentally with phase compositions $x_1^l = 0.020150$ and $x_1^{ll} = 0.35970$. Heidemann and Mandhane [71] modeled this system using NRTL with $\alpha = 0.4$. They obtained three solutions for the binary parameters, as shown in Table 4.1, in terms of τ_{12} and τ_{21} . Applying the interval method to solve this system of non-linear equations, with an initial search interval of $\theta_{12} \in [-1 \times 10^6, 1 \times 10^6]$ and $\theta_{21} \in [-1 \times 10^6, 1 \times 10^6]$, we find only two solutions, as also shown in Table 4.1. The extra solution found by Heidemann and Mandhane [71] is well within the search space used by the interval method and, so is clearly a spurious solution resulting from numerical difficulties in the local method used to solve equation (5). This can be verified by direct substitution of solution 3 into the equal activity conditions. When equal activity is expressed in the form of equation (5), the residuals for solution 3 are close to zero. However, when equal activity is expressed in terms of $\ln x_i$ and $\ln \gamma_i$, by taking the logarithm of both sides of equation (5), it becomes clear that the residuals for solution 3 are not really zero.

4.3.3 1,4-Dioxane and 1,2,3-Propanetriol

Consider a mixture of 1, 4-dioxane (component 1) and 1, 2, 3-propanetriol at $T = 298$ K and atmospheric pressure. Liquid–liquid phase equilibrium is observed experimentally with phase compositions $x_1^l = 0.2078$ and $x_1^{ll} = 0.9934$. Mattelin and Verhoeve [72] modeled this system using NRTL with various values of α . We will focus on the case of $\alpha = 0.15$. They obtained six solutions for the binary parameters, which are reported graphically without giving exact numerical values. Applying the interval method, with the same initial search interval as given above, we find only four solutions, as shown in Table 4.2 in terms of τ_{12} and τ_{21} . The extra solutions found by Mattelin and Verhoeve [72] are well within the search space

Table 4.2 NRTL parameter estimates for the mixture 1, 4-dioxane and 1, 2, 3-propanetriol ($\alpha = 0.15$ and $T = 298.15$ K)^{a,b}

Solution	Interval method	
	τ_{12}	τ_{21}
1	5.6379	-0.59940
2	13.478	-82.941
3	38.642	13.554
4	39.840	3.0285

^a Estimates are found using an interval method.

^b Mattelin and Verhoeve [72] reported finding six solutions.

used by the interval method. Again, it appears that numerical difficulties in the use of local methods has led to spurious solutions.

4.3.4 Remarks

In this section, we have seen a small non-linear equation system that in some cases is numerically difficult to solve using standard local methods, as evident from the reporting of spurious roots in the literature. Using an interval-Newton method, tight enclosures of all roots in the initial search space could be found very easily and efficiently, with computational times on the order of seconds (3.2-GHz Intel Pentium 4).

4.4 Transition-State Analysis

In molecular modeling, the search for features on a potential energy hypersurface is often required and is a very challenging computational problem. In some cases, finding a global minimum is required, but the existence of a very large number of local minima, the number of which may increase exponentially with the size of a molecule or the number of molecules, makes the problem extremely difficult. Interval methods can play a role in solving these problems [73, 74], but are limited in practice to problems of relatively low dimension. In other problems in computational chemistry, it is desired to find all stationary points. Interval methods for equation solving have been applied to one such problem, involving the use of lattice density functional theory to model adsorption in a nanoscale pore, by Maier and Stadtherr [75]. Another such problem is transition-state analysis, as summarized below, and is described in more detail by Lin and Stadtherr [76].

Transition-state theory is a well-established method which, by providing an approach for computing the kinetics of infrequent events, is useful in the study of numerous physical systems. Of particular interest here is the problem of computing the diffusivity of a sorbate molecule in a zeolite. This can be done using transition-state analysis, as described by June *et al.* [77]. It is assumed that diffusive motion of the sorbate molecules through the zeolite occurs by a series of uncorrelated hops between potential energy minima in the zeolite lattice. A sorption state or site is constructed around each minimum of the potential energy hypersurface. Any such pair of sites i and j is then assumed to be separated by a dividing surface on which a saddle point of the potential energy hypersurface is located. The saddle point can be viewed as the transition state between sites, and a pair of steepest decent paths from the saddle point connects the minima associated with the i and j sites. Obviously, in this application, and in other applications of transition-state theory, finding all local minima and saddle points of the potential energy surface, \mathcal{V} , is critical. We show here, using a sorbate–zeolite system, the use of an interval-Newton method, as outlined at the end of Section 4.2, to find all stationary points of a potential energy surface.

Stationary points satisfy the condition $\mathbf{g} = \nabla\mathcal{V} = \mathbf{0}$; that is, at a stationary point, the gradient of the potential energy surface is zero. Using the eigenvalues of $H = \nabla^2\mathcal{V}$, the Hessian of the potential energy surface, stationary points can be classified into local minima, local maxima, and saddle points (of order determined by the number of negative eigenvalues). There are a number of methods for locating stationary points. A Newton or quasi-Newton method, applied to solve the non-linear equation system $\nabla\mathcal{V} = \mathbf{0}$, yields a solution whenever the initial guess is sufficiently close to a stationary point. This method can be used in an exhaustive search, using many different initial guesses, to locate stationary points. The set of initial guesses to use might be determined by the user (intuitively or arbitrarily) or by some type of stochastic multistart approach. Another popular approach is the use of eigenmode-following methods, as done, e.g., by Tsai and Jordan [78]. These methods can be regarded as variations of Newton's method. In an eigenmode-following algorithm, the Newton step is modified by shifting some of the eigenvalues of the Hessian (from positive to negative or vice versa). By selection of the shift parameters, one can effectively find the desired type of stationary points, e.g., minima and first-order saddles. There are also a number of other approaches, many involving some stochastic component, for finding stationary points.

In the context of sorbate–zeolite systems, June *et al.* [77] use an approach in which minima and saddle points are located separately. A three-step process is employed in an exhaustive search for minima. First, the volume of the search space (one asymmetric unit) is discretized by a grid with a spacing of approximately 0.2 Å, and the potential and gradient vector are tabulated on the grid. Second, each cube

formed by a set of nearest-neighbor grid nodes is scanned and the three components of the gradient vector on the eight vertices of the cube are checked for changes in sign. Finally, if all three components are found to change sign on two or more vertices of the cube, a Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton minimization search algorithm is initiated to locate a local minimum, using the coordinates of the center of the cube as the initial guess. Two different algorithms are tried for determining the location of saddle points. One searches for global minimizers in the function $\mathbf{g}^T \mathbf{g}$, i.e., the sum of the squares of the components of the gradient vector. The other algorithm, due to Baker [79], searches for saddle points directly from an initial point by maximizing the potential energy along the eigenvector direction associated with the smallest eigenvalue and by minimizing along directions associated with all other eigenvalues of the Hessian.

All the methods discussed above have a major shortcoming. They provide no guarantee that *all* local minima and saddle points of interest will actually be found. One approach to resolving this difficulty is given by Westerberg and Floudas [80], who transform the equation-solving problem $\nabla \mathcal{V} = \mathbf{0}$ into an equivalent optimization problem that has global minimizers corresponding to the solutions of the equation system (i.e., the stationary points of \mathcal{V}). A deterministic global optimization algorithm, based on a branch-and-bound strategy with convex underestimators, is then used to find these global minimizers. Whether all stationary points are actually found depends on proper choice of a parameter (α) used in obtaining the convex underestimators, and Westerberg and Floudas do not use a method that guarantees a proper choice. However, there do exist techniques [81, 82], based on an interval representation of the Hessian, that in principle could be used to guarantee a proper value of α , though likely at considerable computational expense. We demonstrate here an approach in which interval analysis is applied directly to the solution of $\nabla \mathcal{V} = \mathbf{0}$ using an interval-Newton methodology. This provides a mathematical and computational *guarantee* that *all* stationary points of the potential energy surface are found (or, more precisely, enclosed within an arbitrarily small interval).

4.4.1 Problem Formulation

Zeolites are materials in which AlO_4 and SiO_4 tetrahedra are the building blocks of a variety of complex porous structures characterized by interconnected cavities and channels of molecular dimensions [83]. Silicalite contains no aluminum and thus no cations. This has made it a common and convenient choice as a model zeolite system. The crystal structure of silicalite, well known from X-ray diffraction studies [84], forms a three-dimensional interconnected pore network through which a sorbate molecule can diffuse. In this work, the phase with orthorhombic symmetry is considered, and a rigid lattice model, in which all silicon and oxygen atoms in the zeolite framework are occupying fixed positions and there is perfect crystallinity, is assumed. One spherical sorbate molecule (united atom) will be placed in the lattice, corresponding to infinitely dilute diffusion. The system comprises 27 unit cells, each of which is $20.07 \times 19.92 \times 13.42 \text{ \AA}$ with 96 silicon atoms and 192 oxygen atoms.

All interactions between the sorbate and the oxygen atoms of the lattice are treated atomistically with a truncated Lennard-Jones 6–12 potential. That is, for the interaction between the sorbate and oxygen atom i , the potential is given by

$$\mathcal{V}_i = \begin{cases} \frac{a}{r_i^{12}} - \frac{b}{r_i^6} & r_i < r_{\text{cut}} \\ 0 & r_i \geq r_{\text{cut}}, \end{cases} \quad (8)$$

where a is a repulsion parameter, b is an attraction parameter, r_{cut} is the cutoff distance, and r_i is the distance between the sorbate and oxygen atom i . This distance is given by

$$r_i^2 = (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2, \quad (9)$$

where (x, y, z) are the Cartesian coordinates of the sorbate, and $(x_i, y_i, z_i), i = 1, \dots, N$, are the Cartesian coordinates of the N oxygen atoms. The silicon atoms, being recessed within the SiO_4 tetrahedra, are

neglected in the potential function. Therefore, the total potential energy, \mathcal{V} , of a single sorbate molecule in the absence of neighboring sorbate molecules is represented by a sum over all lattice oxygens;

$$\mathcal{V} = \sum_{i=1}^N \mathcal{V}_i. \quad (10)$$

The interval-Newton approach is applied to determine the sorbate locations (x, y, z) that are stationary points on the potential energy surface \mathcal{V} given by equation (10), i.e., to solve the non-linear equation system $\nabla\mathcal{V} = \mathbf{0}$. To achieve tighter interval extensions of the potential function and its derivatives, and thus improve the performance of the interval-Newton method, the mathematical properties of the Lennard-Jones potential and its first- and second-order derivatives can be exploited, as described in detail by Lin and Stadtherr [76].

4.4.2 Results and Discussion

Due to the orthorhombic symmetry of the silicalite lattice, the search space for stationary points is only one asymmetric unit, $[0, 10.035] \times [0, 4.98] \times [0, 13.42]$ Å, which is one-eighth of a unit cell. This defines the initial interval for the interval-Newton method, namely $X^{(0)} = [0, 10.035]$ Å, $Y^{(0)} = [0, 4.98]$ Å, and $Z^{(0)} = [0, 13.42]$ Å. Following June *et al.* [77], stationary points with extremely high potential, such as $\mathcal{V} > 0$, will not be sought. To do this, we calculate the interval extension of \mathcal{V} over the interval currently being tested. If its lower bound is greater than zero, the current interval is discarded.

Using the interval-Newton method, with the linear programming strategy of Lin and Stadtherr [42], a total of 15 stationary points were found in a computation time of 724 s (1.7-GHz Intel Xeon). The locations of the stationary points, their energy values, and their types are provided in Table 4.3. Five local minima were found, along with eight first-order saddle points and two second-order saddle points. June *et al.* [77] report the same five local minima, as well as nine of the ten saddle points. They do not report finding the lower energy second-order saddle point (saddle point #14 in Table 4.3).

The second-order saddle point #14, not reported by June *et al.* [77], is very close to the first-order saddle point #13 and is slightly lower in energy. Apparently, neither of the two methods tried by June *et al.* [77] was able to locate this point. The first method they tried uses the same grid-based optimization

Table 4.3 Stationary points of the potential energy surface of xenon in silicalite

No.	Type	Energy (kcal/mol)	x (Å)	y (Å)	z (Å)
1	Minimum	-5.9560	3.9956	4.9800	12.1340
2	Minimum	-5.8763	0.3613	0.9260	6.1112
3	Minimum	-5.8422	5.8529	4.9800	10.8790
4	Minimum	-5.7455	1.4356	4.9800	11.5540
5	Minimum	-5.1109	0.4642	4.9800	6.0635
6	First order	-5.7738	5.0486	4.9800	11.3210
7	First order	-5.6955	0.0000	0.0000	6.7100
8	First order	-5.6060	2.3433	4.9800	11.4980
9	First order	-4.7494	0.1454	3.7957	6.4452
10	First order	-4.3057	9.2165	4.9800	11.0110
11	First order	-4.2380	0.0477	3.9147	8.3865
12	First order	-4.2261	8.6361	4.9800	12.8560
13	First order	-4.1405	0.5925	4.9800	8.0122
14	Second order	-4.1404	0.5883	4.8777	8.0138
15	Second order	-4.1027	9.1881	4.1629	11.8720

scheme used to locate local minima in \mathcal{V} , but instead applied to minimize $\mathbf{g}^T \mathbf{g}$. However, stationary points #13 and #14 are approximately 0.1\AA apart, while the grid spacing they used was approximately 0.2\AA . This illustrates the danger in using grid-based schemes for finding all solutions to a problem. By using the interval methods described here, one never needs to be concerned about whether a grid spacing is fine enough to find all solutions. The second method they tried was Baker's algorithm [79], as described briefly above, but it is unclear how they initialized the algorithm. A key advantage of the interval method is that no point initialization is required. Only an initial *interval* must be supplied, here corresponding to one asymmetric unit, and this is determined by the geometry of the zeolite lattice. Thus, in this context, the interval method is initialization independent.

4.4.3 Remarks

Lin and Stadtherr [42] have also studied two other sorbate–zeolite systems and used the interval method to find all stationary points on the potential energy surfaces. While we have concentrated here on problems involving transition-state analysis of diffusion in zeolites, we anticipate that the method will be useful in many other types of problems in which transition-state theory is applied.

4.5 Food Web Models

Ecological models, including models of food webs, are being increasingly used as aids in the management and assessment of ecological risks. As a first step in using a food web model, an understanding is needed of the predicted equilibrium states (steady states) and their stability. To determine the equilibrium states, a system of non-linear equations must be solved, with the number of solutions often not known a priori. Finding bifurcations of equilibria (parameter values at which the number of equilibrium states or their stability changes) is another problem of interest, which can also be formulated as a non-linear equation solving problem. For both these problems, continuation methods are typically used, but are initialization dependent and provide no guarantees that all solutions will be found. Gwaltney *et al.* [85] and Gwaltney and Stadtherr [86] have demonstrated the use of an interval-Newton method to find equilibrium states and their bifurcations for some simple food chain models. Interval methods have also been successfully applied to the problem of locating equilibrium states and singularities in traditional chemical engineering problems, such as reaction and reactive distillation systems [87–90]. We will consider here a seven-species food web and use an interval-Newton approach, as outlined at the end of Section 4.2, to solve for all steady states predicted by the model.

4.5.1 Problem Formulation

The seven-species food web is shown schematically in Figure 4.1. It involves two producers (species 1 and 2) and five consumers (species 3–7). The producers are assumed to grow logistically, while the consumers obey predator response functions that will be specified below.

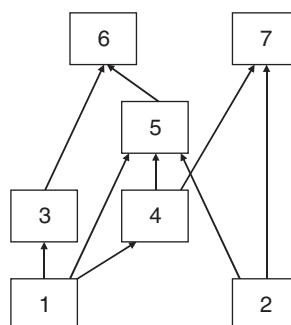


Figure 4.1 Diagram illustrating the predation relationships in the seven-species food web model

The model equations (balance equations) are, for $i = 1, \dots, 7$,

$$f_i(\mathbf{m}) = \frac{dm_i}{dt} = m_i g_i(\mathbf{m}) = m_i \left[r_i + \sum_{j=1}^7 a_{ij} p_{ij}(\mathbf{m}) \right]. \quad (11)$$

Here the variables are the species biomasses $m_i, i = 1, \dots, 7$, which are the components of the biomass vector \mathbf{m} . The constants a_{ij} represent combinations of different model parameters, and also indicate the structure of the food web. The constants r_i consist of intrinsic growth and death rate parameters. The functions $p_{ij}(\mathbf{m})$ are determined by the choice of predator response function for the predator-prey interaction involving species i and j . For the 1-3 interaction, we assume a hyperbolic response function (Holling type II). This leads to $p_{13}(\mathbf{m}) = m_3/(m_1 + B_{13})$ and $p_{31}(\mathbf{m}) = m_1/(m_3 + B_{13})$, where B_{13} is the half-saturation constant for consumption of species 1 by species 3. For all other interactions, we assume a linear response function (Lotka-Volterra), giving $p_{ij}(\mathbf{m}) = m_j$. Values of all constants in the model are given by Gwaltney [91]. To determine the equilibrium states predicted by this model, solution of the non-linear equation system

$$f_i(\mathbf{m}) = m_i g_i(\mathbf{m}) = 0, \quad i = 1, \dots, 7, \quad (12)$$

is required.

4.5.2 Results and Discussion

There are two basic strategies for solving the equation system. In the *simultaneous* strategy, we simply solve equation (12) directly as a system of seven equations in seven variables. In the *sequential* strategy, a sequence of smaller problems is solved, one for each feasible zero–non-zero state. A set of feasible zero–non-zero states can be constructed from the structure of the food web. For example, the state [1030060] (indicating that species 1, 3, and 6 have non-zero biomasses and that species 2, 4, 5, and 7 are absent) is feasible. However, the state [1204067] is not feasible, since in the absence of species 3 and 5 species 6 cannot exist. For a relatively small food web, it is not difficult to construct the set of feasible zero–non-zero states. However, for large food webs this is non-trivial, as the number of such states can become very large. For the seven-species web of interest here, there are 55 feasible zero–non-zero states. For each zero–non-zero state, an equation system is formulated to solve for the corresponding steady states. For example, for the [1030060] state, $m_1 \neq 0$; thus, it is required that $g_1 = 0$. Similarly, $g_3 = 0$ and $g_6 = 0$. This provides three equations in the three non-zero variables m_1, m_3 , and m_6 . The remaining components of equation (12) are satisfied because $m_2 = m_4 = m_5 = m_7 = 0$.

An interval-Newton approach was used to solve the non-linear equation system (12) in connection with both the simultaneous and sequential approaches. This was done for several different values of the model parameter K_2 , the carrying capacity for producer species 2. A partial set of results (m_1 and m_2 only) is shown in Figure 4.2. It is clear that for a particular value of K_2 , there are often several steady states. When non-linear predator response functions are used, the number of steady states is also unknown a priori. The interval method provides a means to guarantee that all steady-state solutions will be found. When the simultaneous approach was used, and a single 7×7 equation system solved, the CPU time required for each value of K_2 averaged about 60 s (3.2-GHz Pentium 4). When the sequential approach was used, and a sequence of many smaller systems solved, the CPU time required for each value of K_2 averaged about 0.02 s. Clearly, it is much more effective to use a sequential strategy. For further discussion of this problem and an interpretation of the results, see Gwaltney [91].

4.5.3 Remarks

In computing the equilibrium states in non-linear food web models, it is possible to have a very large number of solutions. For example, Gwaltney [91] also considered a food web with 12 species and explicit resource dynamics (4 nutrients). For some sets of parameter values, well over 300 steady-state solutions

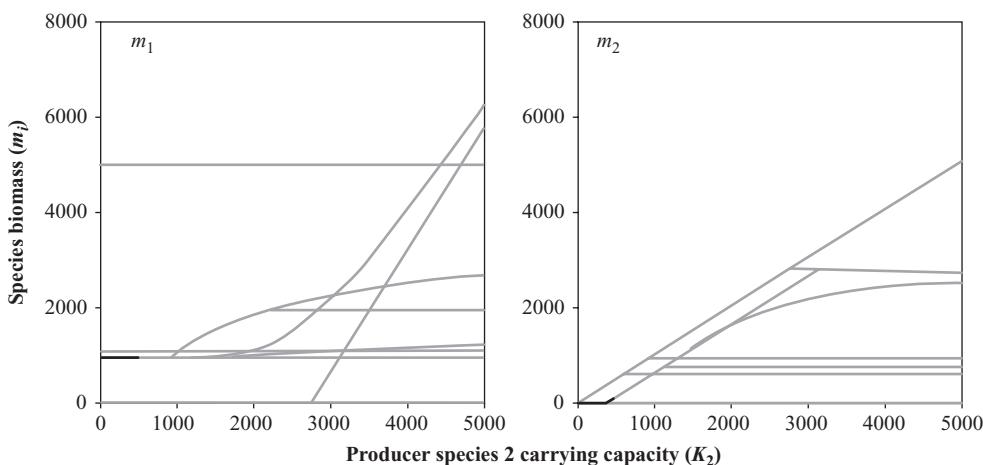


Figure 4.2 Solution branch diagrams illustrating the change in the steady-state biomass values of species 1 (m_1) and species 2 (m_2) with change in species 2 carrying capacity (K_2) for the seven-species food web model. Black lines indicate stable equilibria. Gray lines indicate unstable equilibria

were found by using the sequential approach with an interval-Newton method. In cases for which a large number of solutions is possible, and the number of solutions is not known, the use of interval methods for non-linear equation solving is an attractive approach for ensuring that no solutions will be missed.

4.6 Concluding Remarks

In the examples presented here, we have shown that an interval method for non-linear equation solving, in particular an approach incorporating the interval-Newton method, is a powerful approach for the solution of systems of non-linear equation systems. The method provides a mathematical and computational guarantee that *all* solutions within a specified initial interval are enclosed. Continuing improvements in solution methods, together with advances in software and hardware for the use of intervals, will make this an increasingly attractive problem-solving tool.

The verification provided by the interval approach comes at the expense of additional computation time. Essentially one has a choice between fast methods that may give an incorrect or incomplete answer, or a slower method that is guaranteed to give the correct results. Thus, a modeler may need to consider the trade-off between the additional computing time and the risk of getting the wrong answer to a problem. Certainly, for ‘mission critical’ situations, the additional computing expense is well spent.

Acknowledgments

This work was supported in part by the Department of Education Graduate Assistance in Areas of National Needs (GAANN) Program under Grant #P200A010448, by the donors of the Petroleum Research Fund, administered by the ACS, under Grant 35979-AC9, by the State of Indiana 21st Century Research and Technology Fund under Grant #909010455, and by the National Oceanic and Atmospheric Administration under Grant #NA050AR4601153.

References

- [1] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer Academic Publishers, Norwell, MA, 2003.
- [2] R.E. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1966.

- [3] E.R. Hansen and G.W. Walster. *Global Optimization Using Interval Analysis*. Marcel Dekker, New York, 2004.
- [4] J.D. Pryce and G.F. Corliss. Interval arithmetic with containment sets. *Computing* 78 (2006) 251–276.
- [5] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter. *Applied Interval Analysis*. Springer-Verlag, London, 2001.
- [6] R.B. Kearfott. Interval computations: introduction, uses, and resources. *Euromath. Bull.* 2 (1996) 95–112.
- [7] R.B. Kearfott. *Rigorous Global Search: Continuous Problems*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.
- [8] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, UK, 1990.
- [9] R.B. Kearfott, M. Dawande, K. Du, and C. Hu. INTLIB: a portable Fortran-77 elementary function library. *Interval Comput.* 3 (1992) 96–105.
- [10] R.B. Kearfott, M. Dawande, K. Du, and C. Hu. Algorithm 737: INTLIB: a portable Fortran-77 elementary function library. *ACM Trans. Math. Softw.* 20 (1994) 447–459.
- [11] R.B. Kearfott. Algorithm 763; INTERVAL_ARITHMETIC: a Fortran-90 module for an interval data type. *ACM Trans. Math. Softw.* 22 (1996) 385–392.
- [12] PROFIL/BIAS, <http://www.ti3.tu-harburg.de/Software/PROFILEnglisch.html>, accessed January 15, 2008.
- [13] M. Lerch, G. Tischler, J. Wolff von Gudenberg, W. Hofschuster, and W. Krämer. FILIB++, a fast interval library supporting containment computations. *ACM Trans. Math. Softw.* 32 (2006) 299–324.
- [14] S.M. Rump. INTLAB – INTerval LABoratory. In: T. Csendes (ed), *Developments in Reliable Computing*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999, pp. 77–104.
- [15] H. Ratschek and J. Rokne. *Computer Methods for the Range of Functions*. Ellis Horwood, Chichester, UK, 1984.
- [16] E. Hansen. Sharpening interval computations. *Reliab. Comput.* 12 (2006) 21–34.
- [17] V.M. Nesterov. How to use monotonicity-type information to get better estimates of the range of real-valued functions. *Interval Comput.* 4 (1993) 3–12.
- [18] K. Makino and M. Berz. Efficient control of the dependency problem based on Taylor model methods. *Reliab. Comput.* 5 (1999) 3–12.
- [19] A. Neumaier. Taylor forms – use and limits. *Reliab. Comput.* 9 (2002) 43–79.
- [20] K. Yamamura. Finding all solutions of non-linear equations using linear combinations of functions. *Reliab. Comput.* 6 (2000) 105–113.
- [21] R.B. Kearfott. Validated constraint solving – practicalities, pitfalls, and new developments. *Reliab. Comput.* 11 (2005) 383–391.
- [22] S. Herfort and D. Ratz. *Improving the Efficiency of a Non-linear-System-Solver Using a Componentwise Newton Method*. Bericht 2/1997. Institut für Angewandte Mathematik, Universität Karlsruhe (TH), Karlsruhe, Germany, 1997.
- [23] L. Granvilliers. On the combination of interval constraint solvers. *Reliab. Comput.* 7 (2001) 467–483.
- [24] L. Granvilliers and F. Benhamou. Algorithm 852: RealPaver: an interval solver using constraint satisfaction techniques. *ACM. Trans. Math. Softw.* 32 (2006) 138–156.
- [25] P. van Hentenryck, L. Michel, and Y. Deville. *Numerica: A Modeling Language for Global Optimization*. The MIT Press, Cambridge, MA, 1997.
- [26] ICOS, <http://ylebbah.googlepages.com/icos>, accessed February 26, 2008.
- [27] G. Alefeld. Interval arithmetic tools for range approximation and inclusion of zeros. In: H. Bulgak and C. Zenger (eds), *Error Control and Adaptivity in Scientific Computing*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999, pp. 1–21.
- [28] R.E. Moore. A test for existence of solutions to non-linear systems. *SIAM J. Numer. Anal.* 14 (1977) 611–615.
- [29] R.E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.
- [30] S.P. Shary. Krawczyk operator revised. In: *Proceedings of International Conference on Computational Mathematics ICCM-2004*, Novosibirsk, Russia, June 21–25, 2004, Institute of Computational Mathematics and Mathematical Geophysics (ICM&MG), 2004.
- [31] L. Simcik and P. Linz. Boundary-based interval Newton's method. *Interval Comput.* 4 (1993) 89–99.
- [32] K. Min, L. Qi, and S. Zuhe. On the componentwise Krawczyk-Moore iteration. *Reliab. Comput.* 5 (1999) 359–370.
- [33] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, New York, 1983.
- [34] N.S. Dimitrova and S.M. Markov. On validated Newton type method for non-linear equations. *Interval Comput.* 1994(2) (1994) 27–51.
- [35] E. Hansen and S. Sengupta. Bounding solutions of systems of equations using interval-analysis. *BIT* 21 (1981) 203–211.
- [36] E.R. Hansen and R.I. Greenberg. An interval Newton method. *Appl. Math. Comput.* 12 (1983) 89–98.
- [37] R.B. Kearfott. Preconditioners for the interval Gauss-Seidel method. *SIAM J. Numer. Anal.* 27 (1990) 804–822.
- [38] R.B. Kearfott, C. Hu, and M. Novoa, III. A review of preconditioners for the interval Gauss-Seidel method. *Interval Comput.* (1) (1991) 59–85.

- [39] C.-Y. Gau and M.A. Stadtherr. New interval methodologies for reliable chemical process modeling. *Comput. Chem. Eng.* 26 (2002) 827–840.
- [40] R.B. Kearfott and M. Novoa III. Algorithm 681: INTBIS, a portable interval Newton/bisection package. *ACM Trans. Math. Softw.* 16 (1990) 152–157.
- [41] Y. Lin and M.A. Stadtherr. Advances in interval methods for deterministic global optimization in chemical engineering. *J. Glob. Optim.* 29 (2004) 281–296.
- [42] Y. Lin and M.A. Stadtherr. LP strategy for the interval-Newton method in deterministic global optimization. *Ind. Eng. Chem. Res.* 43 (2004) 3741–2749.
- [43] S.M. Rump. Verification methods for dense and sparse systems of equations. In: J. Herzberger (ed), *Topics in Validated Computations – Studies in Computational Mathematics*. Elsevier, Amsterdam, The Netherlands, 1994, pp. 63–135.
- [44] Y.G. Dolgov. Developing interval global optimization algorithms on the basis of branch-and-bound and constraint propagation methods. *Reliab. Comput.* 11 (2005) 343–358.
- [45] L.V. Kolev. A new method for global solution of systems of non-linear equations. *Reliab. Comput.* 4 (1998) 125–146.
- [46] L.V. Kolev. An improved method for global solution of non-linear systems. *Reliab. Comput.* 5 (1999) 103–111.
- [47] C.-Y. Gau and M.A. Stadtherr. Dynamic load balancing for parallel interval-Newton using message passing. *Comput. Chem. Eng.* 26 (2002) 811–825.
- [48] V. Kreinovich and A. Bernal. Parallel algorithms for interval computations: An introduction. *Interval Comput.* 3 (1994) 6–62.
- [49] C.A. Schnepper and M.A. Stadtherr. Application of a parallel interval Newton/generalized bisection algorithm to equation-based chemical process flowsheeting. *Interval Comput.* 4 (1993) 40–64.
- [50] G.I. Burgos-Solórzano, J.F. Brennecke, and M.A. Stadtherr. Validated computing approach for high-pressure chemical and multiphase equilibrium. *Fluid Phase Equilib.* 219 (2004) 245–255.
- [51] G. Xu, W.D. Haynes, and M.A. Stadtherr. Reliable phase stability analysis for asymmetric models. *Fluid Phase Equilib.* 235 (2005) 152–165.
- [52] C.-Y. Gau, J.F. Brennecke, and M.A. Stadtherr. Reliable non-linear parameter estimation in VLE modeling. *Fluid Phase Equilib.* 168 (2000) 1–18.
- [53] C.-Y. Gau and M.A. Stadtherr. Reliable nonlinear parameter estimation using interval analysis: error-in-variable approach. *Comput. Chem. Eng.* 24 (2000) 631–637.
- [54] J.Z. Hua, J.F. Brennecke, and M.A. Stadtherr. Reliable phase stability analysis for cubic equation of state models. *Comput. Chem. Eng.* 20 (1996) S395–S400.
- [55] J.Z. Hua, J.F. Brennecke, and M.A. Stadtherr. Reliable prediction of phase stability using an interval-Newton method. *Fluid Phase Equilib.* 116 (1996) 52–59.
- [56] J.Z. Hua, J.F. Brennecke, and M.A. Stadtherr. Enhanced interval analysis for phase stability: cubic equation of state models. *Ind. Eng. Chem. Res.* 37 (1998) pp. 1519–1527.
- [57] J.Z. Hua, J.F. Brennecke, and M.A. Stadtherr. Reliable computation of phase stability using interval analysis: Cubic equation of state models. *Comput. Chem. Eng.* 22 (1998) 1207–1214.
- [58] J.Z. Hua, R.W. Maier, S.R. Tessier, J.F. Brennecke, and M.A. Stadtherr. Interval analysis for thermodynamic calculations in process design: a novel and completely reliable approach. *Fluid Phase Equilib.* 158 (1999) 607–615.
- [59] R.W. Maier, J.F. Brennecke, and M.A. Stadtherr. Reliable computation of homogeneous azeotropes. *AIChE J.* 44 (1998) 1745–1755.
- [60] R.W. Maier, J.F. Brennecke, and M.A. Stadtherr. Reliable computation of reactive azeotropes. *Comput. Chem. Eng.* 24 (2000) 1851–1858.
- [61] K.I.M. McKinnon, C.G. Millar, and M. Mongeau. Global optimization for the chemical and phase equilibrium problem using interval analysis. In: C.A. Floudas and P.M. Pardalos (eds), *State of the Art in Global Optimization Computational Methods and Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.
- [62] A.M. Scurto, G. Xu, J.F. Brennecke, and M.A. Stadtherr. Phase behavior and reliable computation of high-pressure solid-fluid equilibrium with cosolvents. *Ind. Eng. Chem. Res.* 42 (2003) 6464–6475.
- [63] M.A. Stadtherr, C.A. Schnepper, and J.F. Brennecke. Robust phase stability analysis using interval methods. *AIChE Symp. Ser.* 91(304) (1995) 356–359.
- [64] B.A. Stradi, J.F. Brennecke, J.P. Kohn, and M.A. Stadtherr. Reliable computation of mixture critical points. *AIChE J.* 47 (2001) 212–221.
- [65] S.R. Tessier, J.F. Brennecke, and M.A. Stadtherr. Reliable phase stability analysis for excess Gibbs energy models. *Chem. Eng. Sci.* 55 (2000) 1785–1796.
- [66] G. Xu, J.F. Brennecke, and M.A. Stadtherr. Reliable computation of phase stability and equilibrium from the SAFT equation of state. *Ind. Eng. Chem. Res.* 41 (2002) 938–952.

- [67] G. Xu, A.M. Scurto, M. Castier, J.F. Brennecke, and M.A. Stadtherr. Reliable computational of high-pressure solid-fluid equilibrium. *Ind. Eng. Chem. Res.* 39 (2000) 1624–1636.
- [68] H. Renon and J.M. Prausnitz. Local compositions in thermodynamic excess functions for liquid mixtures. *AIChE J.* 14 (1968) 135–144.
- [69] J.M. Sørensen and W. Arlt. *Liquid-Liquid Equilibrium Data Collection*. Chemistry Data Series, Vol. V, Parts 1–3. DECHEMA, Frankfurt/Main, Germany, 1979–1980.
- [70] J. Jacq and L. Asselineau. Binary liquid-liquid equilibria. Multiple solutions for the NRTL equation. *Fluid Phase Equilib.* 14 (1983) 185–192.
- [71] R.A. Heidemann and J.M. Mandhane. Some properties of the NRTL equation in correlating liquid-liquid equilibrium data. *Chem. Eng. Sci.* 28 (1973) 1213–1221.
- [72] A.C. Mattelin and L.A.J. Verhoeve. The correlation of binary miscibility data by means of the NRTL equation. *Chem. Eng. Sci.* 30 (1975) 193–200.
- [73] C. Lavor. A deterministic approach for global minimization of molecular potential energy functions. *Int. J. Quantum Chem.* 95 (2003) 336–343.
- [74] Y. Lin and M.A. Stadtherr. Deterministic global optimization of molecular structures using interval analysis. *J. Comput. Chem.* 26 (2005) 1413–1420.
- [75] R.W. Maier and M.A. Stadtherr. Reliable density-functional-theory calculations of adsorption in nanoscale pores. *AIChE J.* 47 (2001) 1874–1884.
- [76] Y. Lin and M.A. Stadtherr. Locating stationary points of sorbate-zeolite potential energy surfaces using interval analysis. *J. Chem. Phys.* 121 (2004) 10159–10166.
- [77] R.L. June, A.T. Bell, and D.N. Theodorou. Transition-state studies of xenon and SF₆ diffusion in silicalite. *J. Phys. Chem.* 95 (1991) 8866–8878.
- [78] C.J. Tsai and K.D. Jordan. Use of an eigenmode method to locate the stationary-points on the potential-energy surfaces of selected argon and water clusters. *J. Phys. Chem.* 97 (1993) 11227–11237.
- [79] J. Baker. An algorithm for the location of transition-states. *J. Comput. Chem.* 7 (1986) 385–395.
- [80] K.M. Westerberg and C.A. Floudas. Locating all transition states and studying the reaction pathways of potential energy surfaces. *J. Chem. Phys.* 110 (1999) 9259–9295.
- [81] C.S. Adjiman, I.P. Androulakis, and C.A. Floudas. A global optimization method, α BB, for general twice-differentiable constrained NLPs – II. Implementation and computational results. *Comput. Chem. Eng.* 22 (1998) 1159–1179.
- [82] C.S. Adjiman, S. Dallwig, C.A. Floudas, and A. Neumaier. A global optimization method, α BB, for general twice-differentiable constrained NLPs – I. Theoretical advances. *Comput. Chem. Eng.* 22 (1998) 1137–1158.
- [83] J. Karger and D.M. Ruthven. *Diffusion in Zeolites and Other Microporous Solids*. Wiley, New York, 1992.
- [84] D.H. Olson, G.T. Kokotailo, S.L. Lawton, and W.M. Meier. Crystal structure and structure-related properties of ZSM-5. *J. Phys. Chem.* 85 (1981) 2238–2243.
- [85] C.R. Gwaltney, M.P. Styczynski, and M.A. Stadtherr. Reliable computation of equilibrium states and bifurcations in food chain models. *Comput. Chem. Eng.* 28 (2004) 1981–1996.
- [86] C.R. Gwaltney and M.A. Stadtherr. Reliable computation of equilibrium states and bifurcations in nonlinear dynamics. *Lect. Notes Comput. Sci.* 3732 (2006) 122–131.
- [87] C.H. Bischof, B. Lang, W. Marquardt, and M. Mönnigmann. Verified determination of singularities in chemical processes. Presented at SCAN 2000, 9th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics, Karlsruhe, Germany, September 18–22, 2000.
- [88] V. Gehrke and W. Marquardt. A singularity theory approach to the study of reactive distillation. *Comput. Chem. Eng.* 21 (1997) S1001–S1006.
- [89] M. Mönnigmann and W. Marquardt. Normal vectors on manifolds of critical points for parametric robustness of equilibrium solutions of ODE systems. *J. Non-linear Sci.* 12 (2002) 85–112.
- [90] C.A. Schnepper and M.A. Stadtherr. Robust process simulation using interval methods. *Comput. Chem. Eng.* 20 (1996) 187–199.
- [91] C.R. Gwaltney. *Reliable Location of Equilibrium States and Bifurcations in Non-linear Dynamical Systems with Applications in Food Web Modeling and Chemical Engineering*. Ph.D. Thesis. University of Notre Dame, Notre Dame, IN, 2006.

5

Fuzzy Sets as a User-Centric Processing Framework of Granular Computing

Witold Pedrycz

5.1 Introduction

This chapter serves as a general introduction to fuzzy sets being regarded as one of the key technologies of granular computing. Fuzzy sets are information granules modeled by the underlying concept of partial membership. Partial membership is crucial to a variety of everyday phenomena. Linguistic concepts are inherently non-binary. In this way fuzzy sets provide a badly needed formalism rooted in many-valued logic.

The material is organized in the following way. We start with some general observations (Section 5.2) by highlighting the origin of fuzzy sets, linking them to the concept of dichotomy, and underlying a central role of fuzzy sets in system modeling. In Section 5.3, we offer some generic descriptors of fuzzy sets (membership functions). Section 5.4 is concerned with the notion of granulation of information, where we elaborate on linkages between various formalisms being used. Characterizations of families of fuzzy sets are presented in Section 5.5. Next, in Section 5.6, we elaborate on some selected, yet highly representative, methods of membership function estimation by distinguishing between expert-driven and data-driven estimation methods. Granular modeling is presented in Section 5.7. Concluding comments are covered in Section 5.8.

5.2 General Observations

The concept of dichotomy becomes profoundly imprinted into our education, philosophy, and many branches of science, management, and engineering. While the formalism and vocabulary of Boolean concepts being effective in handling various discrimination processes involving binary quantification (yes–no, true–false) has been with us from the very beginning of our early education, in many cases it becomes evident that this limited, two-valued view at world could be overly simplified and in many circumstances may lack required rapport with the reality. In real world, there is nothing like black–white, good–bad, etc. All of us recognize that the notion of dichotomy is quite simple and does not look realistic. Concepts do not possess sharp boundaries. Definitions are not binary unless they tackle very

simple concepts (say odd–even numbers). Let us allude here to the observation being made by B. Russell (1923):

[T]he law of excluded middle is true when precise symbols are employed, but it is not true when symbols are vague, as, in fact, all symbols are.

In reality, we use terms whose complexity is far higher and which depart from the principle of dichotomy. Consider the notions used in everyday life such as *warm* weather, *low* inflation, and *long* delay. How could you define them if you were to draw a single line? Is 25°C *warm*? Is 24.9°C *warm*? Or is 24.95°C *warm* as well? Likewise in any image could you draw a single line to discriminate between objects such as sky, land, trees, and lake. Experimental data do not come in well-formed and distinct clusters; there are always some points in between well-delineated groups.

One might argue that those are concepts that are used in everyday language and therefore they need not possess any substantial level of formalism. Yet, one has to admit that the concepts that do not adhere to the principle of dichotomy are also visible in science, mathematics, and engineering. For instance, we often carry out a linear approximation of non-linear function and make a quantifying statement that such linearization is valid in some *small* neighborhood of the linearization point. Under these circumstances the principle of dichotomy does not offer too much.

The principle of dichotomy, or as we say an Aristotelian perspective at the description of the world, has been subject to a continuous challenge predominantly from the standpoint of philosophy and logic. Let us recall some of the most notable developments which have led to the revolutionary paradigm shift. Indisputably, the concept of a three-valued and multivalued logic put forward by Jan Lukasiewicz [1–3] and then pursued by others including Emil Post is one of the earliest and the most prominent logical attempts made toward the direction of abandoning the supremacy of the principle of dichotomy. As noted by Lukasiewicz, the question of the suitability or relevance of two-valued logic in evaluating the truth of propositions was posed in the context of those statements that allude to the future. ‘Tomorrow will rain,’ is this statement true? If we can answer this question, this means that we have already predetermined the future. We start to sense that this two-valued model, no matter how convincing it could be, is conceptually limited if not wrong. The non-Aristotelian view of the world was vividly promoted by Alfred Korzybski [4]. While the concept of the three-valued logic was revolutionary in 1920s, we somewhat quietly endorsed it over the passage of time. For instance, in database engineering, a certain entry may be two valued (yes–no) but the third option of ‘unknown’ is equally possible – here we simply indicate that no value of this entry has been provided.

In light of these examples, it becomes apparent that we need a suitable formalism to cope with these phenomena. Fuzzy sets offer an important and unique feature of describing information granules whose contributing elements may belong with varying degrees of membership (belongingness). This helps us describe the concepts that are commonly encountered in real word. The notions such as *low* income, *high* inflation, *small* approximation error, and many others are examples of concepts to which the yes–no quantification does not apply or becomes quite artificial and restrictive. We are cognizant that there is no way of quantifying the Boolean boundaries, as there are a lot of elements whose membership to the concept is only partial and quite different from 0 and 1.

The binary view of the world supported by set theory and two-valued logic has been vigorously challenged by philosophy and logic. The revolutionary step in logic was made by Lukasiewicz with his introduction of three and afterward multivalued logic [1]. It however took more decades to dwell on the ideas of the non-Aristotelian view of the world before fuzzy sets were introduced. This happened in 1965 with the publication of the seminal paper on fuzzy sets by Zadeh [5]. Refer also to other influential papers by Zadeh [6–12]. The concept of fuzzy set is surprisingly simple and elegant. Fuzzy set A captures its elements by assigning them to it with some varying degrees of membership. A so-called membership function is a vehicle that quantifies different degrees of membership. The higher the degree of membership $A(x)$, the stronger is the level of belongingness of this element to A [13–16].

The obvious, yet striking, difference between sets (intervals) and fuzzy sets lies in the notion of partial membership supported by fuzzy sets. In fuzzy sets, we discriminate between elements that are ‘typical’ to the concept and those of borderline character. Information granules such as *high* speed, *warm* weather,

and *fast* car are examples of information granules falling under this category that can be conveniently represented by fuzzy sets. As we cannot specify a single, well-defined element that forms a solid border between full belongingness and full exclusion, fuzzy sets offer an appealing alternative and a practical solution to this problem. Fuzzy sets with their smooth transition boundaries form an ideal vehicle to capture the notion of partial membership. In this sense information granules formalized in the language of fuzzy sets support a vast array of human-centric pursuits. They are predisposed to play a vital role when interfacing human to intelligent systems.

In problem formulation and problem solving, fuzzy sets may arise in two fundamentally different ways;

1. **Explicit.** Here, they typically pertain to some generic and fairly basic concepts we use in our communication and description of reality. There is a vast amount of examples, such as concepts being commonly used every day, say *short* waiting time, *large* data set, *low* inflation, *high* speed, *long* delay, etc. All of them are quite simple as we can easily capture their meaning. We can easily identify a universe of discourse over which such variables are defined. For instance, this could be time, number of records, velocity, and alike.
2. **Implicit.** Here, we are concerned with more complex and inherently multifaceted concepts and notions where fuzzy sets could be incorporated into the formal description and quantification of such problems yet not in so instantaneous manner. Some examples could include concepts such as '*preferred* car,' '*stability* of the control system,' '*high-performance* computing architecture,' '*good convergence* of the learning scheme,' '*strong* economy,' etc. All these notions incorporate some components that could be quantified with the use of fuzzy sets, yet this translation is not that completely straightforward and immediate as it happens for the category of the explicit usage of fuzzy sets. For instance, the concept of '*preferred* car' is evidently multifaceted and may involve a number of essential descriptors that when put together are really reflective of the notion we have in mind. For instance, we may involve a number of qualities such as speed, economy, reliability, depreciation, maintainability, and alike. Interestingly, each of these features could be easily rephrased in simpler terms and through this process at some level of this refinement phase we may arrive at fuzzy sets that start to manifest themselves in an explicit manner.

As we stressed, the omnipresence of fuzzy sets is surprising. Even going over any textbook or research monograph, not mentioning newspapers and magazines, we encounter a great deal of fuzzy sets coming in their implicit or explicit format. Table 5.1 offers a handful of selected examples.

From the optimization standpoint, the properties of continuity and commonly encountered differentiability of the membership functions becomes a genuine asset. We may easily envision situations where those information granules incorporated as a part of the neurofuzzy system are subject to optimization – hence the differentiability of their membership functions becomes of critical relevance. What becomes equally important is the fact that fuzzy sets bridge numeric and symbolic concepts. On one hand, fuzzy set can be treated as some symbol. We can regard it as a single conceptual entity by assigning to it some symbol, say L (for *low*). In the sequel, it could be processed as a purely symbolic entity. On the other hand, a fuzzy set comes with a numeric membership function and these membership grades could be processed in a numeric fashion.

Fuzzy sets can be viewed from several fundamentally different standpoints. Here we emphasize the three of them that play a fundamental role in processing and knowledge representation.

As an Enabling Processing Technology of Some Universal Character and of Profound Human-Centric Character

Fuzzy sets build on the existing information technologies by forming a user-centric interface, using which one could communicate essential design knowledge, thus guiding problem solving and making it more efficient. For instance, in signal processing and image processing we might incorporate a collection of rules capturing specific design knowledge about filter development in a certain area. Say, 'if the level of noise is *high*, consider using a *large* window of averaging.' In control engineering, we may incorporate some domain knowledge about the specific control objectives. For instance, 'if the constraint of fuel

Table 5.1 Examples of concepts whose description and processing invoke the use of the technology of fuzzy sets and granular computing

-
- p. 65: *small* random errors in the measurement vector ...
 - p. 70: The success of the method depends on whether the first initial guess is already *close enough* to the global minimum ...
 - p. 72: Hence, the convergence region of a numerical optimizer will be *large*
- F. van der Heijden *et al.* *Classification, Parameter Estimation and State Estimation*. J. Wiley Chichester, 2004.
- p. 162: Comparison between bipolar and MOS technology (a part of the table)

	bipolar	MOS
integration	<i>low</i>	<i>very high</i>
power	<i>high</i>	<i>low</i>
cost	<i>low</i>	<i>low</i>

R.H. Katz and G. Borriello. *Contemporary Logic Design*, 2nd ed. Prentice Hall, Upper Saddle River, NJ, 2005.

- p. 50: validation costs are *high* for *critical systems*
- p. 660: A *high* value for fan-in means that X is *highly coupled* to the rest of the design and changes to X will have extensive knock-on effect. A *high* value for fan-out suggests that the overall complexity of X may be *high* because of the complexity of control logic needed to coordinate the called components.
- Generally, the *larger* the size of the code of a component, the more *complex* and error-prone the component is likely to be
- The *higher* the value of the Fog index, the more difficult the document is to understand
- I. Sommerville. *Software Engineering*, 8th ed. Addison-Wesley, Harlow, 2007.
-

consumption is *very important*, consider settings of a Proportional-Integral-Derivation (PID) controller producing *low overshoot*.' Some other examples of highly representative human-centric systems concern those involving (a) construction and usage of relevance feedback in retrieval, organization, and summarization of video and images; (b) queries formulated in natural languages; and (c) summarization of results coming as an outcome of some query.

Secondly, there are unique areas of applications in which fuzzy sets form a methodological backbone and deliver the required algorithmic setting. This concerns fuzzy modeling in which we start with collections of information granules (typically realized as fuzzy sets) and construct a model as a web of links (associations) between them. This approach is radically different from the numeric, function-based models encountered in 'standard' system modeling. Fuzzy modeling emphasizes an augmented agenda in comparison with the one stressed in numeric models. While we are still concerned with the accuracy of the resulting model, its interpretability and transparency become of equal, and sometimes even higher, relevance.

It is worth stressing that fuzzy sets provide an additional conceptual and algorithmic layer to the existing and well-established areas. For instance, there are profound contributions of fuzzy sets to pattern recognition. In this case, fuzzy sets build on the well-established technology of feature selection, classification, and clustering.

Fuzzy sets are an ultimate mechanism of communication between humans and computing environment. The essence of this interaction is illustrated in Figure 5.1. Any input is translated in terms of fuzzy sets and thus made comprehensible at the level of the computing system. Likewise, we see a similar role of fuzzy sets when communicating the results of detailed processing, retrieval, and alike. Depending on application and the established mode of interaction, the communication layer may involve a substantial deal of processing of fuzzy sets. Quite often we combine the mechanisms of communication and represent

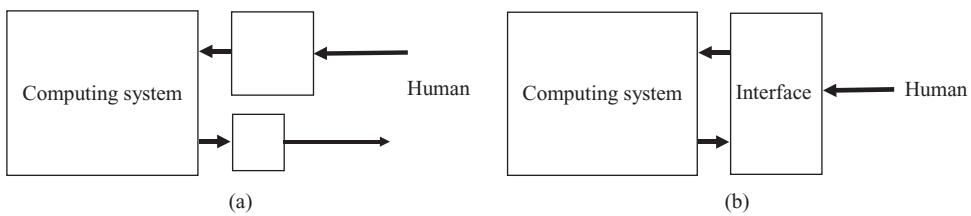


Figure 5.1 Fuzzy sets in the realization of communication mechanisms: (a) both at the user end and at the computing system side; (b) a unified representation of input and output mechanisms of communication in the form of the interface which could also embrace a certain machinery of processing at the level of fuzzy sets

them in the form of a single module (Figure 5.1b). This architectural representation stresses the human centricity aspect of the developed systems.

As an Efficient Computing Framework of Global Character

Rather than processing individual elements, say a single numeric datum, an encapsulation of a significant number of the individual elements that is realized in the form of some fuzzy sets offers immediate benefits of joint and orchestrated processing. Instead of looking at the individual number, we embrace a more general point of view and process an entire collection of elements represented now in the form of a single fuzzy set. This effect of a *collective* handling of individual elements is seen very profoundly in so-called fuzzy arithmetic. The basic constructs here are fuzzy numbers. In contrast to single numeric quantities (real numbers) fuzzy numbers represent collections of numbers where each of them belongs to the concept (fuzzy number) to some degree. These constructs are then subject to processing, say addition, subtraction, multiplication, division, etc. Noticeable is the fact that by processing fuzzy numbers we are in fact handling a significant number of individual elements at the same time. Fuzzy numbers and fuzzy arithmetic provide an interesting advantage over interval arithmetic (viz. arithmetic in which we are concerned with intervals – sets of numeric values). Intervals come with abrupt boundaries as elements can belong to or are excluded from the given set. This means, for example, that any gradient-based techniques of optimization invoked when computing solutions become very limited: the derivative is equal to zero, with an exception at the point where the abrupt boundary is located.

Fuzzy Sets as a Vehicle of Raising and Quantifying Awareness About Granularity of Outcomes

Fuzzy sets form the results of granular computing. As such they convey a global view at the elements of the universe of discourse over which they are constructed. When visualized, the values of the membership function describe a suitability of the individual points as compatible (preferred) with the solution. In this sense, fuzzy sets serve as a useful visualization vehicle: when displayed, the user could gain an overall view at the character of solution (regarded as a fuzzy set) and make a final choice. Note that this is very much in line with the idea of the human centricity: we present the user with all possible results; however, we do not put any pressure as to the commitment of selecting a certain numeric solution.

Fuzzy Sets as a Mechanism Realizing a Principle of the Least Commitment

As the computing realized in the setting of granular computing returns a fuzzy set as its result, it could be effectively used to realize a principle of the least commitment. The crux of this principle is to use fuzzy set as a mechanism of making us cognizant of the quality of obtained result. Consider a fuzzy set being a result of computing in some problem of multiphase decision making. The fuzzy set is defined over various alternatives and associates with them the corresponding degrees of preference (see Figure 5.2). If there are several alternatives with very similar degrees of membership, this serves as a clear indicator of uncertainty or hesitation as to the making of a decision. In other words, in light of the form of the generated fuzzy set, we do not intend to commit ourselves to making any decision (selection of

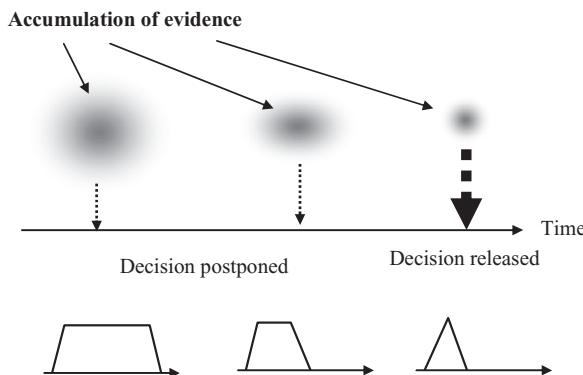


Figure 5.2 An essence of the principle of the least commitment; the decision is postponed until the phase where there is enough evidence accumulated and the granularity of the result becomes specific enough. Shown are also examples of fuzzy sets formed at successive phases of processing that become more specific along with the increased level of evidence

one of the alternatives) at this time. Our intent would be to postpone decision and collect more evidence. For instance, this could involve further collecting of data, soliciting expert opinion, and alike. With this evidence, we could continue with computing and evaluate the form of the resulting fuzzy set. It could well be that the collected evidence has resulted in more specific fuzzy set of decisions on the basis of which we could either still postpone decision and keep collecting more evidence or proceed with decision making. Thus the principle of the least commitment offers us an interesting and useful guideline as to the mechanism of decision making versus evidence collection.

5.3 Some Selected Descriptors of Fuzzy Sets

In principle, any function $A: X \rightarrow [0, 1]$ becomes potentially eligible to represent the membership function of fuzzy set A . Let us recall that any fuzzy set defined in X is represented by its membership function mapping the elements of the universe of discourse to the unit interval. The degree of membership, $A(x)$, quantifies an extent to which ' x ' is assigned to A . Higher values of $A(x)$ indicate stronger association of ' x ' with the concept conveyed by A .

In practice, however, the type and shape of membership functions should fully reflect the nature of the underlying phenomenon we are interested to model. We require that fuzzy sets should be semantically sound, which implies that the selection of membership functions needs to be guided by the character of the application and the nature of the problem we intend to solve.

Given the enormous diversity of potentially useful (viz. semantically sound) membership functions, there are certain common characteristics (descriptors) that are conceptually and operationally qualified to capture the essence of the granular constructs represented in terms of fuzzy sets. In what follows, we provide a list of the descriptors commonly encountered in practice [17–19].

Normality

We say that the fuzzy set A is *normal* if its membership function attains 1; that is,

$$\sup_{x \in X} A(x) = 1. \quad (1)$$

If this property does not hold, we call the fuzzy set *subnormal*. An illustration of the corresponding fuzzy set is shown in Figure 5.3. The supremum (\sup) in the above expression is also referred to as the height of the fuzzy set A , $\text{hgt}(A) = \sup_{x \in X} A(x) = 1$.

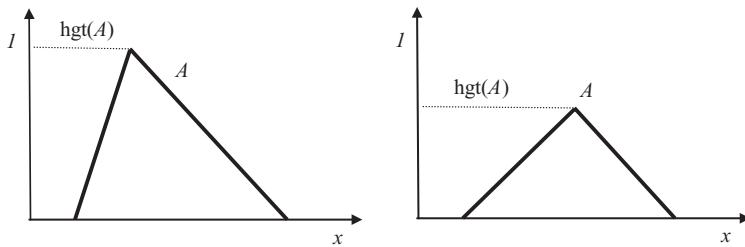


Figure 5.3 Examples of normal and subnormal fuzzy sets

The normality of A has a simple interpretation: by determining the height of the fuzzy set, we identify an element with the highest membership degree. The value of the height being equal to 1 states that there is at least one element in \mathbf{X} whose typicality with respect to A is the highest one and which could be sought as fully compatible with the semantic category presented by A . A subnormal fuzzy set whose height is lower than 1, viz. $\text{hgt}(A) < 1$, means that the degree of typicality of elements in this fuzzy set is somewhat lower (weaker) and we cannot identify any element in \mathbf{X} which is fully compatible with the underlying concept. Generally, while forming a fuzzy set we expect its normality. (Otherwise why would such a fuzzy set for which there are no typical elements come into existence in the first place?)

Normalization

The normalization operation, $\text{Norm}(\cdot)$, is a transformation mechanism that is used to convert a subnormal non-empty fuzzy set A into its normal counterpart. This is done by dividing the original membership function by the height of this fuzzy set; that is,

$$\text{Norm}(A) = \frac{A(x)}{\text{hgt}(A)}. \quad (2)$$

While the height describes the global property of the membership grades, the following notions offer an interesting characterization of the elements of \mathbf{X} vis-à-vis their membership degrees.

Support

Support of a fuzzy set A , denoted by $\text{Supp}(A)$, is a set of all elements of \mathbf{X} with non-zero membership degrees in A :

$$\text{Supp}(A) = \{x \in \mathbf{X} | A(x) > 0\}. \quad (3)$$

In other words, support identifies all elements of \mathbf{X} that exhibit some association with the fuzzy set under consideration (by being allocated to A with non-zero membership degrees).

Core

The core of a fuzzy set A , $\text{Core}(A)$, is a set of all elements of the universe that are typical to A , viz., they come with membership grades equal to 1;

$$\text{Core}(A) = \{x \in \mathbf{X} | A(x) = 1\}. \quad (4)$$

The support and core are related in the sense that they identify and collect elements belonging to the fuzzy set, yet at two different levels of membership. Given the character of the core and support, we note that all elements of the core of A are subsumed by the elements of the support of this fuzzy set. Note that both support and core are sets, not fuzzy sets (Figure 5.4.). We refer to them as the set-based characterizations of fuzzy sets.

While core and support are somewhat extreme (in the sense that they identify the elements of A that exhibit the strongest and the weakest linkages with A), we may also be interested in characterizing sets

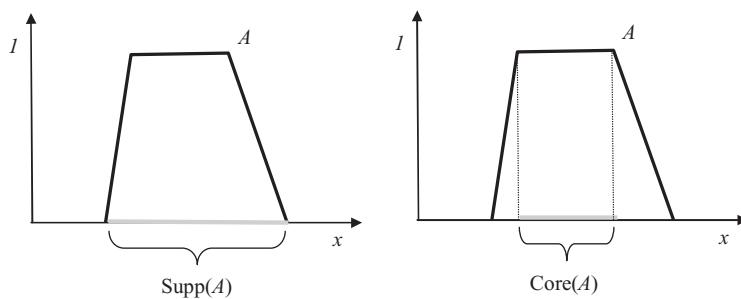


Figure 5.4 Support and core of A

of elements that come with some intermediate membership degrees. A notion of a so-called α -cut offers here an interesting insight into the nature of fuzzy sets.

α -Cut

The α -cut of a fuzzy set A , denoted by A_α , is a set consisting of the elements of the universe whose membership values are equal to or exceed a certain threshold level α , where $\alpha \in [0, 1]$. Formally speaking, we have $A_\alpha = \{x \in \mathbf{X} | A(x) \geq \alpha\}$. A strong α -cut differs from the α -cut in the sense that it identifies all elements in \mathbf{X} for which we have the following equality: $A_\alpha = \{x \in \mathbf{X} | A(x) > \alpha\}$. An illustration of the concept of the α -cut and strong α -cut is presented in Figure 5.5. Both support and core are limited cases of α -cuts and strong α -cuts. For $\alpha = 0$ and the strong α -cut, we arrive at the concept of the support of A . The threshold $\alpha = 1$ means that the corresponding α -cut is the core of A .

Convexity

We say that a fuzzy set is convex if its membership function satisfies the following condition:

For all $x_1, x_2 \in \mathbf{X}$ and all $\lambda \in [0, 1]$,

$$A[\lambda x_1 + (1 - \lambda)x_2] \geq \min[A(x_1), A(x_2)]. \quad (5)$$

The above relationship states that whenever we choose a point x on a line segment between x_1 and x_2 , the point $(x, A(x))$ is always located above or on the line passing through the two points $(x_1, A(x_1))$ and $(x_2, A(x_2))$ (refer to Figure 5.6).

Let us recall that a set S is convex if for all $x_1, x_2 \in S$, then $x = \lambda x_1 + (1 - \lambda)x_2 \in S$ for all $\lambda \in [0, 1]$. In other words, convexity means that any line segment identified by any two points in S is also contained in S . For instance, intervals of real numbers are convex sets. Therefore, if a fuzzy set is convex, then all of its α -cuts are convex, and conversely, if a fuzzy set has all its α -cuts convex, then it is a convex fuzzy set (refer to Figure 5.7). Thus we may say that a fuzzy set is convex if all its α -cuts are convex (intervals).

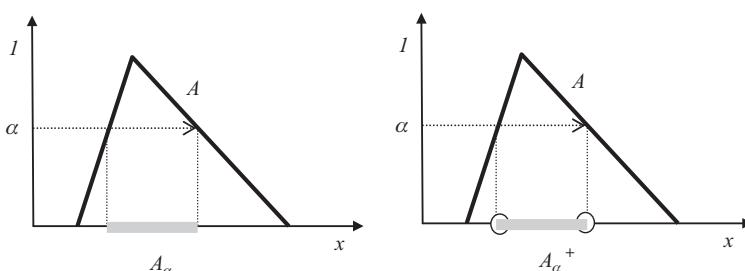


Figure 5.5 Examples of α -cut and strong α -cut

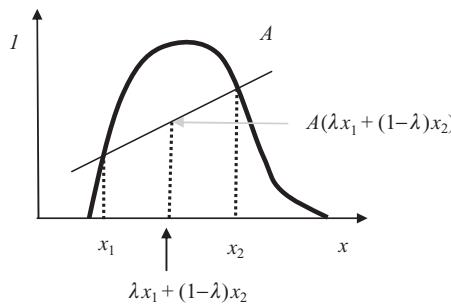


Figure 5.6 An example of a convex fuzzy set A

Fuzzy sets can be characterized by counting their elements and bringing a single numeric quantity as a meaningful descriptor of this count. While in case of sets, this sounds convincing; here we have to take into account different membership grades. In the simplest form this counting comes under the name of cardinality.

Cardinality

Given a fuzzy set A defined in a finite or countable universe \mathbf{X} , its cardinality, denoted by $\text{card}(A)$, is expressed as the following sum:

$$\text{card}(A) = \sum_{x \in X} A(x), \quad (6)$$

or, alternatively, as the following integral:

$$\text{card}(A) = \int_X A(x) \, dx. \quad (7)$$

(We assume that the integral shown above does make sense.) The cardinality produces a count of the number of elements in the given fuzzy set. As there are different degrees of membership, the use of the sum here makes sense as we keep adding contributions coming from the individual elements of this fuzzy set. Note that in the case of sets, we count the number of elements belonging to the corresponding sets. We also use the alternative notation of $\text{Card}(A) = |A|$ and refer to it as a sigma count (σ -count).

The cardinality of fuzzy sets is explicitly associated with the concept of granularity of information granules realized in this manner. More descriptively, the more the elements of A we encounter, the higher the level of abstraction supported by A and the lower the granularity of the construct. Higher values of

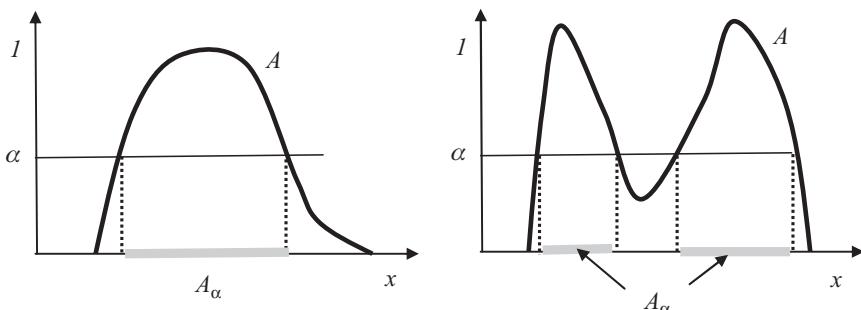


Figure 5.7 Examples of convex and non-convex fuzzy sets

cardinality come with the higher level of abstraction (generalization) and the lower values of granularity (specificity).

Equality and Inclusion Relationships in Fuzzy Sets

We investigate two essential relationships between two fuzzy sets defined in the same space that offer a useful insight into their fundamental dependencies. When defining these notions, bear in mind that they build on the well-known definitions encountered in set theory.

Equality

We say that two fuzzy sets A and B defined in the same universe \mathbf{X} are equal if and only if their membership functions are identical, meaning that

$$A(x) = B(x) \quad \forall x \in \mathbf{X}. \quad (8)$$

Inclusion

Fuzzy set A is a subset of B (A is included in B), denoted by $A \subseteq B$, if and only if every element of A also is an element of B . This property expressed in terms of membership degrees means that the following inequality is satisfied:

$$A(x) \leq B(x) \quad \forall x \in \mathbf{X}. \quad (9)$$

Interestingly, the definitions of equality and inclusion exhibit an obvious dichotomy as the property of equality (or inclusion) is satisfied or is not satisfied. While this quantification could be acceptable in the case of sets, fuzzy sets require more attention with this regard given that the membership degrees are involved in expressing the corresponding definitions.

The approach being envisioned here takes into consideration the degrees of membership and sets up a conjecture that any comparison of membership values should rather return a degree of equality or inclusion. For a given element of \mathbf{X} , let us introduce the following degree of inclusion of $A(x)$ in $B(x)$ and denote it by $A(x) \Rightarrow B(x)$: (\Rightarrow is the symbol of implication; the operation of implication itself will be discussed in detail later on; we do not need these details for the time being.)

$$A(x) \Rightarrow B(x) = \begin{cases} 1 & \text{if } A(x) \leq B(x) \\ 1 - A(x) + B(x) & \text{otherwise.} \end{cases} \quad (10)$$

If $A(x)$ and $B(x)$ are confined to 0 and 1 as in the case of sets, we come up with the standard definition of Boolean inclusion being used in set theory. Computing (10) for all elements of \mathbf{X} , we introduce a degree of inclusion of A in B , denoted by $\|A \subset B\|$, to be in the form

$$\|A \subset B\| = \frac{1}{\text{Card}(\mathbf{X})} \int_{\mathbf{X}} (A(x) \Rightarrow B(x)) \, dx. \quad (11)$$

We characterize the equality of A and B , $\|A = B\|$, using the following expression:

$$\|A = B\| = \frac{1}{\text{Card}(\mathbf{X})} \int_{\mathbf{X}} [\min((A(x) \Rightarrow B(x)), (B(x) \Rightarrow A(x))] \, dx. \quad (12)$$

Again this definition is appealing as it results as a direct consequence of the inclusion relationships that have to be satisfied with respect to the inclusion of A in B and B in A .

Examples. Let us consider two fuzzy sets A and B described by the Gaussian and triangular membership functions. Recall that Gaussian membership function is described as $(\exp(-(x - m)^2/\sigma^2))$, where the modal value and spread are denoted by ' m ' and ' σ ', respectively. The triangular fuzzy set is fully characterized by the spreads (a and b) and the modal value is equal to ' n '. Figure 5.8 provides some examples of A and B for selected values of the parameters and the resulting degrees of inclusion. They are intuitively appealing, reflecting the nature of relationship. (A is included in B .)

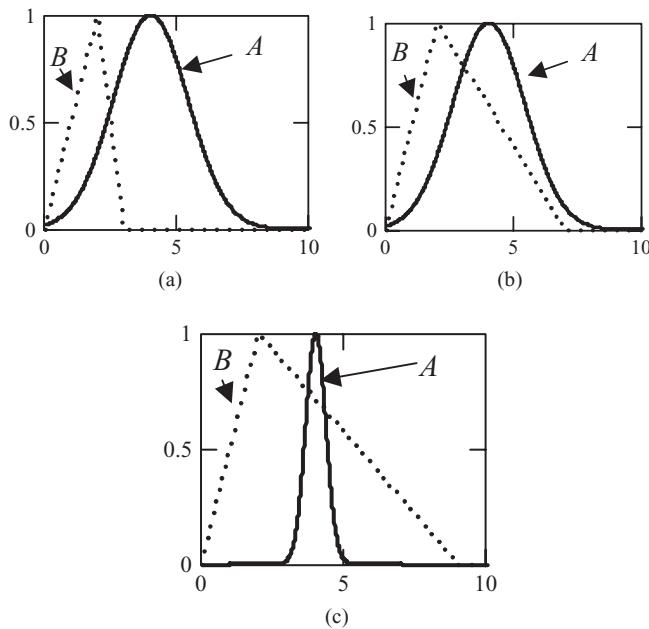


Figure 5.8 Examples of fuzzy sets A and B along with their degrees of inclusion: (a) $a = 0, n = 2, b = 3, m = 4, s = 2, \|A = B\| = 0.637$; (b) $b = 7, \|A = B\| = 0.864$; (c) $a = 0, n = 2, b = 9, m = 4, s = 0.5, \|A = B\| = 0.987$

Energy and Entropy Measures of Fuzziness

We can offer a global view at the collection of membership grades conveyed by fuzzy sets by aggregating them in the form of so-called measures of fuzziness. Two main categories of such measures are known in the form of energy and entropy measures of fuzziness [20, 21].

Energy measure of fuzziness, of a fuzzy set A in \mathbf{X} , denoted by $E(A)$, is a functional of the membership degrees:

$$E(A) = \sum_{i=1}^n e[A(x_i)] \quad (13)$$

if $\text{Card}(\mathbf{X}) = n$. In the case of the infinite space, the energy measure of fuzziness is the following integral:

$$E(A) = \int_{\mathbf{X}} e(A(x)) \, dx. \quad (14)$$

The mapping $e: [0, 1] \rightarrow [0, 1]$ is a functional monotonically increasing over $[0, 1]$ with the boundary conditions $e(0) = 0$ and $e(1) = 1$.

As the name of this measure stipulates, its role is to quantify a sort of energy associated with the given fuzzy set. The higher the membership degrees, the more essential are their contributions to the overall energy measure. In other words, by computing the energy measure of fuzziness we can compare fuzzy sets in terms of their overall count of membership degrees.

A particular form of the above functional comes with the identity mapping that is $e(u) = u$ for all u in $[0, 1]$. We can see that in this case, (13) and (14) reduce to the cardinality of A ,

$$E(A) = \sum_{i=1}^n A(x_i) = \text{Card}(A). \quad (15)$$

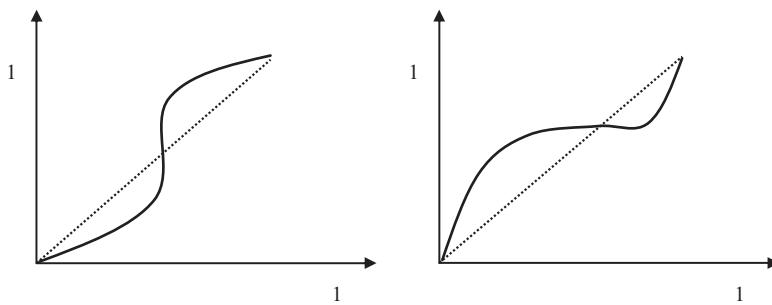


Figure 5.9 Two selected forms of the functional ‘ e ’: in (a) high values of membership are emphasized (accentuated), while in (b) the form of ‘ e ’ shown puts emphasis on lower membership grades

The energy measure of fuzziness forms a convenient way of expressing a total mass of the fuzzy set. Since $\text{card}(\emptyset) = 0$ and $\text{card}(\mathbf{X}) = n$, the more a fuzzy set differs from the empty set, the larger is its mass. Indeed, rewriting (15) we obtain

$$E(A) = \sum_{i=1}^n A(x_i) = \sum_{i=1}^n |A(x_i) - \emptyset(x_i)| = d(A, \emptyset) = \text{Card}(A), \quad (16)$$

where $d(A, \emptyset)$ is the Hamming distance between fuzzy set A and the empty set.

While the identity mapping (e) is the simplest alternative one could think of, in general, we can envision an infinite number of possible options. For instance, one could consider the functionals such as $e(u) = u^p$, $p > 0$, and $e(u) = \sin(\frac{\pi}{2}u)$. Note that by choosing a certain form of the functional, we accentuate a varying contribution of different membership grades. For instance, depending on the form of ‘ e ’, the contribution of the membership grades close to 1 could be emphasized, while those located close to 0 could be very much reduced. Figure 5.9 illustrates this effect by showing two different forms of the functional (e).

When each element x_i of \mathbf{X} appears with some probability p_i , the energy measure of fuzziness of the fuzzy set A can include this probabilistic information in which case it assumes the following format:

$$E(A) = \sum_{i=1}^n p_i e[A(x_i)]. \quad (17)$$

A careful inspection of the above expression reveals that $E(A)$ is the expected value of the functional $e(A)$. For infinite \mathbf{X} , we use an integral format of the energy measure of fuzziness:

$$E(A) = \int_{\mathbf{X}} p(x) e[A(x)] dx, \quad (18)$$

where $p(x)$ is the probability density function (pdf) defined over \mathbf{X} . Again, $E(A)$ is the expected value of $e(A)$.

Entropy Measure of Fuzziness

The entropy measure of fuzziness of A , denoted by $H(A)$, is built on the entropy functional (h) and comes in the form

$$H(A) = \sum_{i=1}^n h[A(x_i)] \quad (19)$$

or in the continuous case of \mathbf{X}

$$H(A) = \int_{\mathbf{X}} h(A(x)) \, dx, \quad (20)$$

where $h: [0, 1] \times [0, 1]$ is a functional such that (a) it is monotonically increasing in $[0, \frac{1}{2}]$ and monotonically decreasing in $[\frac{1}{2}, 1]$ and (b) comes with the boundary conditions $h(0) = h(1) = 0$ and $h(\frac{1}{2}) = 1$. This functional emphasizes membership degrees around $\frac{1}{2}$; in particular, the value of $\frac{1}{2}$ is stressed to be the most ‘unclear’ (causing the highest level of hesitation with its quantification by means of the proposed functional).

5.4 Granulation of Information

The notion of granulation emerges as a direct and immediate need to abstract and summarize information and data to support various processes of comprehension and decision making. For instance, we often sample an environment for values of attributes of state variables, but we rarely process all details because of our physical and cognitive limitations. Quite often, just a reduced number of variables, attributes, and values are considered because those are the only features of interest given the task under consideration. To avoid all necessary and highly distractive details, we require an effective abstraction procedure. As discussed earlier, detailed numeric information is aggregated into a format of information granules where the granules themselves are regarded as collections of elements that are perceived as being indistinguishable, similar, close, or functionally equivalent. Fuzzy sets are examples of information granules. When talking about a family of fuzzy sets, we are typically concerned with fuzzy partitions of \mathbf{X} .

More generally, the mechanism of granulation can be formally characterized by a four-tuple of the form [22–24]

$$< \mathbf{X}, G, S, C > \quad (21)$$

where \mathbf{X} is a universe of discourse (space), G a formal framework of granulation (resulting from the use of fuzzy sets, rough sets, etc.), S a collection of information granules, and C a transformation mechanism that realizes communication among granules of different nature and granularity levels [25, 26], (see Figure 5.10.) In Figure 5.10, notice the communication links that allow for communication between information granules expressed in the same formal framework but at different levels of granularity as well as communication links between information granules formed in different formal frameworks.

For instance, in the case of fuzzy granulation shown in Figure 5.10, if G is the formal framework of fuzzy sets, $S = \{F_1, F_2, F_3, F_4\}$, and C is a certain communication mechanism, then communicating the results of processing at the level of fuzzy sets to the framework of interval calculus, one could consider the use of some α -cuts. The pertinent computational details will be discussed later on.

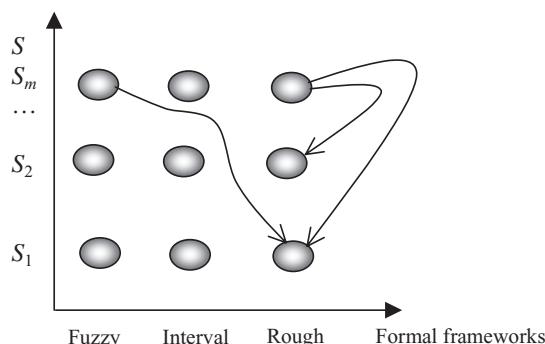


Figure 5.10 Granular computing and communication mechanisms in the coordinates of formal frameworks (fuzzy sets, intervals, rough sets, etc.) and levels of granularity

5.5 Characterization of the Families of Fuzzy Sets

As we have already mentioned, when dealing with information granulation we often develop a family of fuzzy sets and move on with the processing that inherently uses all the elements of this families. Alluding to the existing terminology, we will be referring such collections of information granules as frames of cognition. In what follows, we introduce the underlying concept and discuss its main properties.

Frame of Cognition

A frame of cognition is a result of information granulation in which we encounter a finite collection of fuzzy sets – information granules that ‘represent’ the entire universe of discourse and satisfy a system of semantic constraints. The frame of cognition is a notion of particular interest in fuzzy modeling, fuzzy control, classification, data analysis, to name a few, of representative examples. In essence, the frame of cognition is crucial to all applications where local and globally meaningful granulation is required to capture the semantics of the conceptual and algorithmic settings in which problem solving has to be placed.

A frame of cognition consists of several labeled, normal fuzzy sets. Each of these fuzzy sets is treated as a reference for further processing. A frame of cognition can be viewed as a codebook of conceptual entities. Being more descriptive, we may view them as a family of linguistic landmarks, say *small*, *medium*, *high*, etc. More formally, a frame of cognition Φ ,

$$\Phi = \{A_1, A_2, \dots, A_m\}, \quad (22)$$

is a collection of fuzzy sets defined in the same universe \mathbf{X} that satisfies at least two requirements of coverage and semantic soundness.

Coverage

We say that Φ covers \mathbf{X} if any element $x \in \mathbf{X}$ is compatible with at least one fuzzy set A_i in Φ , $i \in I = \{1, 2, \dots, m\}$, meaning that it is compatible (coincides) with A_i to some non-zero degree; that is,

$$\forall_{x \in \mathbf{X}} \exists_{i \in I} A_i(x) > 0. \quad (23)$$

Being more strict, we may require a satisfaction of the so-called δ -level coverage, which means that for any element of \mathbf{X} , fuzzy sets are activated to a degree not lower than δ :

$$\forall_{x \in \mathbf{X}} \exists_{i \in I} A_i(x) > 0 \quad \forall_{x \in \mathbf{X}} \exists_{i \in I} A_i(x) > \delta, \quad (24)$$

where $\delta \in [0, 1]$. Put it in a computational perspective, the coverage assures that each element of \mathbf{X} is represented by at least one of the elements of Φ and guarantees any absence of gaps viz. elements of \mathbf{X} for which there is no fuzzy set being compatible with it.

Semantic Soundness

The concept of semantic soundness is more complicated and difficult to quantify. In principle, we are interested in information granules of Φ that are meaningful. While there is far more flexibility in a way in which a suite of detailed requirements could be structured, we may agree on a collection several fundamental properties:

1. Each A_i , $i \in I$, is a unimodal and normal fuzzy set.
2. Fuzzy sets A_i , $i \in I$, are made disjoint enough to assure that they are sufficiently distinct to become linguistically meaningful. This imposes a maximum degree λ of overlap among any two elements of Φ . In other words, given any $x \in \mathbf{X}$, there is no more than one fuzzy set A_i such that $A_i(x) \geq \lambda$, $\lambda \in [0, 1]$.
3. The number of elements of Φ is kept low; following the psychological findings reported by Miller [27] and others we consider the number of fuzzy sets forming the frame of cognition to be maintained in the range of 7 ± 2 items.

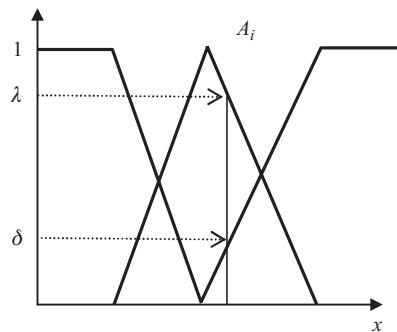


Figure 5.11 Coverage and semantic soundness of a cognitive frame

Coverage and semantic soundness [28] are the two essential conditions that should be fulfilled by the membership functions of A_i to achieve interpretability. In particular, δ -coverage and λ -overlapping induce a minimal (δ) and maximal (λ) level of overlap between fuzzy sets (Figure 5.11).

Main Characteristics of the Frames of Cognition

Considering the families of linguistic labels and associated fuzzy sets embraced in a frame of cognition, several characteristics are worth emphasizing.

Specificity

We say that the frame of cognition Φ_1 is more specific than Φ_2 if all the elements of Φ_1 are more specific than the elements of Φ_2 , (for some illustration refer to Figure 5.12.) The less specific cognition frames promote granulation realized at the higher level of abstraction (generalization). Subsequently, we are provided with the description that captures less details. The notion of specificity could be articulated as proposed in [29].

Granularity

Granularity of a frame of cognition relates to the granularity of fuzzy sets used there. The higher the number of fuzzy sets in the frame, the finer the resulting granulation. Therefore, the frame of cognition Φ_1 is finer than Φ_2 if $|\Phi_1| > |\Phi_2|$. If the converse holds, Φ_1 is coarser than Φ_2 (Figure 5.12).

Focus of Attention

A focus of attention (scope of perception) induced by a certain fuzzy set $A = A_i$ in Φ is defined as a certain α -cut of this fuzzy set. By moving A along \mathbf{X} while keeping its membership function unchanged, we can focus attention on a certain selected region of \mathbf{X} (as portrayed in Figure 5.13).

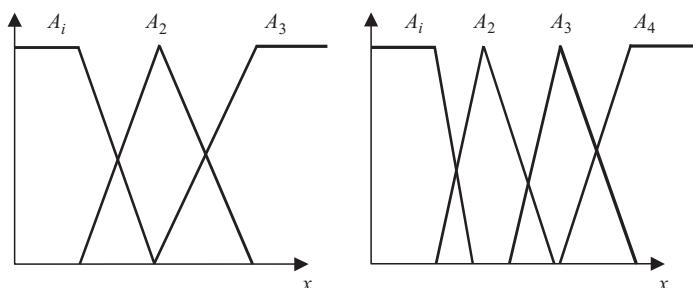


Figure 5.12 Examples of two frames of cognition; Φ_1 is coarser (more general) than Φ_2

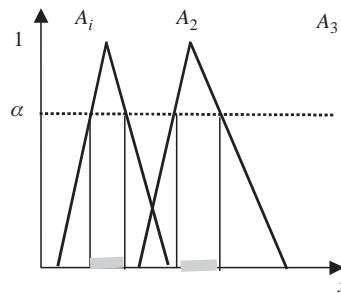


Figure 5.13 Focus of attention; shown are two regions of focus of attention implied by the corresponding fuzzy sets

Information Hiding

The idea of information hiding is closely related to the notion of focus of attention, and it manifests through a collection of elements that are hidden when viewed from the standpoint of membership functions. By modifying the membership function of $A = A_i$ in Φ we can produce an equivalence of the elements positioned within some region of \mathbf{X} . For instance, consider a trapezoidal fuzzy set A on \mathbf{R} and its 1-cut (viz., core), the closed interval $[a_2, a_3]$, as depicted in Figure 5.14.

All elements within the interval $[a_2, a_3]$ are made *indistinguishable*. Through the use of this specific fuzzy set they are made equivalent – in other words, when expressed in terms of A . Hence, more detailed information, viz., a position of a certain point falling within this interval, is ‘hidden.’ In general, by increasing or decreasing the level of the α -cut we can accomplish a so-called α -information hiding through normalization. For instance, as shown in Figure 5.15, the triangular fuzzy set subjected to its α -cut leads to the hiding of information about elements of \mathbf{X} falling within this α -cut.

5.6 Semantics of Fuzzy Sets: Some General Observations and Membership Estimation Techniques

There has been a great deal of methods aimed at the determination of membership functions (cf. [30–41]). Fuzzy sets are constructs that come with a well-defined meaning. They capture the semantics of the framework they intend to operate within. Fuzzy sets are the building conceptual blocks (generic constructs) that are used in problem description, modeling, control, and pattern classification tasks. Before discussing specific techniques of membership function estimation, it is worth casting the overall presentation in a certain context by emphasizing the aspect of the use of a finite number of fuzzy sets

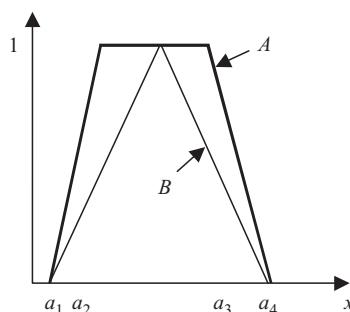


Figure 5.14 A concept of information hiding realized by the use of trapezoidal fuzzy set A : all elements in $[a_2, a_3]$ are made indistinguishable. The effect of information hiding is not present in case of triangular fuzzy set B

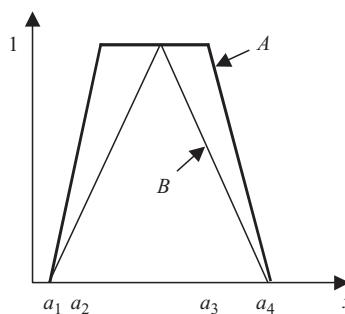


Figure 5.15 Triangular fuzzy set, its successive α -cuts, and the resulting effect of α -information hiding

leading to some essential vocabulary reflective of the underlying domain knowledge. In particular, we are concerned with the related semantics, calibration capabilities of membership functions, and the locality of fuzzy sets.

The limited capacity of a short-term memory, as identified by Miller, suggests that we could easily and comfortably handle and process 7 ± 2 items. This implies that the number of fuzzy sets to be considered as meaningful conceptual entities should be kept at the same level. The observation sounds reasonable – quite commonly, in practice, we witness situations in which this holds. For instance, when describing linguistically quantified variables, say error or change of error, we may use seven generic concepts (descriptors) labeling them as positive *large*, positive *medium*, positive *small*, *around zero*, negative *small*, negative *medium*, and negative *large*. When characterizing speed, we may talk about its quite intuitive descriptors such as *low*, *medium*, and *high* speed. In the description of an approximation error, we may typically use the concept of a *small* error around a point of linearization. (In all these examples, the terms are indicated in italics to emphasize the granular character of the constructs and the role being played there by fuzzy sets.) While embracing very different tasks, all these descriptors exhibit a striking similarity. All of them are information granules, not numbers (whose descriptive power is very much limited). In modular software development when dealing with a collection of modules (procedures, functions, and alike), the list of their parameters is always limited to a few items, which is again a reflection of the limited capacity of the short-term memory. The excessively long parameter list is strongly discouraged due to the possible programming errors and rapidly increasing difficulties of an effective comprehension of the software structure and ensuing flow of control.

In general, the use of an excessive number of terms does not offer any advantage. To the contrary, it remarkably clutters our description of the phenomenon and hampers further effective usage of the concepts we intend to establish to capture the essence of the domain knowledge. With the increase in the number of fuzzy sets, their semantics also becomes negatively impacted. Fuzzy sets may be built into a hierarchy of terms (descriptors) but at each level of this hierarchy (when moving down toward higher specificity that is an increasing level of detail), the number of fuzzy sets is kept at a certain limited level.

While fuzzy sets capture the semantics of the concepts, they may require some calibration, depending on the specification of the problem at hand. This flexibility of fuzzy sets should not be treated as any shortcoming but rather viewed as a certain and fully exploited advantage. For instance, the term *low* temperature comes with a clear meaning, yet it requires a certain calibration depending on the environment and the context it was put into. The concept of *low* temperature is used in different climate zones and is of relevance in any communication between people, yet for each of the community the meaning of the term is different thereby requiring some calibration. This could be realized, e.g., by shifting the membership function along the universe of discourse of temperature, affecting the universe of discourse by some translation, dilation, and alike. As a communication means, linguistic terms are fully legitimate and as such they appear in different settings. They require some refinement so that their meaning is fully understood and shared by the community of the users.

When discussing the methods aimed at the determination of membership functions or membership grades, it is worthwhile to underline the existence of the two main categories of approaches being reflective

of the origin of the numeric values of membership. The first one is reflective of the domain knowledge and opinions of experts. In the second one, we consider experimental data whose global characteristics become reflected in the form and parameters of the membership functions. In the first group we can refer to the pairwise comparison (Saaty's approach) as one of the representative examples, while fuzzy clustering is usually presented as a typical example of the data-driven method of membership function estimation. In what follows, we elaborate on several representative methods that will help us appreciate the level and flexibility of fuzzy sets.

Fuzzy Set as a Descriptor of Feasible Solutions

The aim of the method is to relate membership function to the level of feasibility of individual elements of a family of solutions associated with the problem at hand. Let us consider a certain function $f(x)$ defined in Ω ; that is, $f: \Omega \rightarrow \mathbf{R}$, where $\Omega \subset \mathbf{R}$. Our intent is to determine its maximum, namely $x^{\text{opt}} = \arg \max_x f(x)$. On the basis of the values of $f(x)$, we can form a fuzzy set A describing a collection of feasible solutions that could be labeled as optimal. Being more specific, we use the fuzzy set to represent an extent (degree) to which some specific values of ' x ' could be sought as potential (optimal) solutions to the problem. Taking this into consideration, we relate the membership function of A with the corresponding value of $f(x)$ cast in the context of the boundary values assumed by ' f '. For instance, the membership function of A could be expressed in the following form:

$$A(x) = \frac{f(x) - f_{\min}}{f_{\max} - f_{\min}}. \quad (25)$$

The boundary conditions are straightforward: $f_{\min} = \min_x f(x)$ and $f_{\max} = \max_x f(x)$, where the minimum and the maximum are computed over Ω . For other values of ' x ' where f attains its maximal value, $A(x)$ is equal 1, and around this point, the membership values are reduced when ' x ' is likely to be a solution to the problem $f(x) < f_{\max}$. The form of the membership function depends on the character of the function under consideration.

Linearization, its quality, and description of quality falls under the same banner as the optimization problem. When linearizing a function around some given point, a quality of such linearization can be represented in a form of some fuzzy set. Its membership function attains 1 for all these points where the linearization error is equal to zero. (In particular, this holds at the point around which the linearization is carried out.)

Fuzzy Set as a Descriptor of the Notion of Typicality

Fuzzy sets address an issue of gradual *typicality* of elements to a given concept. They stress the fact that there are elements that fully satisfy the concept (are typical for it) and there are various elements that are allowed only with partial membership degrees. The form of the membership function is reflective of the semantics of the concept. Its details could be captured by adjusting the parameters of the membership function or choosing its form depending on experimental data. For instance, consider a fuzzy set of squares. Formally, a rectangle includes a square shape as its special example when the sides are equal, $a = b$ (Figure 5.16). What if $a = b + \varepsilon$, where ε is a very small positive number? Could this figure be sought as a square? It is very likely so. Perhaps the membership value of the corresponding membership

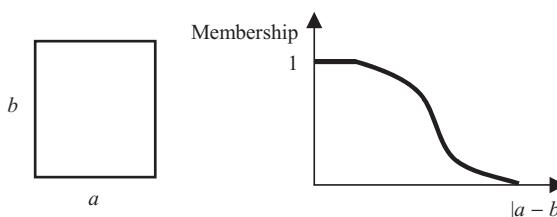


Figure 5.16 Perception of geometry of squares and its quantification in the form of membership function of the concept of fuzzy square

function could be equal to 0.99. Our perception, which comes with some level of tolerance to imprecision, does not allow us to tell apart this figure from the ideal square (Figure 5.16).

Non-Linear Transformation of Fuzzy Sets

In many problems, we encounter a family of fuzzy sets defined in the same space. The family of fuzzy sets $\{A_1, A_2, \dots, A_c\}$ is referred to as referential fuzzy sets. To form a family of semantically meaningful descriptors of the variable at hand, we usually require that these fuzzy sets satisfy the requirements of unimodality, limited overlap, and coverage. Technically, all these features are reflective of our intention to provide this family of fuzzy sets with some semantics. These fuzzy sets could be sought as generic descriptors (say, *small*, *medium*, *high*, etc.) being described by some typical membership functions. For instance, those could be uniformly distributed triangular or Gaussian fuzzy sets with some standard level of overlap between the successive terms (descriptors).

As mentioned, fuzzy sets are usually subject to some calibration depending on the character of the problem at hand. We may use the same terms *small*, *medium*, and *large* in various contexts, yet their detailed meaning (viz., membership degrees) has to be adjusted (adjusted). For the given family of the referential fuzzy sets, their calibration could be accomplished by taking the space $\mathbf{X} = [a, b]$ over which they are originally defined and transforming it into itself, that is, $[a, b]$ through some non-decreasing monotonic and continuous function $\Phi(x, \mathbf{p})$, where \mathbf{p} is a vector of some adjustable parameters bringing the required flexibility of the mapping. The non-linearity of the mapping is such that some regions of \mathbf{X} are contracted and some of them are stretched (expanded) and in this manner capture the required local context of the problem. This affects the membership functions of the referential fuzzy sets $\{A_1, A_2, \dots, A_c\}$ whose membership functions are now expressed as $A_i(\Phi(x))$. The construction of the mapping Φ is optimized, taking into account some experimental data concerning membership grades given at some points of \mathbf{X} . More specifically, the experimental data come in the form of the input–output pairs:

$$\begin{aligned} &x_1 - \mu_1(1), \mu_2(1), \dots, \mu_c(1) \\ &x_2 - \mu_1(2), \mu_2(2), \dots, \mu_c(2) \\ &\dots \\ &x_N - \mu_1(N), \mu_2(N), \dots, \mu_c(N), \end{aligned} \tag{26}$$

where the k th input–output pair consists of x_k , which denotes some point in \mathbf{X} , while $\mu_1(k), \mu_2(k), \dots, \mu_c(k)$ are the numeric values of the corresponding membership degrees.

The objective is to construct a non-linear mapping that is optimizing it with respect to the available parameters \mathbf{p} . More formally, we could translate the problem into the minimization of the following sum of squared errors:

$$\sum_{i=1}^c (A_i(\Phi(x_1; \mathbf{p}) - \mu_i(1))^2 + \sum_{i=1}^c (A_i(\Phi(x_2; \mathbf{p}) - \mu_i(2))^2 + \dots + \sum_{i=1}^c (A_i(\Phi(x_N; \mathbf{p}) - \mu_i(N))^2. \tag{27}$$

One of the feasible mapping comes in the form of a piecewise linear function shown in Figure 5.17. Here the vector of the adjustable parameters \mathbf{p} involves a collection of the split points r_1, r_2, \dots, r_L and the associated differences D_1, D_2, \dots, D_L ; hence, $\mathbf{p} = [r_1, r_2, \dots, r_L, D_1, D_2, \dots, D_L]$. The regions of expansion or compression are used to affect the referential membership functions and adjust their values given the experimental data.

Examples. We consider some examples of non-linear transformations of Gaussian fuzzy sets through the piecewise linear transformations (here $L = 3$) shown in Figure 5.18.

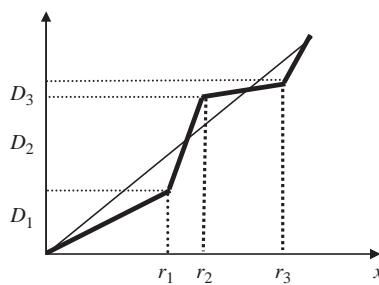


Figure 5.17 A Piecewise linear transformation function Φ ; shown also is a linear mapping not affecting the universe of discourse and not exhibiting any impact on the referential fuzzy sets. The proposed piecewise linear mapping is fully invertible

Note the fact that some fuzzy sets become more specific, while the others are made more general and expanded over some regions of the universe of discourse. This transformation leads to the membership functions illustrated in Figure 5.19.

Considering the same non-linear mapping as before, two triangular fuzzy sets are converted into fuzzy sets described by piecewise membership functions as shown in Figure 5.20.

Some other examples of the transformation of fuzzy sets through the piecewise mapping are shown in Figure 5.21.

Vertical and Horizontal Schemes of Membership Estimation

The vertical and horizontal modes of membership estimation are two standard approaches used in the determination of fuzzy sets. They reflect distinct ways of looking at fuzzy sets whose membership functions at some finite number of points are quantified by experts. In the horizontal approach we identify a collection of elements in the universe of discourse \mathbf{X} and request that an expert answers the question

Does x belong to concept A ?

The answers are expected to come in a binary (yes–no) format. The concept A defined in \mathbf{X} could be any linguistic notion, say *high speed*, *low temperature*, etc. Given ‘ n ’ experts whose answers for a given point of \mathbf{X} form a mix of yes–no replies, we count the number of ‘yes’ answers and compute the ratio of the positive answers (p) versus the total number of replies (n), i.e., is p/n . This ratio (likelihood) is treated as a membership degree of the concept at the given point of the universe of discourse. When all experts accept that the element belongs to the concept, then its membership degree is equal to 1. Higher disagreement between the experts (quite divided opinions) results in lower membership degrees. The concept A defined in \mathbf{X} requires collecting results for some other elements of \mathbf{X} and determining the corresponding ratios as outlined in Figure 5.22.

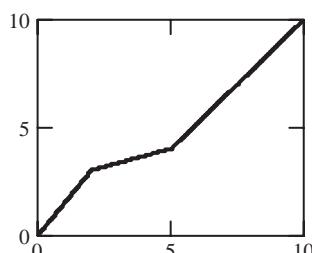


Figure 5.18 An example of the piecewise linear transformation

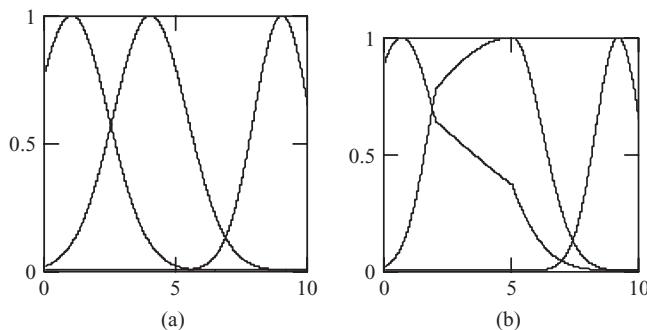


Figure 5.19 Examples of (a) original membership functions and (b) the resulting fuzzy sets after the piecewise linear transformation

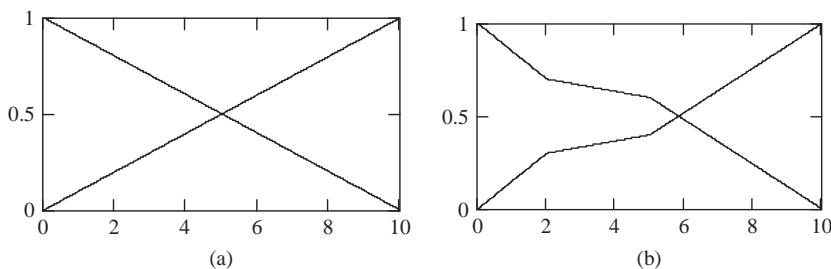


Figure 5.20 Two triangular fuzzy sets along with their piecewise linear transformation

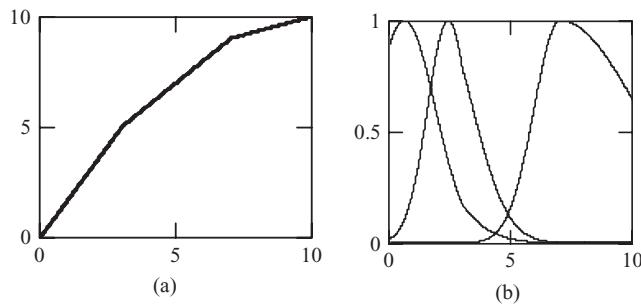


Figure 5.21 (a) The piecewise linear mapping and (b) the transformed Gaussian fuzzy sets

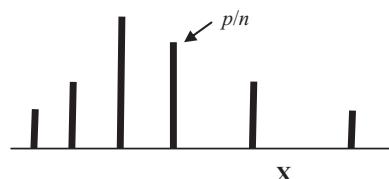


Figure 5.22 A horizontal method of the estimation of the membership function; observe a series of estimates determined for selected elements of \mathbf{X} . Note also that the elements of \mathbf{X} need not be evenly distributed

If replies follow some, e.g., binomial distribution, then we could determine a confidence interval of the individual membership grade. The standard deviation of the estimate of the positive answers associated with the point x , denoted here by σ , is given in the form

$$\sigma = \sqrt{\frac{p(1-p)}{n}}. \quad (28)$$

The associated confidence interval which describes a range of membership values is then determined as

$$[p - \sigma, p + \sigma]. \quad (29)$$

In essence, when the confidence intervals are taken into consideration, the membership estimates become intervals of possible membership values and this leads to the concept of so-called interval-valued fuzzy sets. By assessing the width of the estimates, we could control the execution of the experiment: when the ranges are too long, one could redesign the experiment and closely monitor the consistency of the responses collected in the experiment.

The advantage of the method comes with its simplicity as the technique explicitly relies on a direct counting of responses. The concept is also intuitively appealing. The probabilistic nature of the replies helps build confidence intervals that are essential to the assessment of the specificity of the membership quantification. A certain drawback is related to the local character of the construct: as the estimates of the membership function are completed separately for each element of the universe of discourse, they could exhibit a lack of continuity when moving from certain point to its neighbor. This concern is particularly valid in the case when X is a subset of real numbers.

The vertical mode of membership estimation is concerned with the estimation of the membership function by focusing on the determination of the successive α -cuts. The experiment focuses on the unit interval of membership grades. The experts involved in the experiment are asked the questions of the form

What are the elements of X which belong to fuzzy set A at degree not lower than α ?

Here α is a certain level (threshold) of membership grades in $[0, 1]$. The essence of the method is illustrated in Figure 5.23. Note that the satisfaction of the inclusion constraint is obvious: we envision that for higher values of α , the expert is going to provide more limited subsets of X ; the vertical approach leads to the fuzzy set by combining the estimates of the corresponding α -cuts. Given the nature of this method, we are referring to the collection of random sets as these estimates appear in the successive stages of the estimation process.

The elements are identified by the expert as they form the corresponding α -cuts of A . By repeating the process for several selected values of α we end up with the α -cuts and using them we reconstruct the fuzzy set. The simplicity of the method is its genuine advantage. Like in the horizontal method of membership estimation, a possible lack of continuity is a certain disadvantage one has to be aware of.

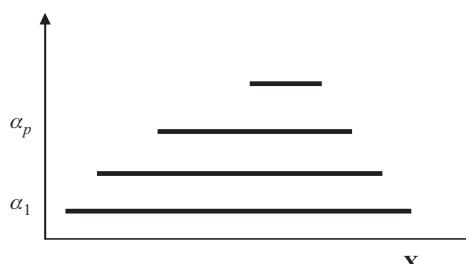


Figure 5.23 A vertical approach of membership estimation through the reconstruction of a fuzzy set through its estimated α -cuts

Here the selection of suitable levels of α needs to be carefully investigated. Similarly, an order at which different levels of α are used in the experiment could impact the estimate of the membership function.

Saaty's Priority Method of Pairwise Membership Function Estimation

The priority method introduced by Saaty [42, 43] forms another interesting alternative used to estimate the membership function. To explain the essence of the method, let us consider a collection of elements x_1, x_2, \dots, x_n (those could be, for instance, some alternatives whose allocation to a certain fuzzy set is sought) for which given are membership grades $A(x_1), A(x_2), \dots, A(x_n)$. Let us organize them into a so-called reciprocal matrix of the following form:

$$R = [r_{ij}] = \begin{bmatrix} \frac{A(x_1)}{A(x_1)} & \frac{A(x_1)}{A(x_2)} & \dots & \frac{A(x_1)}{A(x_n)} \\ \frac{A(x_2)}{A(x_1)} & \frac{A(x_2)}{A(x_2)} & \dots & \frac{A(x_2)}{A(x_n)} \\ \frac{A(x_3)}{A(x_1)} & \frac{A(x_3)}{A(x_2)} & \dots & \frac{A(x_3)}{A(x_n)} \\ \vdots & \vdots & & \vdots \\ \frac{A(x_n)}{A(x_1)} & \frac{A(x_n)}{A(x_2)} & \dots & \frac{A(x_n)}{A(x_n)} \end{bmatrix} = \begin{bmatrix} 1 & \frac{A(x_1)}{A(x_2)} & \dots & \frac{A(x_1)}{A(x_n)} \\ \frac{A(x_2)}{A(x_1)} & 1 & \dots & \frac{A(x_2)}{A(x_n)} \\ \vdots & \vdots & & \vdots \\ \frac{A(x_n)}{A(x_1)} & \frac{A(x_n)}{A(x_2)} & \dots & 1 \end{bmatrix}. \quad (30)$$

Noticeably, the diagonal values of R are equal to 1. The entries that are symmetrically positioned with respect to the diagonal satisfy the condition of reciprocity; that is, $r_{ij} = 1/r_{ji}$. Furthermore, an important transitivity property holds; that is, $r_{ik}r_{kj} = r_{ij}$ for all indexes i, j , and k . This property holds because of the way in which the matrix has been constructed. By plugging in the ratios one gets $r_{ik}r_{kn} = \frac{A(x_i)}{A(x_k)} \frac{A(x_k)}{A(x_j)} = \frac{A(x_i)}{A(x_j)} = r_{ij}$. Let us now multiply the matrix by the vector of the membership grades $A = [A(x_1) A(x_2) \dots A(x_n)]^T$. For the i th row of R (i.e., the i th entry of the resulting vector of results) we obtain

$$[RA]_i = \left[\frac{A(x_i)}{A(x_1)} \frac{A(x_i)}{A(x_2)} \dots \frac{A(x_i)}{A(x_n)} \right] \begin{bmatrix} A(x_1) \\ A(x_2) \\ \vdots \\ A(x_n) \end{bmatrix}, \quad (31)$$

$i = 1, 2, \dots, n$. Thus, the i th element of the vector is equal to $nA(x_i)$. Overall, once completed the calculations for all ' i ', this leads us to the expression $RA = nA$. In other words, we conclude that A is the eigenvector of R associated with the largest eigenvalue of R , which is equal to ' n '. In the above scenario, we assumed that the membership values $A(x_i)$ are given and then showed what form of results could they lead to. In practice, the membership grades are not given and have to be looked for.

The starting point of the estimation process are entries of the reciprocal matrix which are obtained through collecting results of pairwise evaluations offered by an expert, designer, or user (depending on the character of the task at hand). Prior to making any assessment, the expert is provided with a finite scale with values spread in between 1 and 7. Some other alternatives of the scales such as those involving five or nine levels could be sought as well. If x_i is strongly preferred over x_j when being considered in the context of the fuzzy set whose membership function we would like to estimate, then this judgment is expressed by assigning high values of the available scale, say 6 or 7. If we still sense that x_i is preferred over x_j , yet the strength of this preference is lower in comparison with the previous case, then this is quantified using some intermediate values of the scale, say 3 or 4. If no difference is sensed, the values close to 1 are the preferred choice, say 2 or 1. The value of 1 indicates that x_i and x_j are equally preferred. On the other hand, if x_j is preferred over x_i , the corresponding entry assumes values below 1. Given the reciprocal character of the assessment, once the preference of x_i over x_j has been quantified, the inverse of this number is plugged into the entry of the matrix that is located at the (j, i) th coordinate. As indicated earlier, the elements on the main diagonal are equal to 1. Next the maximal eigenvalue is computed along with its corresponding eigenvector. The normalized version of the eigenvector is then the membership function of the fuzzy set we considered when doing all pairwise assessments of the elements of its universe of discourse. The pairwise evaluations are far more convenient and manageable in comparison to any effort we make when assigning membership grades to all elements of the universe in a single step.

Practically, the pairwise comparison helps the expert focus only on two elements once at a time, thus reducing uncertainty and hesitation, while leading to the higher level of consistency. The assessments are not free of bias and could exhibit some inconsistent evaluations. In particular, we cannot expect that the transitivity requirement could be fully satisfied. Fortunately, the lack of consistency could be quantified and monitored. The largest eigenvalue computed for R is always greater than the dimensionality of the reciprocal matrix (recall that in reciprocal matrices the elements positioned symmetrically along the main diagonal are inverse of each other), $\lambda_{\max} > n$, where the equality $\lambda_{\max} = n$ occurs only if the results are fully consistent. The ratio

$$\nu = (\lambda_{\max} - n)/(n - 1) \quad (32)$$

can be regarded as an index of inconsistency of the data; the higher its value, the less consistent are the collected experimental results. This expression can be sought as the indicator of the quality of the pairwise assessments provided by the expert. If the value of ν is too high, exceeding a certain superimposed threshold, the experiment may need to be repeated. Typically if ν is less than 0.1, the assessment is sought to be consistent, while higher values of ν call for a reexamination of the experimental data and a rerun of the experiment. To quantify how much the experimental data deviate from the transitivity requirement, we calculate the absolute differences between the corresponding experimentally obtained entries of the reciprocal matrix, namely r_{ik} and $r_{ij}r_{jk}$. The sum expressed in the form

$$V(i, k) = \sum_{j=1}^n |r_{ij}r_{jk} - r_{ik}| \quad (33)$$

serves as a useful indicator of the lack of transitivity of the experimental data for the given pair of elements (i, k) . If required, we may repeat the experiment if the above sum takes high values. The overall sum $\sum_{i,k} V(i, k)$ then becomes a global evaluation of the lack of transitivity of the experimental assessment.

Fuzzy Sets as Granular Representatives of Numeric Data

In general, a fuzzy set is reflective of numeric data that are put together in some context. Using its membership function we attempt to embrace them in a concise manner. The development of the fuzzy set is supported by the following experiment-driven and intuitively appealing rationale:

- (a) First, we expect that A reflects (or matches) the available experimental data to the highest extent.
- (b) Second, the fuzzy set is kept specific enough so that it comes with a well-defined semantics.

These two requirements point at the multiobjective nature of the construct: we want to maximize the coverage of experimental data (as articulated by (a)) and minimize the spread of the fuzzy set (as captured by (b)). These two requirements give rise to a certain optimization problem. Furthermore, which is quite legitimate, we assume that the fuzzy set to be constructed has a unimodal membership function or its maximal membership grades occupy a contiguous region in the universe of discourse in which this fuzzy set has been defined. This helps us separately build a membership function for its rising and declining sections. The core of the fuzzy set is determined first. Next, assuming the simplest scenario when using the linear type of membership functions, the essence of the optimization problem boils down to the rotation of the linear section of the membership function around the upper point of the core of A (for the illustration refer to Figure 5.24.) The point of rotation of the linear segment of this membership function is marked by an empty circle. By rotating this segment, we intend to maximize (a) and minimize (b).

Before moving on with the determination of the membership function, we concentrate on the location of its numeric representative. Typically, one could view an average of the experimental data x_1, x_2, \dots, x_n to be its sound representative. While its usage is quite common in practice, a better representative of the numeric data is a median value. There is a reason behind this choice. The median is a robust statistic, meaning that it allows for a high level of tolerance to potential noise existing in the data. Its important ability is to ignore outliers. Given that the fuzzy set is sought to be a granular and ‘stable’ representation of the numeric data, our interest is in the robust development, not being affected by noise. Undoubtedly,

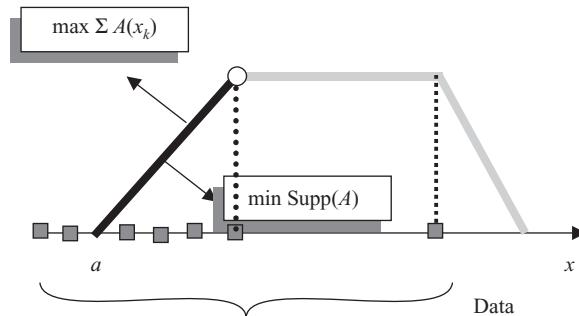


Figure 5.24 Optimization of the linear increasing section of the membership function of A ; highlighted are the positions of the membership function originating from the realization of the two conflicting criteria

the use of the median is a good starting point. Let us recall that the median is an order statistic and is formed on the basis of an ordered set of numeric values. In the case of the odd number of data in the data set, the point located in the middle of this ordered sequence is the median. When we encounter an even number of data in the granulation window, instead of picking up an average of the two points located in the middle, we consider these two points to form a core of the fuzzy set. Thus depending on the number of data points, we end up either with triangular or with trapezoidal membership function.

Having fixed the modal value of A (that could be a single numeric value, ' m ', or a certain interval $[m, n]$), the optimization of the spreads of the linear portions of the membership functions are carried out separately for their increasing and decreasing portions. We consider the increasing part of the membership function. (The decreasing part is handled in an analogous manner.) Referring to Figure 5.24, the two requirements guiding the design of the fuzzy set are and transformed into the corresponding multiobjective optimization problem as outlined as follows:

- (a) Maximize the experimental evidence of the fuzzy set; this implies that we tend to ‘cover’ as many numeric data as possible, viz., the coverage has to be made as high as possible. Graphically, in the optimization of this requirement, we rotate the linear segment up (clockwise), as illustrated in Figure 5.13. Formally, the sum of the membership grades $A(x_k)$, $\sum_k A(x_k)$, where A is the linear membership function to be optimized and x_k is located to the left to the modal value, has to be maximized.
- (b) Simultaneously, we would like to make the fuzzy set as specific as possible so that it comes with some well-defined semantics. This requirement is met by making the support of A as small as possible, i.e., $\min_a |m - a|$.

To accommodate the two conflicting requirements, we combine (a) and (b) in the form of the ratio that is maximized with respect to the unknown parameter of the linear section of the membership function

$$\max_a \frac{\sum_k A(x_k)}{|m - a|}. \quad (34)$$

The linearly decreasing portion of the membership function is optimized in the same manner. The overall optimization returns the parameters of the fuzzy number in the form of the lower and upper bound (a and b , respectively) and its support (m or $[m, n]$). We can write down such fuzzy numbers as $A(a, m, n, b)$. We exclude a trivial solution of $a = m$ in which case the fuzzy set collapses to a single numeric entity.

Fuzzy Sets as Aggregates of Numeric Data

Fuzzy sets can be formed on the basis of numeric data through their clustering (groupings). The groups of data give rise to membership functions that convey a global, more abstract view at the available data.

With this regard, fuzzy c -means (FCM, for brief) is one of the commonly used mechanisms of fuzzy clustering [16, 44].

Let us review its formulation, develop the algorithm, and highlight the main properties of the fuzzy clusters. Given a collection of n -dimensional data set $\{\mathbf{x}_k\}$, $k = 1, 2, \dots, N$, the task of determining its structure – a collection of ‘ c ’ clusters – is expressed as a minimization of the following objective function (performance index), Q being regarded as a sum of the squared distances,

$$Q = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m \|\mathbf{x}_k - \mathbf{v}_i\|^2, \quad (35)$$

where \mathbf{v}_i are n -dimensional prototypes of the clusters, $i = 1, 2, \dots, c$, and $U = [u_{ik}]$ stands for a partition matrix expressing a way of allocation of the data to the corresponding clusters; u_{ik} is the membership degree of data \mathbf{x}_k in the i th cluster. The distance between the data \mathbf{z}_k and prototype \mathbf{v}_i is denoted by $\|\cdot\|$. The fuzzification coefficient $m (> 1.0)$ expresses the impact of the membership grades on the individual clusters.

A partition matrix satisfies two important properties:

$$0 < \sum_{k=1}^N u_{ik} < N, \quad i = 1, 2, \dots, c. \quad (36a)$$

$$\sum_{i=1}^c u_{ik} = 1, \quad k = 1, 2, \dots, N. \quad (36b)$$

Let us denote by \mathbf{U} a family of matrices satisfying (36a) and (36b). The first requirement states that each cluster has to be non-empty and different from the entire set. The second requirement states that the sum of the membership grades should be confined to 1.

The minimization of Q completed with respect to $U \in \mathbf{U}$ and the prototypes \mathbf{v}_i of $V = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c\}$ of the clusters. More explicitly, we write it down as follows:

$$\min Q \text{ with respect to } U \in \mathbf{U}, \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c \in \mathbf{R}^n. \quad (37)$$

It is worth emphasizing that the FCM algorithm is a highly representative method of membership estimation that profoundly dwells on the use of experimental data. In contrast to some other techniques presented so far that are also data driven, FCM can easily cope with multivariable experimental data.

5.7 Granular Modeling with Fuzzy Sets

Fuzzy sets being information granules constitute building blocks of fuzzy models. There has been a substantial wealth of architectures of these models supporting the development of human-centric systems. In what follows, we briefly highlight the main directions by concentrating on the architectural blueprints and underlying functional modules by showing that in spite of the enormous diversity of the topologies of the models, there are some general and uniform underpinnings. We discuss validation and verification phases, which play a pivotal role in the context of fuzzy modeling.

5.7.1 The Architectural Blueprint of Fuzzy Models

In general, fuzzy models operate at a level of information granules – fuzzy sets – and in this way constitute highly abstract and flexible constructs [25, 45–48]. Given the environment of physical variables describing the surrounding world and an abstract view at the system under modeling, a very general view at the architecture of the fuzzy model can be portrayed as presented in Figure 5.25 (cf. [15, 16]).

We clearly distinguish between three functional components of the model where each of them comes with well-defined objectives. The input interface builds a collection of modalities (fuzzy sets and fuzzy relations) that are required to link the fuzzy model and its processing core with the external world. This processing core realizes all computing being carried out at the level of fuzzy sets (membership functions) already used in the interfaces. The output interface converts the results of granular (fuzzy) processing

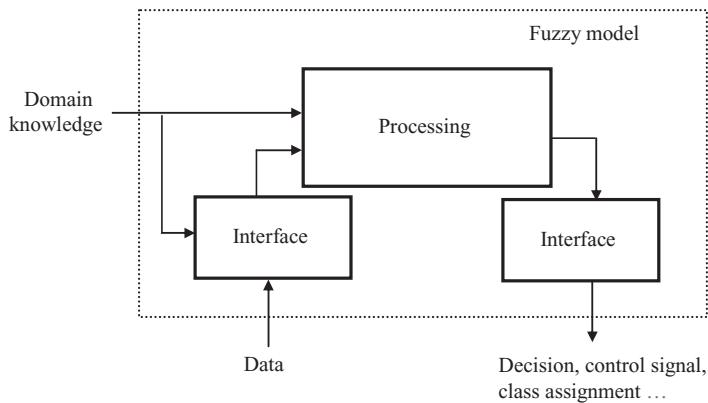


Figure 5.25 A general view at the underlying architecture of fuzzy models along with its three fundamental functional modules

into the format acceptable by the modeling environment. In particular, this transformation may involve numeric values being the representatives of the fuzzy sets produced by the processing core. The interfaces could be present in different categories of the models, yet they may show up to a significant extent. Their presence and relevance of the pertinent functionality depend on the architecture of the specific fuzzy model and a way in which the model is utilized. The interfaces are also essential when the models are developed on the basis of available numeric experimental evidence as well as some prior knowledge provided by designers and experts.

For instance, for rule-based topology of the model that is based on fuzzy sets in input and output variables we require well-developed interfaces. The generic models in this category are formulated as follows:

$$\text{if } X_1 \text{ is } A \text{ and } X_2 \text{ is } B \text{ and } \dots, \text{then } y \text{ is } C, \quad (38)$$

where X_1, X_2, \dots are input variables and Y is the output variable, while A, B, C, \dots are the fuzzy sets defined in the corresponding spaces (variables). Any logic processing carried out by the rule-based inference mechanism requires that any input is transformed, i.e., expressed in terms of fuzzy sets and the results of reasoning are offered in its numeric format (at which stage we require that the result is produced through the transformation of the fuzzy set of conclusion).

The rule-based models endowed with local models forming their consequents (conclusion parts), commonly referred to as fuzzy functional or Takagi–Sugeno fuzzy models [49], are governed by the formula

$$\text{if } x_1 \text{ is } A \text{ and } x_2 \text{ is } B \text{ and } \dots, \text{then } y \text{ is } f_i(\mathbf{x}, \mathbf{a}_i), \quad (39)$$

where $f_i(\mathbf{x}, \mathbf{a}_i)$ denotes a multivariable local model, where $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]$ and a vector of parameters $\mathbf{a}_i = [a_{i1} \ a_{i2} \ \dots \ a_{in}]$. In particular, one can envision a linear form of the model in which ' f_i ' becomes a linear function of its parameters, namely $f_i(\mathbf{x}, \mathbf{a}_i) = \mathbf{a}_i^T \mathbf{x}$. Obviously, depending on the specificity of the problem and a structure of available data, these regression models could be made non-linear. Depending on their character dictated by the problem at hand, we may be concerned with polynomial regression models (say, quadratic, cubic, trigonometric, etc.). The region of operation (viz., the area where the rule is relevant) of the rule is determined by the form and a location of the fuzzy sets located in the input space, which then occur in this particular rule (see Figure 5.26).

In this case the output interface is not required as the output of the model is numeric. Obviously, we still have to use a well-defined input interface as its components (fuzzy sets) form a condition part of the rules. Any input has to be transformed and communicated to the inference procedure making use of the fuzzy sets of the interface. Rule-based models are central architectures of fuzzy models. We will

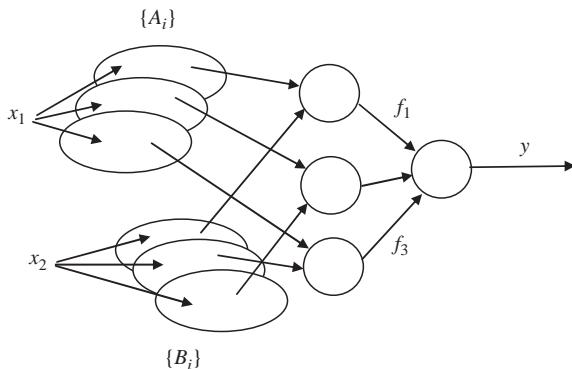


Figure 5.26 A schematic view of the two-input (x_1 and x_2) Takagi–Sugeno model with local regression models; the connections of the output unit realize processing through the local models f_i

devote a separate chapter to cover the fundamentals and algorithmic developments of fuzzy rule-based computing.

5.7.2 Key Phases of the Development and Use of Fuzzy Models

There are several fundamental schemes that support the design and the use of fuzzy models. Referring to Figure 5.27, we encounter four essential modes of their usage:

- The use of numeric data and generation of results in the numeric form (Figure 5.27a). This mode reflects a large spectrum of modeling scenarios we typically encounter in system modeling. Numeric data available in the problem are transformed through the interfaces and used to construct the processing core of the model. Once developed, the model is then used in a numeric fashion: it accepts numeric entries and produces numeric values of the corresponding output. From the perspective of the external ‘numeric’ world, the fuzzy model manifests itself as a multivariable non-linear input–output mapping. Later on, we discuss the non-linear character of the mapping in the context of rule-based systems. It will be demonstrated how the form of the mapping depends directly on the number of the rules, membership functions of fuzzy sets used there, inference scheme, and other design parameters. Owing to the number of design parameters, rule-based systems bring in a substantial level of modeling flexibility and this becomes highly advantageous to the design of fuzzy models.
- The use of numeric data and a presentation of results in a granular format (through some fuzzy sets) (Figure 5.27b). This mode makes the model highly user centric. The result of modeling comes as a collection of elements with the corresponding degrees of membership and in this way it becomes more informative and comprehensive than a single numeric quantity. The user/decision maker is provided with preferences (membership degrees) associated with a collection of possible outcomes.
- The use of granular data as inputs and the presentation of fuzzy sets as outcomes of the models (Figure 5.27c). This scenario is typical for granular modeling in which instead of numeric data we encounter a collection of linguistic observations, such as expert’s judgments, readings coming from unreliable sensors, outcomes of sensors summarized over some time horizons, etc. The results presented in the form of fuzzy sets are beneficial for the interpretation purposes and support the user-centric facet of fuzzy modeling.
- The use of fuzzy sets as inputs of the model and a generation of numeric outputs of modeling (Figure 5.27d). Here we rely on expert opinions as well as granular data forming aggregates of detailed numeric data. The results of the fuzzy model are then conveyed (through the interface) to the numeric environment in the form of the corresponding numeric output values. While this becomes feasible, we should be cognizant that the nature of the numeric output is not fully reflective of the character of the granular input.

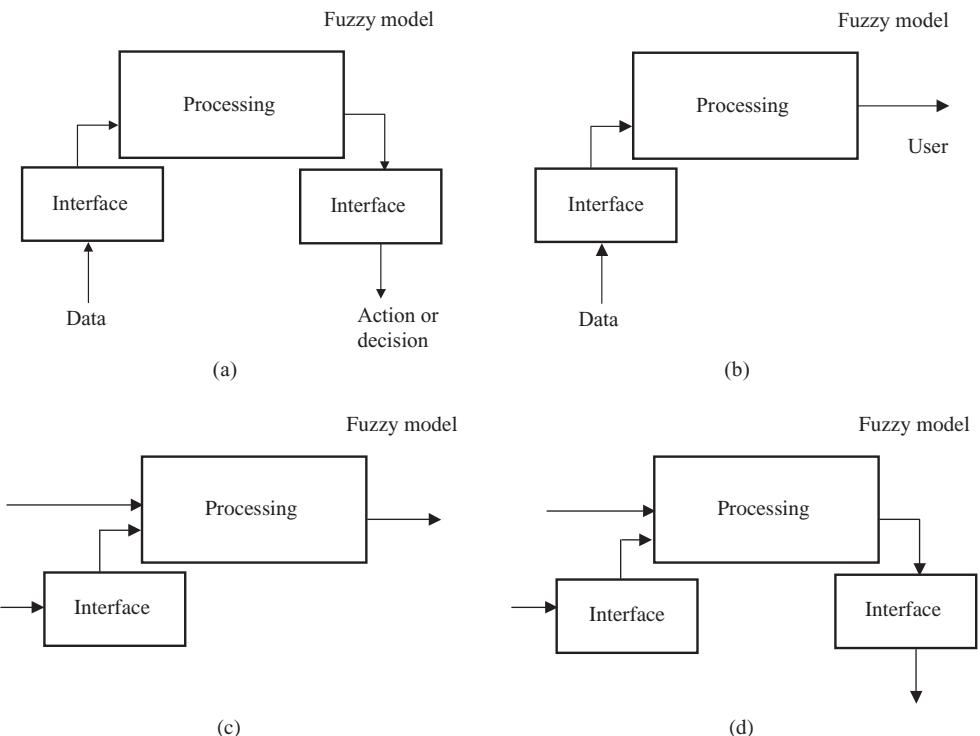


Figure 5.27 Four fundamental modes of the use of fuzzy models; note a role of input and output interfaces played in each of them (see the details in the text)

5.7.3 Main Categories of Fuzzy Models: An Overview

The landscape of fuzzy models is highly diversified. There are several categories of models where each class of the constructs comes with interesting topologies, functional characteristics, learning capabilities, and the mechanisms of knowledge representation. In what follows, we offer a general glimpse at some of the architectures which are the most visible and commonly envisioned in the area of fuzzy modeling.

Tabular fuzzy models are formed as some tables of relationships between the variables of the system granulated by some fuzzy sets (Figure 5.28). For instance, given two input variables with fuzzy sets A_1-A_3 and B_1-B_5 and the output fuzzy set C_1-C_3 , the relationships are articulated by filling in the entries of the table; for each combination of the inputs quantified by fuzzy sets, say A_i and B_j , we associate the corresponding fuzzy set C_k formed in the output space.

	B_1	B_2	B_3	B_4	B_5
A_1					
A_2			C_3		
A_3		C_2			C_1

Figure 5.28 An illustrative example of the two-input tabular fuzzy model (fuzzy decision table)

The tabular models produce a fairly compact suite of transparent relationships represented at the level of information granules. In the case of many input variables, we end up with multidimensional tables (relations). The evident advantage of the tabular fuzzy models resides with their evident readability. The shortcoming comes with a lack of existence of the direct mapping mechanisms. This means that we do not have any machinery of transforming input (either numeric or granular) into the respective output. Furthermore, the readability of the model could be substantially hampered when dealing with the growing number of variables we consider in this model.

Rule-based systems are highly modular and easily expandable fuzzy models composed of a family of conditional ‘if–then’ statements (rules) where fuzzy sets occur in their conditions and conclusions. The standard format of the rule with many inputs (conditions) comes in the form

$$\text{if condition}_1 \text{ is } A \text{ and condition}_2 \text{ is } B \text{ and } \dots \text{ and condition}_n \text{ is } W, \text{ then conclusion is } Z, \quad (40)$$

where A, B, C, \dots, W, Z are fuzzy sets defined in the corresponding input and output spaces.

The models support a principle of locality and a distributed nature of modeling as each rule can be interpreted as an individual local descriptor of the data (problem), which is invoked by the fuzzy sets defined in the space of conditions (inputs). The local nature of the rule is directly expressed through the support of the corresponding fuzzy sets standing in its condition part. The level of generality of the rule depends on many aspects that could be easily adjusted making use of the available design components associated with the rules. In particular, we could consider fuzzy sets of condition and conclusion whose granularity could be adjusted so that we could easily capture the specificity of the problem. By making the fuzzy sets in the condition part very specific (that is being of high granularity) we come up with the rule that is very limited and confined to some small region in the input space. When the granularity of fuzzy sets in the condition part is decreased, the generality of the rule increases. In this way the rule could be applied to more situations. To emphasize a broad spectrum of possibilities emerging in this way, refer to Figure 5.29, which underlines the very nature of the cases discussed above.

While the rules discussed so far form a single-level structure (the rules are built at the same level), there are also hierarchical architectures composed of several levels of knowledge representation, where there are collections of rules formed at a few very distinct levels of granularity (generality) (refer to

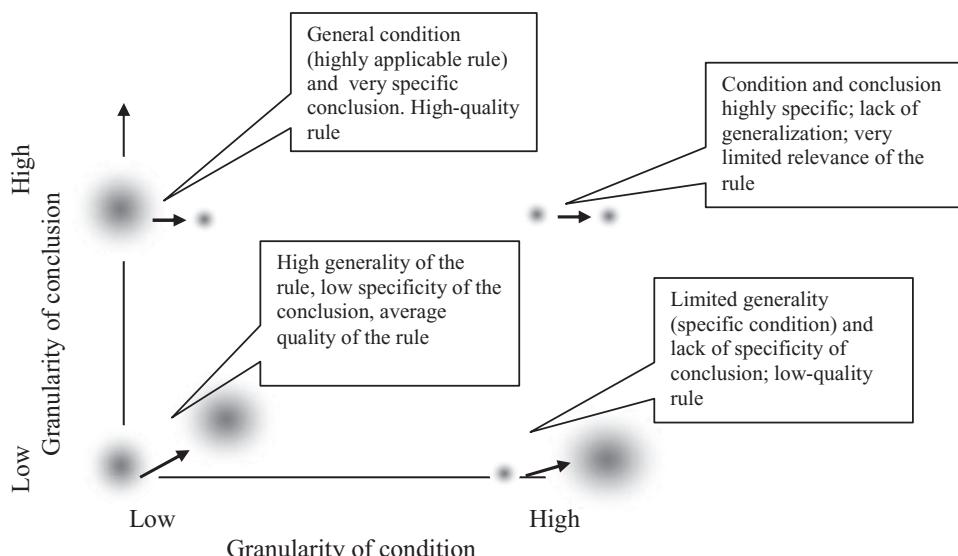


Figure 5.29 Examples of rules and their characterization with respect to the level of granularity of condition and conclusion parts

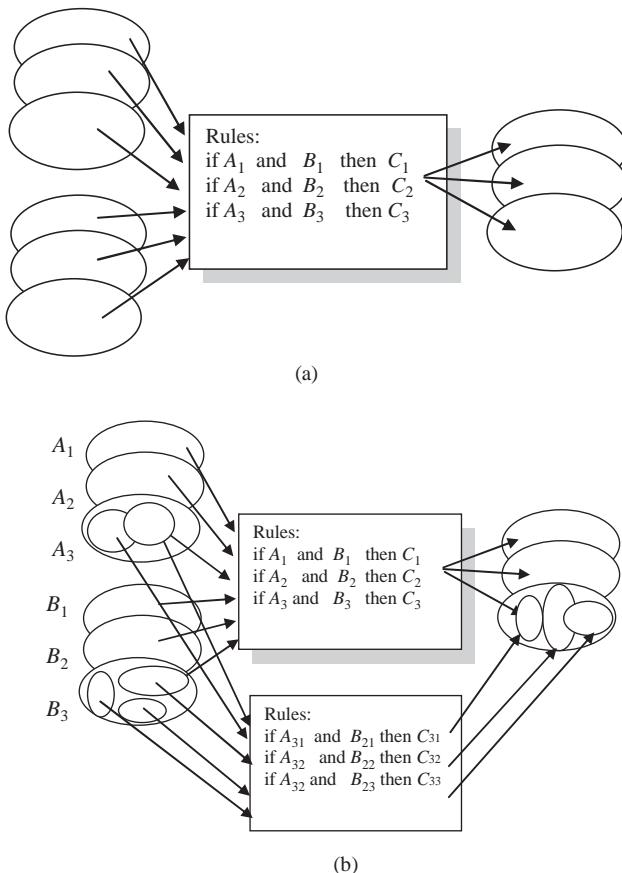


Figure 5.30 Examples of rule-based systems: (a) single-level architecture with all rules expressed at the same level of generality, (b) rules formed at several levels of granularity (specificity) of fuzzy sets standing in the condition parts of the rules. A_i and B_j stand for fuzzy sets forming the condition part of the rules

Figure 5.30). The level of generality of the rules is directly implied by the information granules forming the input and output interfaces.

Fuzzy relational models and associative memories are examples of constructs whose computing dwells on logic processing of information granules. The spirit of the underlying architectures is that the mapping between input and output information granules is realized in the form of some relational transformation (Figure 5.31). The mapping itself comes in the form of a certain composition operator (say, max-min

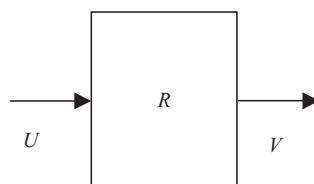


Figure 5.31 A schematic view of the relational models and associative memories

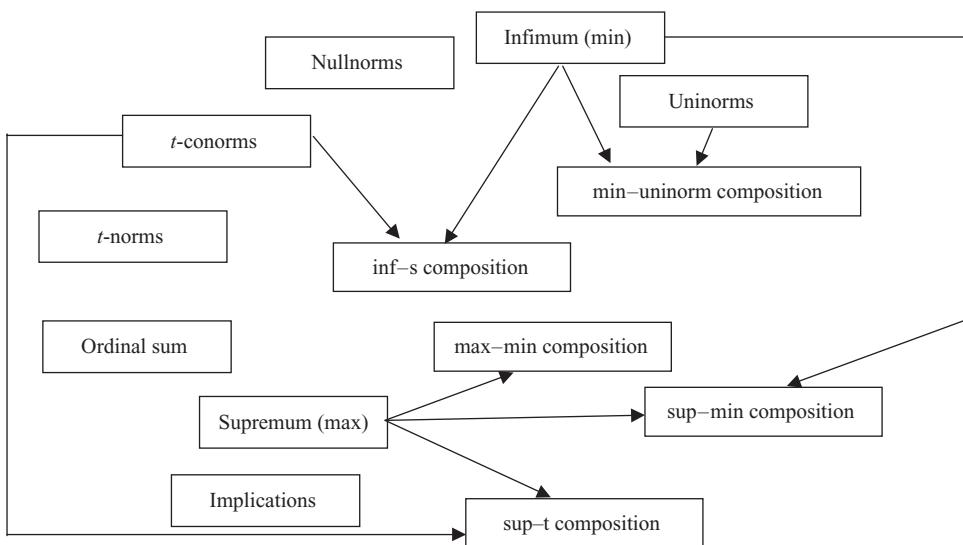


Figure 5.32 A general taxonomy of fuzzy relational equations (modeling structures) presented with respect to the combination of composition operators used in their realization

or being more general, $s-t$ composition). The same development scheme applies to fuzzy associative memories. The quality of recall carried out in the presence of incomplete or distorted (noisy) inputs is regarded as one of the leading indicators describing the performance of the memories. Given the input and associated output items – fuzzy sets A_1, A_2, \dots, A_c and B_1, B_2, \dots, B_c , respectively – we construct a fuzzy memory (relation) storing all pairs of items by OR-wise aggregating the Cartesian products of the input–output pairs, $R = \cup(A_{k=1}^N \times B_k)$. Next any input item A leads to the recall of the corresponding output B through some relational operator (say, max–min composition), $V = U \circ R$.

In fuzzy relational equations (which constitute an operational framework of associative memories) we encounter a wealth of architectures that is driven by a variety of composition operators. While the sup-t (max–min) composition is commonly used, there are also other alternatives available; some of them are presented in Figure 5.32. The selection of the composition operation (hence the form of the equation) depends on the problem at hand as each composition operator comes with its own well-defined semantics (with the underlying logic underpinnings and interpretation abilities). From the algorithmic standpoint, let us note that some of the composition operators lead to fuzzy relational equations for which we could derive analytical solutions. In other more advanced cases, one has to proceed with some numeric optimization and develop pertinent learning schemes.

Fuzzy decision trees are generalizations of well-known and commonly used decision trees [29, 50–53]. In essence, a decision tree is a directed acyclic graph whose nodes are marked by the attributes (input variables of the model) and links are associated with the discrete (finite) values of the attributes associated with the corresponding nodes (Figure 5.33). The terminal nodes concern the values of the output variable (which depending on the nature of the problem could assume discrete or continuous values).

By traversing the tree starting from the root node, we arrive at one of its final nodes. In decision tree only one terminal node can be reached as the values of the inputs uniquely determine the path one traverses through the tree. In contrast, in fuzzy decision trees several paths could be traversed in parallel. When moving down a certain path, several alternative edges originating from a given node are explored where each of them comes with the degree of matching of the current data (more specifically the value of the attribute that is associated with the node) and the fuzzy sets representing the values of the attribute coming with each node. The reachability of the node is computed by aggregating the degrees of matching along the path that has been traversed to reach it. Typically, we use here some t -norm as we adhere to

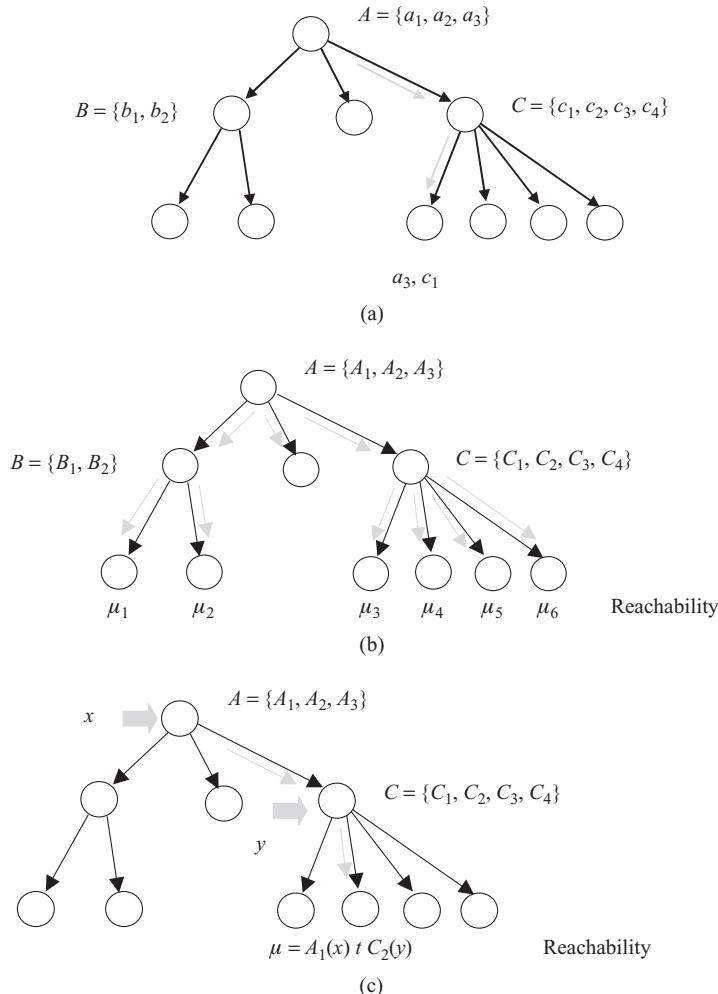


Figure 5.33 An example of (a) the decision tree and (b) the fuzzy decision tree; in this case, one can reach several terminal nodes at different levels of reachability (μ_i). The level of reachability is determined by aggregating activation levels along the path leading to the terminal node (c)

the *and*-like aggregation of the activation levels reported along the edges of the tree visited so far. In this way, several terminal nodes are reached and each of them comes with its own value of the reachability index computed by an *and* aggregation (using some *t*-norm) of the activation (matching) degrees between the data and the values of the attributes represented as fuzzy sets. The pertinent details are illustrated in Figure 5.33c.

Fuzzy neural networks are fuzzy-set-driven models composed with the aid of some logic processing units – fuzzy neurons [16, 45, 54]. These neurons realize a suite of generic logic operators (such as AND, OR, INCLUSION, DOMINANCE, SIMILARITY, and DIFFERENCE). Each neuron comes with a collection of the connections (weights). These weights bring a highly required flexibility to the processing units that could be exploited during the learning of the network. From the perspective of the topology of the network, we can envision several well-delineated layers of the processing units (see Figure 5.34).

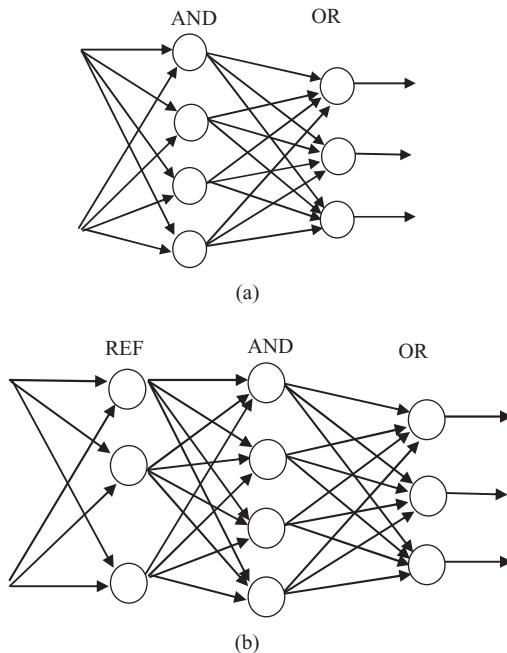


Figure 5.34 Examples of architectures of fuzzy neural networks: generalized (multivalued) logic functions realized in terms of AND and OR neurons (a), and the network with an auxiliary referential layer consisting of referential neurons (b)

There are some interesting linkages between the fuzzy neural networks and the relational structures (fuzzy relational equations) we have discussed earlier. Both of them rely on the same pool of composition operators (logic mappings); however, when it comes to the networks, those are typically a multilayer architectures.

Network of fuzzy processing units. The essence of these modeling architectures is to allow for a higher level of autonomy and flexibility. In contrast to the fuzzy neural networks, there is no layered structure. Rather than that, we allow for loosely connected processing units that can operate individually and communicate with the others. Furthermore when dealing with dynamic systems, the network has to exhibit some recurrent links.

One of the interesting and representative architectures in this category are fuzzy cognitive maps [55–58]. These maps, being the generalization of the binary concepts introduced by Axelrod [59], represent concepts and show linkages between them. A collection of basic concepts are represented as nodes of the graph, which are interrelated through a web of links (edges of the graph). The links could be excitatory (so the increase of intensity of one concept triggers the increased level of manifestation of the related one) or inhibitory (in which case we see an opposite effect: the increase of intensity of one concept triggers the decline of intensity of the other one). Traditionally, the connections (links) assume numeric values from -1 to 1 . An example of the fuzzy cognitive map is shown in Figure 5.35.

The detailed computing realized at the level of the individual node is governed by the expression $x_j = f(\sum_{\substack{j=1 \\ j \neq i}}^n w_{ji}x_j)$, where ' x_j ' denotes a resulting level of activity (intensity) at the node of interest and x_i is the intensity level associated with the i th node. The connections (linkages) between the two nodes are denoted by w_{ij} . The non-linear mapping ' f ' is typically a monotonically increasing function, say a sigmoid one, $f(u) = 1/(1 + \exp(-u))$. A node could be equipped with its own dynamics (internal feedback loop) and in this case we consider a non-zero link w_{jj} for this particular node. Given this, we arrive at a recurrent expression of the form $x_j = f(\sum_{j=1}^n w_{ji}x_j)$. The interfaces of this fuzzy model are

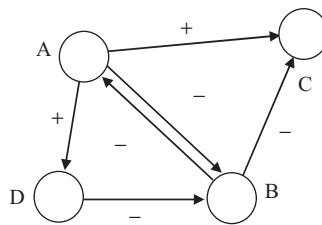


Figure 5.35 An example of a fuzzy cognitive map composed of four concepts (A, B, C, and D). The sign of the corresponding connection identifies the effect of inhibition (−) or excitation (+) between the concepts (nodes). For instance, A excites C

not shown explicitly; however, we should have in mind that the inputs to the nodes are the inputs from the modeled world that were subject to a certain transformation realized through some fuzzy sets defined in the corresponding variables or the Cartesian products of these variables.

The structure of the network offers a great deal of flexibility and is far less rigid than the fuzzy neural networks where typically the nodes (neurons) are organized into some layers. The individual nodes of the fuzzy cognitive maps could be realized as some logic expressions and implemented as AND or OR neurons. The connections could assume values in [0, 1] and the inhibitory effect can be realized by taking the complement of the activation level of the node linked to the one under consideration. An example of the logic-based fuzzy cognitive map is presented in Figure 5.36.

In all these categories of fuzzy models we can envision a hierarchy of the structures that could be formed at each level of the hierarchy [60]. We start from the highest, the most general level and expand it by moving down to capture more details. In general, we can envision a truly hierarchical structure shown in Figure 5.37.

A more specific and detailed visualization of the hierarchy of the model is shown in Figure 5.38, where we are concerned with fuzzy cognitive maps. Here, a certain concept present at the higher level and represented as one of the nodes of the map unfolds into several subconcepts present at the lower level. The computing occurring at the lower, more detailed level produces some level of activation of the more detailed nodes (subconcepts) and these levels aggregated OR-wise are then used in computing realized at the higher level of generality.

5.7.4 Verification and Validation of Fuzzy Models

The processes of verification and validation (referred to as V&V) are concerned with the fundamental issues of the development of the model and assessment of its usefulness. Following the standard

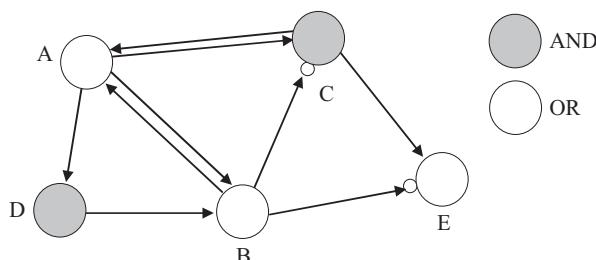


Figure 5.36 An example of a fuzzy cognitive map whose nodes are realized as logic expressions (AND and OR neurons). The inhibitory effect is realized by taking the complement of the activation level of the interacting node (here indicated symbolically by a small dot)

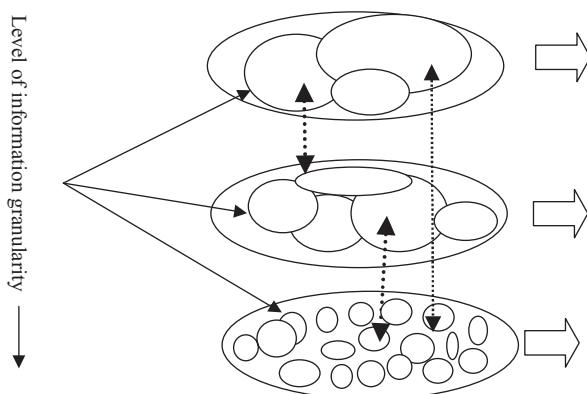


Figure 5.37 A general concept of hierarchy in fuzzy modeling; depending on a certain level of specificity, various sources of data could be accommodated, processed, and afterward the results communicated to the modeling environment

terminology (which is well established in many disciplines, such as software engineering), verification is concerned with the analysis of the underlying processes of constructing the fuzzy model. Are the design principles guiding a systematic way the model is built fully adhered to? In other words, rephrasing the concept in the setting of software engineering, we are focusing on the following question: ‘Are we building the product *right*?’ Validation, on the other hand, is concerned with ensuring that the model (product) meets the requirements of the customer. Here, we concentrate on the question ‘Are we building the *right* product?’ Put it differently, is the resulting model in compliance with the expectations (requirements) of the users or groups of users of the model?

Let us elaborate on the verification and validation in more detail.

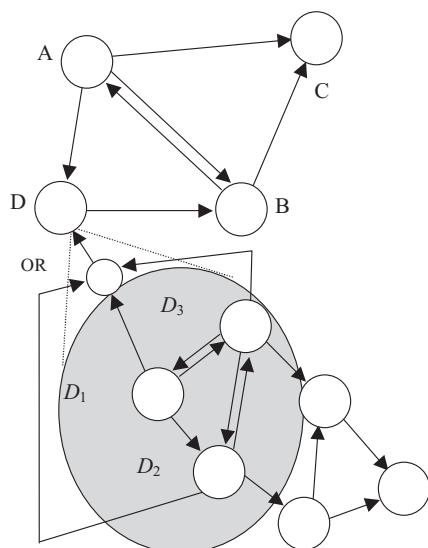


Figure 5.38 An example of a hierarchy of fuzzy cognitive maps; a certain concept at the higher level of generality is constructed as a logic OR-type of aggregation of the more detailed ones used in the cognitive map at the higher level of detail

Verification of Fuzzy Models

Fuzzy models and fuzzy modeling are the pursuits that in spite of their specific requirements still adhere to the fundamentals of system modeling. In this sense, they also have to follow the same principles of model verification. There are several fundamental guidelines with this respect. Let us highlight the essence of them:

- (a) An iterative process of constructing a model in which we successively develop a structure of the model and estimate its parameters. There are well-established estimation procedures which come with a suite of optimization algorithms. It is quite rare that the model is completely built through a single pass through these two main phases.
- (b) Thorough assessment of accuracy of the developed model. The underlying practice is that one should avoid any bias in the assessment of this quality, especially by developing a false impression about the high accuracy of the model. To avoid this and gain a significant level of objective evaluation, we split the data into training and testing data. While the model is constructed, we use the training data.
- (c) Generalization capabilities of the resulting model. While the accuracy is evaluated for the experimental data being used to construct the model, the accuracy obtained quantified in this way could lead to highly and optimistically biased evaluation. The assessment of the performance of the model on the testing data helps eliminate this shortcoming.
- (d) The lowest possible complexity of the model. This is usually referred to as an Occam's razor principle. The principle states that among several models of a very similar accuracy we always prefer a model that is the simplest. The concept of simplicity requires some clarification as the concept itself is not straightforward as one might have envisioned. If you consider a collection of polynomial models, linear models are obviously simpler than those involving second- or higher order polynomials. On the other hand, the notion of complexity could also carry a subjective component. For instance, it could significantly depend on the preferences of designers and users of the model. In a certain environment where neurocomputing is dominant, models of neural networks are far more acceptable and therefore perceived as being simpler than polynomial models. One should stress, however, that this type of assessment comes with a substantial level of subjectivity.
- (e) High level of design autonomy of the model. Given that usually we encounter a significant number of design alternatives as to the architecture of the model, various parameters one can choose from to construct the detailed topology of the model, it is highly desirable to endow the development environment of the model with a significant level of design autonomy, i.e., exploit suitable optimization techniques that offer a variety of capabilities aimed at the structural development of the model. With this regard, evolutionary techniques of optimization play a pivotal role. Their dominant features such as population-based search, minimal level of guidance (a suitable fitness function is just a suitable mechanism guiding optimization efforts), and collaborative search efforts (through some mechanisms of communication between the individual solutions) are of particular interest in this setting. The design of fuzzy models in the presence of a number of objectives is an example of multiobjective optimization in which the objectives are highly conflicting. The set of efficient solutions called non-dominated Pareto optimal is formed by all elements in the solution space for which there is no further improvement of without degradation in other design objectives. Hence the machinery of genetic optimization becomes a highly viable and promising alternative.

When constructing fuzzy models, we also adhere to the same principles (viz. iterative development and successive refinements, accuracy assessment through training and testing sets, and striving for the lowest possible complexity of the construct). When it comes to the evaluation of accuracy of the fuzzy models, it is worth stressing that given the topology of these models in which the interface module constitutes an integral part [38, 61–63], there are two levels at which the accuracy of the model can be expressed. We may refer to them as an internal and external level of accuracy characterization. Their essence is schematically visualized in Figure 5.39. At the external (viz. numeric) level of accuracy quantification, we compute a distance between the numeric data and the numeric output of the model resulting from the transformation realized by the interface of the model. In other words, the performance index expressing

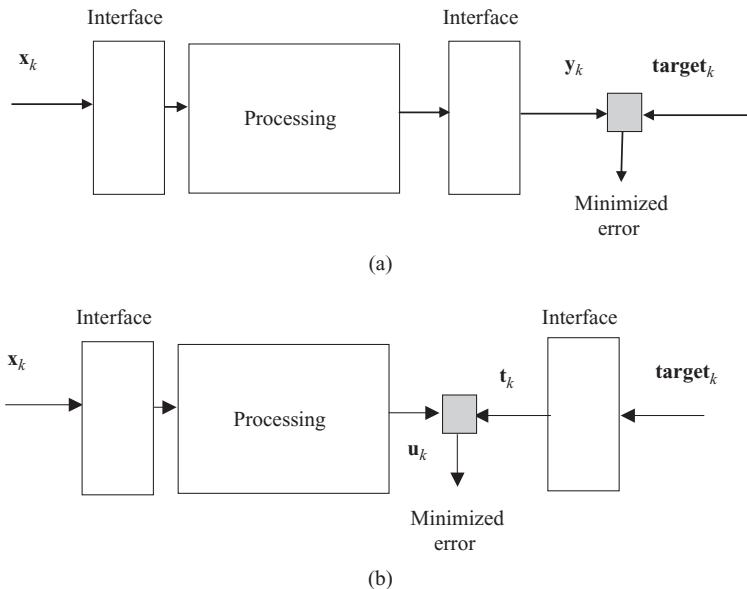


Figure 5.39 Two fundamental ways of expressing the accuracy of fuzzy models: (a) at the numeric level of experimental data and results of mapping through the interface, and (b) at the internal level of processing after the transformation through the interfaces

the (external) accuracy of the reads in the form

$$Q = \sum_{k=1}^N || y_k - \text{target}_k ||^2, \quad (41)$$

where the summation is carried out over the numeric data available in the training, validation, or testing set. The form of the specific distance function (Euclidean, Hamming, Tchebyschev, or more generally, Minkowski distance) could be selected when dealing with the detailed quantification of the proposed performance index.

At the internal level of assessment of the quality of the fuzzy model, we transform the output data through the output interface, so now they become vectors in the $[0, 1]^m$ hypercube and calculate distance at the internal level by dealing with two vectors with the $[0, 1]$ entries. As before, the calculations may involve training, validation, or testing data. More specifically, we have

$$Q = \sum_{k=1}^N || u_k - t_k ||^2 \quad (42)$$

(refer also to Figure 5.39b). These two ways of quantifying the accuracy are conceptually different and there is no equivalence between them unless their granular to numeric interface does not produce any additional error. This issue is elaborated in Chapter 10 when dealing with the matter of interoperability. Quite often the interface itself could introduce an additional error. In other words, we may have a zero error at the level of granular information; however, once we transform these results through the interface, they become associated with the non-zero error. The performance index in the form shown above (3)–(4) is computed at either the numeric or the granular level. In the first case, refer to Figure 5.39a, it concerns the real numbers. At the level of information granules, the distances are determined at the level of the elements located in the unit hypercubes $[0, 1]^m$ (see Figure 5.39b).

Training, Validation, and Testing Data in the Development of Fuzzy Models

When assessing the quality of the fuzzy models in terms of accuracy, and stability, transparency, it is important to quantify these features in an environment we could gain a high confidence on as to the produced findings. Having overly optimistic evaluations is not advisable. Likewise, producing some pessimistic bias is not helpful as well. In order to strive for high reliability of the evaluation process of the model, one should release results of assessment on the basis of a prudent use of available data. Here we follow the general guidelines and evaluation procedures encountered in system modeling (and these guidelines hold in spite of the diversity of the models and their underlying fundamentals).

Splitting of Data into Training and Testing Subsets

To avoid any potential bias, the available data are split randomly into two disjoint subsets of training and testing data (with the split of 60–40% where 60% is used in the training set). The model is built using the training data. Next its performance is evaluated on the testing data. As this data set has not been used in the development of the model, we avoid any potential bias in its evaluation.

Tenfold Cross-Validation

While the use of the training and testing data helps gain some objectivity in the assessment, there could still be some variability in the evaluation, which could be a result of the random split. To reduce this effect, we randomly split the data into training-testing subsets, evaluate the performance of the model, and repeat the split and evaluation ten times in each case producing a random split of the data. In this sense, the obtained results could help reduce variability. Both the mean value of the performance and the related standard deviation are reported. When preparing the data, the split is typically carried out at the level of 90–10%.

Leave-One-Out Evaluation Strategy

This strategy is of particular relevance when dealing with small data sets in which case the 60–40% split is not justifiable. Consider, for instance, 20 data points (which could be quite typical when dealing with data coming from some software projects; we do not have hundreds of those). In this case, the use of the approaches presented above could be quite unstable – as a matter of fact the number of data points is quite low, say 12 data for the training purposes; hence, the development of the fuzzy model could be affected by the reduced size of the data. In this case we consider a leave-one-out strategy. Here, we use all but one data point in the training data, construct the model, and evaluate its performance on the one left out from the training data. The process is repeated for all data points, starting from the first one left out, building the model, and testing it on the single data point. Thus for ‘ N ’ data, this strategy produces results of performance on all points being left out. Then average and standard deviation of the results could serve as a sound assessment of the quality of the fuzzy model.

So far, we have indicated that the available data set is split into its training and testing part. Quite often we also use a so-called validation set. The role of the validation set is to guide the development of the model with respect to its structural optimization. For instance, consider that we are at position to adjust the number of fuzzy sets defined for individual variables. It is quite anticipated that when we start increasing the number of fuzzy sets, the accuracy of the model on the training set is going to become better. It is very likely that the tendency on the testing set is going to be quite different. The question as to the ‘optimal’ number of fuzzy sets cannot be answered on the basis of the training data. The testing set is not supposed to be used at all in the construction of the model. To solve the problem, in addition to the training data set, we set aside a portion of data – validation set that is used to validate the model constructed on the basis of the training data. The development process proceeds as follows: we construct a model (estimate its parameters, in particular) on the basis of the training set. When it comes to the structural development (say, the number of fuzzy sets and alike) where there is a strong monotonic tendency, we have to resort ourselves to the validation set: choose the value of the structural parameter (say, the number of nodes, processing units, fuzzy sets, etc.), optimize the model on the training set, and check

its performance on the validation set. Select the value for which we get the best results on the validation set.

Validation of Fuzzy Models

As already indicated, validation is focused on the issues related to the question ‘are we building the right system?’ In essence, the term of validation is inherently multifaceted. It embraces several important aspects, in particular transparency and stability of the model. Let us discuss them in more detail.

Transparency of the Model

The interpretation of transparency or ‘readability’ of the model is directly associated with the form of the fuzzy model (64, 65). The essence of this feature of the model is associated with the ability to easily comprehend the model, namely pick up the key relationships captured by it. There is also a substantial level of flexibility in the formalization of the concept of transparency. For instance, consider a rule-based model which is composed of a series of rules ‘if condition₁ and condition₂ and ... and condition_n then conclusion,’ where the conditions and conclusions are quantified in terms of some fuzzy sets. The transparency of the model can be quantified by counting the number of rules and taking into consideration a complexity of each rule. This complexity can be expressed by counting the number of the conditions standing in the rule. The larger the number of the rules and the longer they are, they lower the readability (transparency) of the model. When dealing with network type of fuzzy models, such as fuzzy cognitive maps of an immediate criterion, we may take into consideration the number of nodes or the number of connections between them (or alternatively the density of connections which is determined by counting the number connections and dividing them by the number of nodes). The higher these values, the more difficult it becomes to ‘read’ the model and interpret it in a meaningful manner. The transparency of the model is also essential when dealing with an ability to accommodate any prior domain knowledge that is available in any problem solving. The existing components of such domain knowledge are highly instrumental in the development of the models. For instance, one could easily reduce learning effort going toward the estimation of the parameters of the model (say, a fuzzy neural network) once the learning starts from a certain promising point in the usually huge search space.

Stability of the Model

The substantial value of fuzzy models comes with their stability. We always prefer a model that is ‘stable’ so that it does not change over some minor variations of the environment, and experimental data in particular. Practically, if we take some subsets of training data, we anticipate that the resulting fuzzy model does not radically change and retains its conceptual core, say a subset of rules that are essential descriptors of the phenomenon or the process of interest. Some minor variations of other less essential rules cannot be avoided and are less detrimental to the overall stability of the model. There could also be some changes in the numeric values of the parameters of the model, yet their limited changes could be secondary to the stability of the model. The aspect of model’s stability is somewhat associated with the transparency we have considered so far: once provided with the model, we expect that it concentrates on these aspects of the reality that repeats all time in spite of the variations of the environment. By the same token we intend to avoid some highly variable components of the model as not contributing to the essence of the underlying phenomenon. Intuitively, one could conclude that the stability is inherently associated with the level of granularity we establish for the description. A general tendency is not surprising at all: the higher the generality, the higher the stability of the model.

It should be stressed that these fundamental features of the fuzzy models could be in competition. High accuracy could reduce readability. High transparency could come at the cost of reduced accuracy. Always a sound compromise should be strived for. A suitable choice depends on the relationships between these characteristics. Some examples are illustrated in Figure 5.40. Reaching a compromise should position us at a point where abrupt changes are avoided.

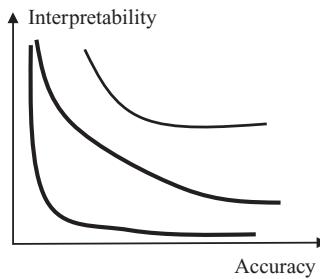


Figure 5.40 Examples of relationships between interpretability and accuracy of fuzzy models; the character of these dependencies is specific to the type of the fuzzy model under considerations

5.8 Conclusion

We have introduced the underlying concepts of fuzzy sets stressing their origin, elaborating at their semantics and knowledge representation issues, and discussing an array of methodological problems of modeling with fuzzy sets (fuzzy modeling). Fuzzy sets constitute one of the interesting realizations of granular computing. The hierarchy of granular constructs is apparent when forming families of fuzzy sets. We presented taxonomy of estimation techniques and reviewed several generic algorithms used to construct membership functions. We have underlined a multifaceted nature of fuzzy modeling by emphasizing a collection of design criteria which are inherent to the development of granular models.

References

- [1] J. Łukasiewicz. O logice trójwartościowej. *Ruch Filozoficzny* 5 (1920) 170.
- [2] J. Łukasiewicz. Philosophische Bemerkungen zu mehrwertigen Systemen des Aussagenkalk. *C. R. Soc. Sci. Lett. Varsovie* 23 (1930) 51–77.
- [3] J. Łukasiewicz. Selected works. In: L. Borkowski (ed), *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, 1970.
- [4] A. Korzybski. *Science and Sanity: An Introduction to Non-Aristotelian Systems and General Semantics*, 3rd ed. The International Non-Aristotelian Library Publishing, Lakeville, CT, 1933.
- [5] L.A. Zadeh. Fuzzy sets. *Inf. Control.* 8 (1965) 338–353.
- [6] L.A. Zadeh. Outline of a new approach to the analysis of complex system and decision process. *IEEE Trans. Syst. Man Cybern.* 3 (1973) 28–44.
- [7] L.A. Zadeh. Fuzzy sets and information granularity. In: M.M. Gupta, R.K. Ragade, and R.R. Yager, (eds), *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, 1979, pp. 3–18.
- [8] L.A. Zadeh. Fuzzy logic = Computing with words. *IEEE Trans. Fuzzy Syst.* 4 (1996) 103–111.
- [9] L.A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–117.
- [10] L.A. Zadeh. From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions. *IEEE Trans. Circuits Syst.* 45 (1999) 105–119.
- [11] L.A. Zadeh Toward a generalized theory of uncertainty (GTU) – an outline. *Inf. Sci.* 172 (2005) 1–40.
- [12] H.J. Zimmermann. *Fuzzy Set Theory and Its Applications*. 3rd ed. Kluwer Academic Publishers, Norwell, MA, 1996.
- [13] S. Gottwald. Mathematical fuzzy logic as a tool for the treatment of vague information. *Inf. Sci.* 172(1–2) (2005) 41–71.
- [14] R. Babuska. *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Dordrecht, 1998.
- [15] W. Pedrycz. *Computational Intelligence: An Introduction*. CRC Press, Boca Raton, FL, 1997.
- [16] W. Pedrycz and F. Gomide. *An Introduction to Fuzzy Sets*. MIT Press, Cambridge, MA, 1998.
- [17] J. Kacprzyk. *Multistage Decision-Making under Fuzziness*. TUV Verlag, Rheinland, Cologne, 1983.
- [18] J. Kacprzyk. *Multistage Fuzzy Control*. Wiley, Chichester, UK, 1997.
- [19] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, Upper Saddle River, NJ, 1995.

- [20] A. De Luca and S. Termini. A definition of nonprobabilistic entropy in the setting of fuzzy sets. *Inf. Control* 20 (1972) 301–312.
- [21] A. De Luca and S. Termini. Entropy of L-fuzzy sets. *Inf. Control* 24 (1974) 55–73.
- [22] W. Pedrycz and J. Valente de Oliveira. Optimization of fuzzy models. *IEEE Trans. Syst. Man Cybern. Part B* 26(4) (1996) 627–636.
- [23] W. Pedrycz and J. Valente de Oliveira. An algorithmic framework for development and optimization of fuzzy models. *Fuzzy Sets Syst.* 80 (1996) 37–55.
- [24] W. Pedrycz (ed.). *Fuzzy Modelling: Paradigms and Practice*. Kluwer Academic Press, Dordrecht, 1996.
- [25] W. Pedrycz and G. Vukovich. Granular neural networks. *Neurocomputing* 36(1–4) (2001) 205–224.
- [26] W. Pedrycz (ed.). *Granular Computing: An Emerging Paradigm*. Physica-Verlag, Heidelberg, 2001.
- [27] G.A. Miller. The magical number seven plus or minus two: some limits of our capacity for processing information. *Psychol. Rev.* 63 (1956) 81–97.
- [28] J. Valente de Oliveira. On optimal fuzzy systems with I/O interfaces. In: *Proceedings of the Second International Conference on Fuzzy Systems*, San Francisco, CA, 1993, 34–40.
- [29] Z. Qin and J. Lawry. Decision tree learning with fuzzy labels. *Inf. Sci.* 172(1–2) (2005) 91–129.
- [30] M. Chen and S. Wang. Fuzzy clustering analysis for optimizing fuzzy membership functions. *Fuzzy Sets Syst.* 103(2) (1999) 239–254.
- [31] M. Civanlar and H. Trussell. Constructing membership functions using statistical data. *Fuzzy Sets Syst.* 18(1) (1986) 1–13.
- [32] H. Dishkant. About membership functions estimation. *Fuzzy Sets Syst.* 5(2) (1981) 141–147.
- [33] J. Dombi. Membership function as an evaluation. *Fuzzy Sets Syst.* 35(1) (1990) 1–21.
- [34] T. Hong and C. Lee. Induction of fuzzy rules and membership functions from training examples. *Fuzzy Sets Syst.* 84(1) (1996) 389–404.
- [35] A. Medaglia, S. Fang, H. Nuttle, and J. Wilson. An efficient and flexible mechanism for constructing membership functions. *Eur. J. Oper. Res.* 139(1) (2002) 84–95.
- [36] S. Medasani, J. Kim, and R. Krishnapuram. An overview of membership function generation techniques for pattern recognition. *Int. J. Approx. Reason.* 19(3–4) (1998) 391–417.
- [37] W. Pedrycz. Why triangular membership functions? *Fuzzy Sets Syst.* 64 (1994) 21–30.
- [38] W. Pedrycz. Fuzzy equalization in the construction of fuzzy sets. *Fuzzy Sets Syst.* 119 (2001) 329–335.
- [39] D. Simon. H_∞ estimation for fuzzy membership function optimization. *Int. J. Approx. Reason.* 40(3) (2005) 224–242.
- [40] I. Turksen. Measurement of membership functions and their acquisition. *Fuzzy Sets Syst.* 40(1) (1991) 5–138.
- [41] C. Yang and N. Bose. Generating fuzzy membership function with self-organizing feature map. *Pattern Recognit. Lett.* 27(5) (2006) 356–365.
- [42] T. Saaty. *The Analytic Hierarchy Process*. McGraw Hill, New York, 1980.
- [43] T. Saaty. Scaling the membership functions. *Eur. J. Oper. Res.* 25(3) (1986) 320–329.
- [44] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer Academic Publishers, Dordrecht, 2003.
- [45] W. Pedrycz and G. Vukovich. On elicitation of membership functions. *IEEE Trans. Syst. Man Cybern. Part A* 32(6) (2002) 761–767.
- [46] W. Pedrycz. Logic-driven fuzzy modeling with fuzzy multiplexers. *Eng. Appl. Artif. Intell.* 17(4) (2004) 383–391.
- [47] W. Pedrycz and M. Reformat. Genetically optimized logic models. *Fuzzy Sets Syst.* 150(2) (2005) 351–371.
- [48] W. Pedrycz. From granular computing to computational intelligence and human-centric systems. *IEEE Connect.* 3(2) (2005) 6–11.
- [49] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modelling and control. *IEEE Trans. Syst. Man Cybern.* 15 (1985) 116–132.
- [50] B. Apolloni, G. Zamponi, and A. Zanaboni. Learning fuzzy decision trees. *Neural Netw.* 11 (1998) 885–895.
- [51] R. Chang and T. Pavlidis. Fuzzy decision tree algorithms. *IEEE Trans. Syst. Man, Cybern. SMC-7(1)* (1977) 28–35.
- [52] C. Janikow. Fuzzy decision trees: issues and methods. *IEEE Trans. Syst. Man, Cybern. Part B* 28(1) (1998) 1–14.
- [53] W. Pedrycz and Z. Sosnowski. The design of decision trees in the framework of granular data and their application to software quality models. *Fuzzy Sets Syst.* 123(3) (2001) 271–290.
- [54] A. Ciaramella, R. Tagliaferri, W. Pedrycz, and A. Di Nola. Fuzzy relational neural network. *Int. J. Approx. Reason.* 41(2) (2006) 146–163.
- [55] B. Kosko. Fuzzy cognitive maps. *Int. J. Man-Mach. Stud.* 24 (1986) 65–75.
- [56] B. Kosko. *Neural Networks and Fuzzy Systems*. Prentice Hall, Englewood Cliffs, NJ, 1992.
- [57] E.I. Papageorgiou, C. Stylios, and P.P. Groumpas. Unsupervised learning techniques for fine-tuning fuzzy cognitive map causal links. *Int. J. Hum. Comput. Stud.* 64(8) (2006) 727–743.

- [58] E. Papageorgiou and P. Groumpas. A new hybrid method using evolutionary algorithms to train fuzzy cognitive maps. *Appl. Soft Comput.* 5(4) (2005) 409–431.
- [59] R. Axelrod. *Structure of Decision: The Cognitive Maps of Political Elites*. Princeton University Press, Princeton, NJ, 1976.
- [60] O. Cordón, F. Herrera, and I. Zwig. A hierarchical knowledge-based environment for linguistic modeling: models and iterative methodology. *Fuzzy Sets Syst.* 138(2) (2003) 307–341.
- [61] G. Bortolan and W. Pedrycz. Linguistic neurocomputing: the design of neural networks in the framework of fuzzy sets. *Fuzzy Sets Syst.* 128(3) (2002) 389–412.
- [62] G. Bortolan and W. Pedrycz. An interactive framework for an analysis of ECG signals. *Artif. Intell. Med.* 24(2) (2002) 109–132.
- [63] C. Mencar, G. Castellano, and A. Fanelli. Interface optimality in fuzzy inference systems. *Int. J. Approx. Reason.* 41(2) (2006) 128–145.
- [64] J. Casillas, O. Cordon, F. Herrera, and Magdalena. (eds). *Interpretability Issues in Fuzzy Modeling*. Springer-Verlag, Berlin, 2003.
- [65] R. Paiva and A. Dourado. Interpretability and learning in neuro-fuzzy systems. *Fuzzy Sets Syst.* 147(1) (2004) 17–38.

6

Measurement and Elicitation of Membership Functions

Taner Bilgiç and İ. Burhan Türkşen

6.1 Introduction

Granular computing involves working with granular information entities that are abstracted at different levels [1]. If these entities are measured at the lowest numerical level and then aggregated for higher levels, their measurement is keen to measurement of membership functions in fuzzy set theory, which has been an elusive problem since the conception of fuzzy sets [2]. We have argued elsewhere that the measurement is closely linked with different interpretations of membership function [3]. After almost 10 years we find that the landscape has not changed much on that front and a ‘cookbook recipe’ for measuring membership functions is still not available. However, there are heartening new results and developments that are the main concern of this review.

Two types of progress are apparent in the recent literature: (i) theoretical development and (ii) new empirical models. Therefore this review focuses on these two types of developments separately.

We first start with the formal definition of a membership function [2]. A fuzzy (sub)set, say F , has a membership function μ_F , defined as a function from a well-defined universe (the referential set), X , into the unit interval as $\mu_F : X \rightarrow [0, 1]$. One can also view another type of membership. Let $\mathcal{F} = \{F_1, F_2, \dots\}$ be a set of fuzzy (sub)sets. A membership function $\nu_X : \mathcal{F} \rightarrow [0, 1]$ denotes the membership of an element of the referential set X to different fuzzy sets.

Anyone who is to use fuzzy sets must answer the following questions:

1. What does graded membership mean?
2. How is it measured?
3. What operations are meaningful to perform on it?

There are various interpretations (likelihood, random set, similarity, and utility) of the membership function. A taxonomy based on various interpretations of membership functions appear in [3]. In this review, we will concentrate on measurement-theoretic aspects (Section 6.2) and elicitation methods (Section 6.3). We borrow from [3] for summarizing elicitation methods augmenting it with new developments. We close the chapter with a remark and a summary in Section 6.4.

6.2 Measurement-Theoretic Approaches

Measurement theory is concerned with *representation* and *meaningfulness* of a particular measurement scheme [4–8]. It is a ‘theory’ about measurement and its objective is to lay down axioms in terms of observable primitives that make measurement possible.

If fuzzy set membership is something measurable, measurement theory should help in pinning down how this measurement should be done.

We have emphasized the difference between two related measurement problems: (i) membership measurement and (ii) property ranking. This distinction is first recognized in [9]. Both types of measurements received considerable attention. However their combination, membership to many fuzzy sets as judged by different subjects, still warrants more attention.

The scales resulting from membership measurement and property ranking may not be necessarily used to measure the same entity. The degree with which a subject belongs to a certain fuzzy set may not be equal to the degree that fuzzy set is associated with him or her (e.g., consider John, who is a basketball player, and consider his tallness. Among all the other attributes of John, let his tallness be the least that one can associate with him. Therefore, $v_{\text{John}}(\text{tall}) = 0$. But when compared with other (ordinary) people $\mu_{\text{tall}}(\text{John}) > 0$).

Measurement-theoretic development in membership measurement upto 2000s has been carried out in [9–15] among others. This development is also reviewed in various sources [3, 16]. Hence, we will not repeat the development once again in this review. Instead, we will survey recent results and comment on how they fit with the rest of the development.

One way to pursue in combining the membership measurement and property ranking problems is to introduce a new structure where the resulting measurement scale necessarily measures the membership degree in a fuzzy set. Following studies have recognized this fact.

In a measurement-theoretic framework, Bollmann-Sdorra *et al.* [9] justify the use of min and max operators as intersection and union, respectively. Their representation is ordinal.

Bilgiç and Turksen [17] introduce two ways this combination can be made. In the first one, the two different problems are simply cast into a bounded semigroup structure. The consequences of this model are analyzed. It is argued that since accepting the Archimedean axiom can be very hard for some fuzzy terms, the ratio scale representations are not likely to arise.

Marchant [18] takes up the property ranking problem in the context of measurement theory. His setting considers statements of the following forms:

- The agent x belongs more to fuzzy set F than to G .
- The agent x belongs more to fuzzy set F than to $F \cup G$.
- The agent x belongs more to fuzzy set F than to the complement of F .

As in [9], by insisting on union and intersection to be idempotent (i.e., $F \cup F = F$ and $F \cap F = F$) Marchant derives maximum and minimum as the possible union and intersection operators, respectively, when only an underlying ordering is available. He then enriches the underlying algebraic structure to cardinal scales and still insisting on idempotent operators, derives max and min as the only reasonable operators for cardinal scales. This result is not surprising when one insists on idempotent operators as min is the only idempotent continuous t -norm. For example if one insists on the law of excluded middle, or contradiction, other representations would be available. Nevertheless, this result justifies the use of min and max on cardinal scales along with strong negation.

Marchant [19] studies the same measurement problem using subjective ratio estimation. The subject is required to provide information on ratios of membership values rather than membership values themselves. From the ratios, a representation of the membership function follows. Marchant first provides a very weak representation and then an exceedingly strong representation based on ratios. Realizing that both representations are not useful, he studies other representations within those two extremes. There are various representations with different scale strengths. All these representations are constructed without considering union, intersection, or negation. When the author considers modeling union and intersection in this context, he once again insists on an idempotent operator (i.e., membership in fuzzy set ‘Young’

and fuzzy set ‘Young and Young’ are to be judged equal by any subject) and derives max and min as the only possible solutions model union and intersection.

Marchant [20] considers the problem of representing a membership in a fuzzy set by a trapezoidal membership function. Trapezoidal functions are widely used in practice. Using a measurement-theoretic development, Marchant derives conditions under which trapezoidal functions can be recovered as a representation for membership functions. The axioms reflect the properties of the trapezoidal function in the qualitative domain. As the author also suggests, these axioms require empirical testing before they are useful to the practice.

Marchant [18–20] provides new measurement-theoretic frameworks for measuring membership functions. As with all measurement-theoretic models, the axioms that are based on observable primitives require an empirical testing. Only then, we can have a better understanding of which theoretical model is more suitable. As an example, why should a subject who thinks that memberships in the fuzzy sets ‘Young’ and ‘Young and Young’ are the same think that memberships in the empty set and the set ‘Young and not Young’ are different? Fuzzy set theory still expects contributions from the empirical psychology domain to test measurement-theoretic axioms.

6.3 Elicitation Methods

There have been different sorts of elicitation methods used for membership functions. Some of these fall in the branch which tries to empirically validate the axioms of fuzzy set theory and some are constructed with a certain interpretation of the grade of membership. It is useful to consider both branches as each has its ways of constructing membership functions.

It is generally acknowledged that elicitation of membership functions is a critical step in fuzzy analysis. However, a viewpoint is that no matter how crude the elicitation method is, fuzzy analysis should be robust to imprecision in membership functions [21]. This is in tune with a qualitative analysis based on ordinal data. But most applications of fuzzy sets actually rely on cardinal properties of the membership functions or implicitly assume cardinality for which elicitation methods can be critical.

In a recent review, Verkuilen [22] identifies three general categories of constructing membership functions: direct assignment, indirect assignment and assignment, by transformation. Although this is generic enough, there are some methods which do not fall in any of these. One can identify six methods used in experiments with the aim of constructing membership functions [3, 13, 23, 24]:

1. Polling: Do you agree that John is tall? (Yes/No).
2. Direct scaling (point estimation): Classify color A according to its darkness and classify John according to his tallness. In general, the question is ‘How F is a ?’
3. Reverse scaling: Identify the person who is tall to the degree 0.6? In general, identify a who is F to the degree $\mu_F(a)$.
4. Interval estimation (set-valued statistics): Give an interval in which you think color A lies and give an interval in which you think the height of John lies.
5. Membership function exemplification: What is the degree of belonging of color A to the (fuzzy) set of dark colors? What is the degree of belonging of John to the set of tall people? In general, ‘to what degree a is F ?’
6. Pairwise comparison: Which color, A or B, is darker (and by how much)?

6.3.1 Polling

In polling one subscribes to the point of view that fuzziness arises from *interpersonal* disagreements. The question ‘Do you agree that a is F ?’ is asked to *different* individuals. The answers are polled and an average is taken to construct the membership function.

Hersh and Carmazza [25] used this approach¹ in their experiment 1. They presented their subjects a phrase (like small, very small, large, very large, etc.) and then showed 12 squares in random order. The subjects responded by ‘yes’ or ‘no,’ depending on whether they think that the phrase applies to the shown square or not. The subjects were also shown the 12 squares at the beginning of the experiment in ascending order so that every subject was operating within approximately *the same context*. Although the experimental results justify $(1 - \mu(x))$ for the connective NOT, the “squaring the membership function” is not justified for VERY. Instead of squaring, VERY seems to *shift* the membership function (a result also reported in [12, 24, 27]). They also report a reasonably good match between the membership function of ‘either small or large’ and the *maximum* of the membership functions of small and large.

Polling is also one of the natural ways of eliciting membership functions for the likelihood interpretation of graded membership.

6.3.2 Direct Scaling

Direct scaling seems to be the most straightforward way to come up with a membership function. This approach subscribes to the point of view that fuzziness arises from *individual subjective* vagueness.² The subject is required to classify *a* with respect to *F over and over again* in time. The experiment has to be carefully designed so that it will be harder for the subject to remember past answers. Harsh and Carmazza [25] use this approach in their experiment 2. Essentially, this is a repetition of their experiment 1 for a single subject repeated over days. They report similar results for negation, disjunction, and emphasizing (VERY) for each individual as in the group membership experiment. Stemming from responses of one particular subject they differentiate between *linguistic* and *logical* interpretations of membership functions.

One can also use direct scaling to *compare* the answers of a subject against a predefined membership function Turksen [13, 29]. In that case the subject is asked ‘How *F* is *a*?’ and the experimenter has a perfect knowledge of the evaluation for *F* (e.g., subject’s height is known to the experimenter but not to the subject). The same question is asked to the same subject over and over again, and the membership is constructed using the assumption of probabilistic errors and by estimating a few key parameters as is usual for this type of construction.

Chameau and Santamarina [24] also discuss this method (which they call membership exemplification; however, we reserve that term for another method). They used *several* subjects and aggregated their answers as opposed to asking a single subject same questions over and over again, as was done in the experiments of [12]. Two studies [24, 30] report that this method results in membership functions with a wider spread (more fuzzy) when compared with polling and pairwise comparison.

Thole *et al.* [31] consider the measurement of membership functions and the justification of connectives within an empirical setting based on measurement theory. They argue that since the numerical membership scale is *bounded*, the scale *has to be* an absolute scale. They mention the biases that one can have in direct elicitation methods, particularly the ‘end effect,’ which is a common problem in all bounded scales (including probability). They opt for an indirect elicitation method in the spirit of measurement theory and admit that only interval scales can be constructed with indirect methods. Then they suggest a combination of a direct scaling technique and a Thurstonian scaling method. This way they create two scales and then try to identify the relationship of the two scales. They also report that neither the minimum nor the product operators are adequate representations of conjunction. This result enables [32] to come up with a ‘compensatory AND’ operator, which is a parametric operator that covers minimum and product.

Nowakowska [33] considers scaling of membership functions. He uses direct scaling and poses the problem in terms of psychophysical scaling. He clarifies the assumptions to be made in order to perform direct scaling.

¹ See also [26] for the same approach.

² This approach is in tune with the claim that the vagueness in verbal concepts is an integral part of the concept rather than a result of summing of variable responses over individuals [28].

Dombi [34] focuses on providing a theoretical basis for membership construction, described with only a few parameters that are *meaningful*. He postulates five axioms, of which the most unnatural is the fourth one: μ is a rational function of polynomials of the following form:

$$\mu(x) = \frac{a_0x^n + a_1x^{n-1} + \cdots + a_n}{A_0x^m + A_1x^{m-1} + \cdots + a_m} \quad (m \neq 0).$$

Furthermore, he requires that a membership function be such that $(n + m)$ is minimal. With these assumptions he derives the form of the membership function and verifies his result using the empirical data obtained by [35].

Chen and Otto [36] consider constructing *continuous* membership functions from a given set of discrete points (subjective assessments). The rationale is that one can answer only a *finite* amount of questions from which a continuous membership function has to be constructed. They propose that the membership function can be constructed on an interval scale. When one requires membership functions to be continuous and convex,³ curve fitting methods might yield membership functions that are outside the unit interval and non-convex. In order to obtain continuous membership functions that are invariably bounded, convex, and continuous, Chen and Otto propose a *constrained interpolation* method.

Boucher and Gogus [37] compare direct scaling method with their proposed ‘fuzzy-spatial instrument.’ This technique constructs trapezoidal membership functions from direct assessments provided by subjects. They also provide an experimental design to test how well the proposed elicitation method performs. They conclude that one should be aware of the imprecision introduced into the model by the elicitation technique which can be in significant amounts.

Gediga *et al.* [38] report their experimental work on eliciting ‘aggregated’ fuzzy information. In doing so, they reanalyze the results of [32]. They report that the ‘compensatory and’ (and probably any t -norm) need not be the best way of measuring the aggregated fuzzy information. Humans use a way of ‘rescaling’ which is consistent with conditionalization efforts.

6.3.3 Reverse Scaling

In this method, the subject is given a membership degree and then asked to identify the object for which that degree corresponds to the fuzzy term in question [13, 29]. This method can be used for individuals by repeating the same question for the same membership function as well as for a group of individuals.

Once the subject’s (or subjects’) responses are recorded the conditional distributions can be taken to be normally distributed and the unknown parameters (mean and variance) can be estimated as usual. This method also requires evaluations to be made on at least interval scales.

Chameau and Santamarina [24] consider reverse scaling as a valuable tool to *verify* the membership function obtained by using another approach rather than an acquisition method.

6.3.4 Interval Estimation

Interval estimation subscribes to the *random set* view of the membership function. The subject is asked to give an *interval* that describes the *Fness* of a . Let I_i be the set-valued observation (the interval) and m_i the frequency with which I_i is observed. Then $R = (I_i, m_i)$ defines a random set [39, 40]. Notice that this method is more appropriate to situations where there is a clear linear ordering in the measurement of the fuzzy concept like in tallness, heat, time, etc.

Chameau and Santamarina [24] find this approach of elicitation particularly advantageous over polling and direct scaling in which the answer mode is necessarily crisp (yes/no). Interval estimation is a relatively

³ A membership function, μ , (over the real numbers) is called *convex* if and only if for all $x, y \in \mathbb{R}$ and for all $\lambda \in [0, 1]$, $\mu(\lambda x + (1 - \lambda)y) \geq \min\{\mu(x), \mu(y)\}$.

simple way of acquiring the membership function and it results in membership functions that are ‘less fuzzy’ (the spread is narrower) when compared with direct scaling and polling.

Interval estimation subscribes to the *uncertainty* view of membership functions as opposed to the *vagueness* view and in that sense it brings the issues of uncertainty modeling using fuzzy set theory, random sets, possibility measures, and their relations to probability theory [41]. [42] and [43] describe statistical methods to come up with the membership function while subscribing to the random set interpretation.

Zwick [44] also considers the random set interpretation and uses the law of comparative judgments to assess the membership function.

Recently, set-valued statistics has been proposed as a more ‘natural’ way of handling data than point-valued statistics. The set-valued statistics methods also assume a random set or likelihood interpretation of the membership function. Such methods are analyzed in [45] in some detail.

Li and Yen [46] discuss various methods of elicitation methods. Their main assumption states fuzziness as a form of uncertainty. As a result of this view the first three methods they propose are suitable for the random set view of the fuzzy sets. They ask their subjects for intervals which include the fuzzy concept at hand. They do not differentiate between single-subject versus multiple-subject fuzzy concepts. And they do not discuss the issue of commensurability between (or lack of) two measurements they obtain for different agents and different fuzzy concepts.

6.3.5 Membership Exemplification

In terms of membership function exemplification, Hersh and Carmazza [25] performed a test for the direct elicitation of the membership function. Hersh and Carmazza ordered 12 squares in ascending order and indicated each square with an ordinal number. They asked the subjects to write the number(s) which is appropriate for ‘large,’ ‘very large,’ ‘small’ etc. The results are at variance with direct scaling and polling most likely because there is no *repetition* in this elicitation method to normalize the effects of error or ‘noise.’

Kochen and Badre [47] also report experimental results on exemplification (they call it *anchoring*), which makes the resulting membership functions more precise (than without exemplification).

Zysno [27] uses exemplification in an empirical setting. He asks 64 subjects from 21 to 25 years of age to rate 52 different statements of age with respect to one of the four sets: very young man, young man, old man, and very old man. He utilizes a scale from 0 to 100 to collect the answers. Since, at the outset, he has some hypothesis on the nature of the membership function, he mainly tries to test those hypotheses.

Kulka and Novak [48] test for whether people use min–max or algebraic product and sum when combining fuzzy concepts. They use the method of exemplification (using ellipses and rectangles) in their experiments. Their conclusions are weak and they call for more experiments to come up with stronger conclusions.

However, the use of computer graphics to give an example membership function to be modified by the subject greatly enhanced this procedure as is usually witnessed in commercial applications of ‘fuzzy expert system shells.’

6.3.6 Pairwise Comparison

Kochen and Badre [47] report experimental results for the ‘precision’ of membership functions using pairwise comparison method. They report experimental evidence for the precision of *greater*, *very much greater*, and *much greater* in decreasing precision in that order. How the addition of *very* makes the adjective more precise is highlighted.

Oden [49] discusses the use of fuzzy set theory in psycholinguistic theories. He considers *comparisons* of the form ‘which is a better example of a bird: an eagle or a pelican?’ and after the answer to this question (say, an eagle is chosen), ‘how *much more* of a bird is an eagle than a pelican?’ But, by asking for the

strength of the preference directly, Oden falls for a pitfall for which many researchers have been cautioning us for a long time (see, e.g., [50], p. 32, Fallacy 3).

Chameau and Santamarina [24] also use the same pairwise comparison technique and report it to be as robust as polling and direct scaling. Following [51], they require the subjects to provide pairwise comparisons and the *strength* of preference. This yields a non-symmetric full matrix of relative weights. The membership function is found by taking the components of the eigenvector corresponding to the maximum eigenvalue. The values are also *normalized*. Chameau and Santamarina also find the requirement that evaluations on a ratio scale to be unnatural.

However, they espouse a ‘comparison-based point estimation,’ which determines the position of a set of stimuli on the reference axis by pairwise comparison and the membership is calculated by *aggregating* the values provided by several subjects. Although the subjects of Chameau and Santamarina experiments ranked this method almost as good as the interval estimation method (which was ranked as the best method), this method also needs the unfortunate assumption of a ratio scale. Furthermore, pairwise comparison requires many comparison experiments in a relatively simple domain.

6.3.7 Learning Techniques

Advances in machine learning has its implication in fuzzy set theory as well. Learning membership, fuzzy rules, a structure, and a fuzzy system all introduce some technique of eliciting a membership function. Learning techniques are always based on certain assumptions (e.g., generalized implication, a specific neurofuzzy construction, etc.). Nevertheless, these studies sometimes provide creative and practical way of eliciting membership functions. These techniques are also (seemingly) more ‘objective’ ways of constructing membership functions. However, the underlying assumptions and the inherent models at the background might still reflect the subjective nature of fuzzy sets.

Fuzzy clustering techniques [52–54] are appropriate tools for such an analysis. Usually the analysis proceeds as follows [55]:

- Apply clustering on the output data and then project it into the input data, generate clusters, and select the variables associated with input–output relations. The clustering method determines clusters, on the data space based on Euclidean norm.
- Form the membership functions for the variables selected (i.e., determine the shape of the membership functions). There is a procedure [55] that lets one to select four parameters which completely characterizes trapezoidal membership functions.⁴
- Select the input variables by dividing the data into three subgroups. Use two groups in the model building for selection of effective (important) variables and cross-validation for data set independence. The third group is used as the test data to validate the goodness of the model.

Since most fuzzy clustering techniques are based on Euclidean norm, formation of fuzzy clusters and hence membership functions with non-Euclidean norms requires further investigation.

Takagi and Hayashi [57] discuss a neural network that generates non-linear, *multidimensional* membership function, which is a membership function generating module of a larger system that utilizes fuzzy logic. They claim that the advantage of using non-linear, multidimensional membership functions is in its effects in reducing the number of fuzzy rules in the rule base.

Yamakawa and Furukawa [58] present an algorithm for learning membership functions using a model of the *fuzzy neuron*. Their method uses example-based learning and optimization of cross-detecting lines.

⁴ There has been a recent interest in using trapezoidal membership functions in fuzzy set theory mainly because their special structure yields more efficient computations. By definition, triangular membership functions are special cases of trapezoidal membership functions. However, trapezoidal functions are not as general as continuous spline functions. See, e.g., [56] for a motivation to use triangular membership functions in fuzzy control.

They assign *trapezoidal* membership functions and automatically come up with its parameters. The context is handwriting recognition. They also report some computational results for their algorithm.

On the experimental side, Erickson *et al.* [59] claim that membership functions and fuzzy set theory *better explain* the classification of taste responses in brain stem. They analyze previously published data and allow each neuron to belong to *several classifications to a degree*. This degree is measured by the neuron's response to the stimuli. They show that their model based on fuzzy set theory explains the data better than other statistical models.

Furukawa and Yamakawa [60] describe two algorithms that yield membership functions for a fuzzy neuron and their application to recognition of handwriting. The crossing points of two (trapezoidal) membership functions are *optimized* for the task at hand.

Dubois *et al.* [61] use a rule-learning strategy on a relational database based on α -cut decompositions of fuzzy sets. The result is a data-mining technique that discovers both the presence and the type of an association. The side effect is a membership function that is learned from data.

6.4 General Remarks and Summary

In general, [24, 62] report good agreement between direct scaling, interval estimation, and membership exemplification with the comment that in most of the cases, fuzzy sets obtained by exemplification method are wider (fuzzier) than the ones obtained by other methods. The main difficulty with the point estimation method is the contradiction between fuzziness of the perception and the crispness of the response mode. This difficulty is overcome by the interval estimation method, which in turn needs a minimum number of assessors or assessments. However, Chameau and Santamarina [24] report that as low as five assessments are sufficient. Exemplification yields membership functions without further processing, which is an advantage. The way they carry out the pairwise comparison method assumes a ratio scale for the measurements, which is hardly justified.

The assessors that took part in the experiments of Chameau and Santamarina (subjectively) rated the interval estimation method as the best in terms of expected consistency and expected quality. The age of the assessors affected their response, particularly in the 'old-not old' task.

One important issue in constructing membership functions is the *context*. It has always been emphasized that fuzzy set theory and particularly the membership functions are context dependent. Hersh *et al.* [63] discuss the effects of *changing context* in determining the membership functions, a problem which eluded linguists for a long time [64]. They report that the frequency of occurrence of the elements does not effect the location and form of the membership function (i.e., if one asks 'Is John tall?' over and over again, the resulting membership is consistently of the same form). On the other hand, the number of unique elements significantly affects the form of the membership function (i.e., the answer to the question 'Is John tall?' varies when there is only one more person to consider versus when there is more than one more person to consider. The context changes!). This observation tends to suggest that the membership function should have as its domain not only the universe of discourse but the discourse as well.

Membership functions are critical to fuzzy set theory. In this review we summarized recent advances in measurement-theoretic developments and tried to give a more comprehensive view of elicitation methods.

One can prove various representation and uniqueness results for membership functions using measurement theory. It appears that that branch of the literature is in need of empirical validation of its axioms which has been mostly neglected.

Elicitation methods differ by their view of fuzziness. Fuzziness can be subjective or objective and it can arise from individual assessments or the assessments of a group. Various elicitation methods are available with explicit or implicit assumptions about the underlying fuzziness. Probably the most important criticism against any fuzzy set based theory is: where do numbers come from? Measurement theory provides a framework in which many different elicitation methods can be evaluated appropriately.

Furthermore, *automatic* elicitation methods based on learning techniques are also available. These techniques respond to a practical need of eliciting membership functions from a set of given data. Many authors consider these methods as more 'objective' than other elicitation methods. However, one should

not forget that implicit assumptions about ‘discovery techniques’ or the underlying fuzzy model can still capture the inherent subjective nature of the process.

Once the scale of measurement is determined and if it allows transformations on a cardinal scale, dilation (e.g., more or less) or concentration (e.g., very) transformations can be defined. As far as granular computing is concerned, higher level abstractions should rely on the scale properties of the lower levels. Care should be taken when the lower level measurements are on an ordinal scale as abstractions on such measurements via transformations may not be meaningful and might lead to unexpected results.

References

- [1] L.A. Zadeh. Toward a generalized theory of uncertainty (GTU) an outline. *Inf. Sci.* 172 (1–2)(2005), 1–40.
- [2] L.A. Zadeh. Fuzzy sets. *Inf. Control* 8 (1965) 338–353.
- [3] T. Bilgiç and İ.B. Türkşen. Measurement of membership functions: theoretical and empirical work. In: D. Dubois and H. Prade (eds), *Handbook of Fuzzy Sets and Systems: Fundamentals of Fuzzy Sets*, Vol. 1. Kluwer Academic Publishers, Dordrecht 1999, chapter 3, pp. 195–232.
- [4] D.H. Krantz, R.D. Luce, P. Suppes, and A. Tversky. *Foundations of Measurement*, Vol. 1. Academic Press, San Diego, 1971.
- [5] F.S. Roberts. *Measurement Theory*. Addison Wesley, Reading, MA, 1979.
- [6] L. Narens. *Abstract Measurement Theory*. MIT Press, Cambridge, MA, 1985.
- [7] P. Suppes, D.H. Krantz, R.D. Luce, and A. Tversky. *Foundations of Measurement*, Vol. 2. Academic Press, San Diego, CA, 1989.
- [8] R.D. Luce, D.H. Krantz, P. Suppes, and A. Tversky. *Foundations of Measurement*, Vol. 3. Academic Press, San Diego, CA, 1990.
- [9] P. Bollmann-Sdorra, S.K.M. Wong, and Y.Y. Yao. A measurement-theoretic axiomatization of fuzzy sets. *Fuzzy Sets Syst.* 60(3) (1993) 295–307.
- [10] R.R. Yager. A measurement–informational discussion of fuzzy union and intersection. *Int. J. Man–Mach. Stud.* 11 (1979) 189–200.
- [11] A.M. Norwich and I.B. Türkşen. The fundamental measurement of fuzziness. In: R.R. Yager (ed), *Fuzzy Sets and Possibility Theory: Recent Developments*. Pergamon Press, New York, 1982, pp. 49–60.
- [12] A.M. Norwich and I.B. Türkşen. A model for the measurement of membership and the consequences of its empirical implementation. *Fuzzy Sets Syst.* 12 (1984) 1–25.
- [13] I.B. Türkşen. Measurement of membership functions and their assessment. *Fuzzy Sets Syst.* 40 (1991) 5–38.
- [14] T. Bilgiç and I.B. Türkşen. Measurement-theoretic justification of fuzzy set connectives. *Fuzzy Sets Syst.* 76(3) (1995) 289–308.
- [15] T. Bilgiç and I.B. Türkşen. Measurement-theoretic frameworks for fuzzy set theory. In: T.P. Martin and A.L. Ralescu (eds), *Fuzzy Logic in Artificial Intelligence: Towards Intelligent Systems, volume 1188 of Lecture Notes in Artificial Intelligence*. Springer, 1997, pp. 252–265. Selected papers from IJCAI ’95 Worksop Montréal, Canada, August 1995.
- [16] I.B. Türkşen. *An Ontological and Epistemological Perspective of Fuzzy Set Theory*. Elsevier, The Netherland, 2006.
- [17] T. Bilgiç and I.B. Türkşen. Measurement-theoretic frameworks for fuzzy set theory. In: *Working notes of the IJCAI-95 workshop on Fuzzy Logic in Artificial Intelligence*, 14th International Joint Conference on Artificial Intelligence Montréal, Canada, 1995, pp. 55–65.
- [18] T. Marchant. The measurement of membership by comparisons. *Fuzzy Sets Syst.* 148 (2004) 157–177.
- [19] T. Marchant. The measurement of membership by subjective ratio estimation. *Fuzzy Sets Syst.* 148 (2004) 179–199.
- [20] T. Marchant. A measurement-theoretic axiomatization of trapezoidal membership functions. *IEEE Trans. Fuzzy Syst.* 15(2) (2007) 238–242.
- [21] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*. Wiley, New York, 1994.
- [22] J. Verkuilen. Assigning membership in a fuzzy set analysis. *Sociol. Methods Res.* 33 (2005) 462–496.
- [23] A.M. Norwich and I.B. Türkşen. The construction of membership functions. In: R.R. Yager (ed), *Fuzzy Sets and Possibility Theory: Recent Developments*. Pergamon Press, New York, 1982, pp. 61–67.
- [24] J.L. Chameau and J.C. Santamarina. Membership functions part I: comparing method of measurement. *Int. J. Approx. Reason.* 1 (1987) 287–301.
- [25] H. Hersh and A. Carmazza. A fuzzy set approach to modifiers and vagueness in natural language. *J. Exp. Psychol. Gen.* 105(3) (1976) 254–276.

- [26] W. Labov. The boundaries of words and their meanings. In: C.J. Bailey and R.W. Shuy (eds), *New Ways of Analyzing Variation in English*. Georgetown University Press, Washington, 1973.
- [27] P. Zysno. Modeling membership functions. In: B.B. Rieger (ed.), *Empirical Semantics I*, volume 1 of *Quantitative Semantics, Vol. 12*. Studienverlag Brockmeyer, Bochum, 1981, pp. 350–375.
- [28] M.E. McCloskey and S. Glucksberg. Natural categories: well defined or fuzzy sets? *Mem. Cogn.* 6 (1978) 462–472.
- [29] I.B. Türkşen. Stochastic fuzzy sets: a survey. In: *Combining Fuzzy Imprecision with Probabilistic Uncertainty in Decision Making*. Springer-Verlag, New York, 1988, pp. 168–183.
- [30] A.M. Norwich and I.B. Türkşen. Stochastic fuzziness. In: M.M. Gupta and E. Sanchez (eds), *Approximate Reasoning in Decision Analysis*. North-Holland, Amsterdam, 1982, pp. 13–22.
- [31] U. Thole, H.J. Zimmermann, and P. Zysno. On the suitability of minimum and product operators for the interpretation of fuzzy sets. *Fuzzy Sets Syst.* 2 (1979) 167–180.
- [32] H.-J. Zimmermann and P. Zysno. Latent connectives in human decision making. *Fuzzy Sets Syst.* 4 (1980) 37–51.
- [33] M. Nowakowska. Methodological problems of measurement of fuzzy concepts in the social sciences. *Behav. Sci.* 22 (1977) 107–115.
- [34] J. Dombi. Membership function as an evaluation. *Fuzzy Sets Syst.* 35 (1990) 1–22.
- [35] H.J. Zimmermann, and P. Zysno. Quantifying vagueness in decision models. *Eur. J. Oper. Res.* 22 (1985) 148–154.
- [36] J.E. Chen and K.N. Otto. Constructing membership functions using interpolation and measurement theory. *Fuzzy Sets Syst.* 73 (1995) 313–327.
- [37] T.O. Boucher and O. Gogus. Reliability, validity, and imprecision in fuzzy multicriteria decision-making. *IEEE Trans. Syst. Man Cybern. Part C* 32(3) (2002) 190–202.
- [38] G. Gediga, I. Duntisch, and J. Adams-Webber. On the direct scaling approach of eliciting aggregated fuzzy information: the psychophysical view. In: *Proceedings of NAFIPS '04*, Vol. 2, IEEE, 2004, pp. 948–953.
- [39] D. Dubois and H. Prade. Fuzzy sets, probability and measurement. *Eur. J. Oper. Res.* 40 (1989) 135–154.
- [40] D. Dubois and H. Prade. Random sets and fuzzy interval analysis. *Fuzzy Sets Syst.* 42 (1991) 87–101.
- [41] D. Dubois and H. Prade. Fuzzy sets and probability: misunderstandings, bridges and gaps. In: *Second IEEE International Conference on Fuzzy Systems*, San Francisco, CA, March 28–April 1, 1993, IEEE, 1993, pp. 1059–1068.
- [42] D. Dubois and H. Prade. Fuzzy sets and statistical data. *Eur. J. Oper. Res.* 25 (1986) 345–356.
- [43] M.R. Civanlar and H.J. Trussel. Constructing membership functions using statistical data. *Fuzzy Sets Syst.* 18 (1986) 1–14.
- [44] R. Zwick. A note on random sets and the Thurstonian scaling methods. *Fuzzy Sets Syst.* 21 (1987) 351–356.
- [45] R. Kruse and K.D. Meyer. *Statistics with Vague data. Theory and Decision Library: Series B Mathematical and Statistical Methods*. D. Reidel Publishing Company, Dordrecht/Holland, 1987.
- [46] H.X. Li and V.C. Yen. *Fuzzy Sets and Fuzzy Decision-Making*. CRC Press, Boca Raton, FL, 1995.
- [47] M. Kochen, and A.N. Badre. On the precision of adjectives which denote fuzzy sets. *J. Cybern.* 4 (1974) 49–59.
- [48] J. Kukla and V. Novak. Have fuzzy operators a psychologocal correspondence? *Stud. Psychol.* 26 (1984) 131–140.
- [49] G.C. Oden. Fuzzy propositional approach to psycholinguistic problems: an application of fuzzy set theory in cognitive science. In: M.M. Gupta, R.K. Ragade, and R.R. Yager (eds), *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, 1979, pp. 409–420.
- [50] R.D. Luce and H. Raiffa. *Games and Decisions*. Wiley, New York, 1957.
- [51] T.L. Saaty. Measuring the fuzziness of sets. *J. Cybern.* 4 (1974) 53–61.
- [52] E.H. Ruspini. A new approach to clustering. *Inf. Control* 15(1) (1969) 22–32.
- [53] J.C. Bezdek and J.D. Harris. Convex decompositions of fuzzy partitions. *J. Math. Anal. Appl.* 6 (1979) 490–512.
- [54] M. Sugeno and T. Yasukawa. A fuzzy logic based approach to qualitative modelling. *IEEE Trans. Fuzzy Syst.* 1(1) (1993) 1–24.
- [55] H. Nakanishi, I.B. Turksen, and M. Sugeno. A review and comparison of six reasoning methods. *Fuzzy Sets Syst.* 57 (1993) 257–294.
- [56] W. Pedrycz. Why triangular membership functions? *Fuzzy Sets Syst.* 64(1) (1994) 21–30.
- [57] H. Takagi and I. Hayashi. Nn-driven fuzzy reasoning. *Int. J. Approx. Reason.* 5(3) (1991) 191–213. (Special issue of IIZUKA'88).
- [58] T. Yamakawa and M. Furukawa. A design algorithm of membership functions for a fuzzy neuron using example-based learning. In: *Proceedings of the First IEEE Conference on Fuzzy Systems*, San Diego, CA, 1992, pp. 75–82.
- [59] R.P. Erickson, P.M. Di Lorenzo, and M.A. Woodbury. Classification of taste responses in brain stem: membership in fuzzy sets. *J. Neurophysiol.* 71(6) (1994) 2139–2150.

- [60] M. Furukawa and T. Yamakawa. The design algorithms of membership functions for a fuzzy neuron. *Fuzzy Sets Syst.* 71(3) (1995) 329–343.
- [61] D. Dubois, H. Prade, and T. Sudkamp. On the representation, measurement, and discovery of fuzzy associations. *IEEE Trans Fuzzy Syst.* 13(2) (2005) 250–262.
- [62] J.L. Chameau and J.C. Santamarina. Membership functions part II: trends in fuzziness and implications. *Int. J. Approx. Reason.* 1 (1987) 303–317.
- [63] H. Hersh, A. Carmazza, and H.H. Brownell. Effects of context on fuzzy membership functions. In M.M. Gupta, R.M. Ragade, and R.R. Yager (eds), *Advances in Fuzzy Set Theory*. North-Holland, Amsterdam, 1979, pp. 389–408.
- [64] J.A.W. Kamp. Two theories about adjectives. In: E.L. Keenan (ed), *Formal Semantics of Natural Language*. Cambridge University Press, London, 1975, pp. 123–155.

7

Fuzzy Clustering as a Data-Driven Development Environment for Information Granules

Paulo Fazendeiro and José Valente de Oliveira

7.1 Introduction

The capability to deal with uncertainty and imprecision is a defining feature of the human being. We are used to think, make decisions, and interact with the external world on the basis of vague or abstract knowledge built over imprecise and incomplete data. Such manner of perceiving the world has been an essential asset to our survival and thus has been developed and improved generation after generation.

However, nowadays, the advent of massive databases and the treatment of their immense volumes of data poses several challenges regarding knowledge discovering and representation. On the one hand, the huge amount of available data obliges, to a greater or lesser extent, to automatic computer manipulation. On the other hand, computers are inherently precise machines and need guidance in order to extract knowledge and represent it in a human-like fashion. Information granules emerge at the conceptual level as a powerful means to accomplish knowledge representation, information processing, and structured thinking and problem solving [1]. The materialization of the information granulation in the framework of fuzzy sets, as proposed by Zadeh [2, 3], presents appealing features regarding the human understandability of the produced granules. The elegant concept of generalized constraint built over the fuzzy generalization of the classical set promotes the necessary flexibility to represent vague and unprecise linguistic terms and also detains sufficient tractability in order to permit computations on them. Fuzzy clustering can be conceived as a privileged collection of techniques to search for structure in data and present it as fuzzy information granules. Furthermore, the linguistic granules can be used as the point of departure for further meaningful refinements and support the modularization of the data set, reducing the computing power that would be necessary to reveal detailed relationships at a numeric level (cf. [4]).

Granular computing (GrC) is a general computation theory based on the commonsense concepts of information granule, granulated view, granularity, and hierarchy aiming at their effective use in order to build an efficient computational model for complex applications with huge amounts of data, information, and knowledge [5, 6]. Some authors (e.g. [5]) emphasize the combination of two different aspects of this pursuit: the algorithmic abstraction of data and the non algorithmic, empirical verification of these abstractions. In this perspective, when incorporated in the conceptual framework of GrC, the methodologies of fuzzy clustering can provide a valuable surplus reducing the gap between those two facets

of GrC. First of all the identification of regions of interest of a data set can be transposed quite easily to propositions on meaningful linguistic labels, thus facilitating the empirical semantic validation of the model. Moreover if one is interested in studying some of these regions of interest in a detailed manner, thus creating an hierarchy of multilevel abstractions, the fuzzy granule may define the searching context (e.g., through the technique described in [7]) and the (same) linguistic labels could be successively redefined at the new subspaces [5]. Another way to take advantage of the semantics of fuzzy sets consists in constructing hierarchies of concepts through sound operators of generalization and specialization and linguistic modifiers. Two possibilities are immediately envisioned: on the one hand, we may start with a clustering output very specific (abstraction with a high granularity level) and gradually define new granularity levels through successive generalizations; on the other hand, we can combine very two distinct clustering abstractions, identifying a different number of clusters, as a way to obtain intermediate levels of abstraction.

The chapter is organized as follows. In the following section we review the fundamental concepts of hard and fuzzy clustering, the major approaches to fuzzy clustering, as well as their limitations and strengths. In Section 7.3 we describe the process of developing fuzzy information granules from data and explain how fuzzy clustering can be used in order to comply with the granulation concerns. Section 7.4 draws some relevant conclusions and ends the chapter.

7.2 Fuzzy Clustering Essentials

Generally speaking, clustering is the process of searching for a finite and discrete set of data structures (categories or clusters) within a finite, otherwise unlabeled, usually multivariate data set. In the literature it is common to find that the goal of clustering is the partition of the data set into groups so that data in one group are similar to each other and are as different as possible from data in other groups, (cf. [8, 9]). Two distinct, but complementary, facets are enclosed in this unsupervised learning task: the elicitation of a model of the overall structure of the data and the pursuit for a manageable representation of a collection of objects into homogeneous groups.

A concept of paramount importance, with direct implications on the clustering endeavor, is the notion of similarity between elements (patterns) of the data set. Usually the similarity is defined at the expense of its dual relation, dissimilarity. Dissimilarity is a dyadic relation, non-negatively defined, symmetric on its two arguments, and attains its minimum (zero) when the two arguments are identical. If this relation also satisfies the triangular inequality property (subadditivity), it is called distance function or metric. For continuous features, each metric induces a particular topology on the data set and consequently a different view of the data (a different geometry of the clusters). In cluster analysis some common choices for distance functions include the Hamming (city block) distance inducing diamond-shaped clusters, the Euclidean distance inducing (hyper)spherical clusters, and the Tchebyshев distance inducing (hyper) box-shaped clusters. As a matter of fact these examples are members of the Minkowski family of distances, or L_p norms, defined as

$$D(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}. \quad (1)$$

In order to hold the triangle inequality, in the above expression $p \in \mathbb{R}$ cannot be less than 1. Notice that this family of distances comprises the three examples above, respectively, for values of p equal to 1, 2, and $+\infty$. Distance can be used to measure the similarity between two data points or between a data point and a prototype of the cluster. The prototype is a mathematical object, usually a point in the feature space (e.g., the center of the cluster) or even a geometric subspace or function, acting as a representative of the cluster while trying to capture the structure (distribution) of its associated data.

7.2.1 Some Taxonomy Remarks

Traditionally, the clustering algorithms are categorized in two main types: hierarchical and objective function-based partitional clustering. Every new cluster determined by a hierarchical algorithm is based

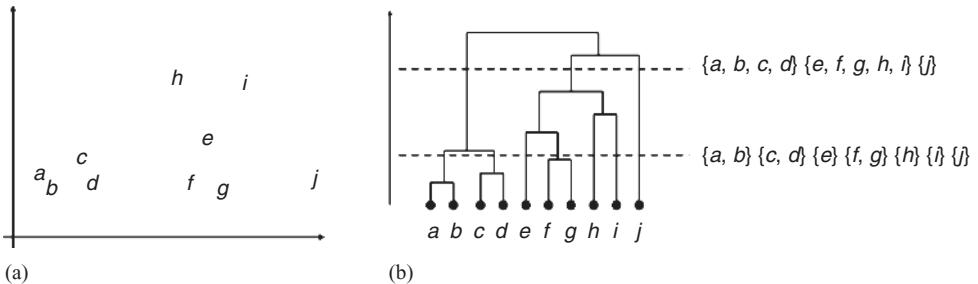


Figure 7.1 Hierarchical agglomerative clustering: (a) synthetic two-dimensional data set; (b) example of a dendrogram with two distinct distance cuts and resulting clusters

on the set of previously established clusters. Typically, this can be done in two distinct ways: the agglomerative ('bottom-up') and more common approach begins with each element as a single-element cluster and successively merges the closest clusters into larger clusters, whereas the divisive ('top-down') approach begins with the whole set and splits it into successively smaller clusters. The process is repeated until a stopping criterion (e.g., the predefined distance threshold or, more frequently, the desired number of clusters) is achieved. The result of the algorithm usually consists in a hierarchical data representation as a tree of clusters (a dendrogram), where every cluster node contains child clusters and the individual elements are at the leafs. By cutting the dendrogram at different levels, different clustering of the data items into disjoint groups is obtained (see Figure 7.1).

The distance between individual points has to be generalized to the distance between subsets (linkage metric) in order to merge (or split) clusters instead of individual points. This type of the linkage metric significantly affects hierarchical algorithms, since each cluster may contain many data points and present different geometrical shapes, sizes, and densities. Some common choices include the computation of the distance between the closest elements of the different clusters (single link method [10], as illustrated in Figure 7.1), the computation of the maximal distance between the elements belonging to different clusters (complete link method [11]), or some sort of average of the dissimilarity between all pairs of elements of the distinct clusters (average link, e.g., [12]). The distance is computed for every pair of points with one point in the first set and another point in the second set. Because of the pairwise combinatorial nature of the process the hierarchical approach tends to be computationally inefficient with the growth of the number of data elements. This approach is very sensitive to anomalous data points (noise and outliers) and is unable to handle overlapping clusters. A reason for this is that bad decisions made at an early stage of the algorithm will be propagated and amplified up to the end since the intermediate clusters are not revisited for further improvement. (The points cannot move from one cluster to another.)

The second major category of clustering algorithms attempts to directly decompose the data set into a collection of disjoint clusters. This partition is builded during an iterative optimization process repeated until its associated cost function reaches a minimum (global or local). The cost function, also designed performance index or objective function, is a mathematical criterion expressing some desired features (emphasizing local or global structure of the data) of the resulting partition.

Theoretically, the determination of the optimal partition is viable by exhaustive enumeration; however, the NP-complete nature of the problem makes it infeasible in practice [13]. The Stirling number of the second kind expresses the number of ways to assign a set of N objects to C non-empty clusters:

$$S(N, C) = \frac{1}{C!} \sum_{i=1}^C (-1)^{C-i} \binom{C}{i} i^N. \quad (2)$$

Even assuming that we are interested in forming a constant number of clusters, this number increases very quickly with the number of objects. (For instance if we consider two sets of 4 and 9 patterns distributed over 3 clusters, we get $S(4, 3) = 6$, whereas $S(9, 3) = 3025$). Thus, there are no known polynomial time algorithms that provide the optimal solution. However, combining some heuristics with

an adequate formulation of the objective function it is possible to design an optimization process which is able to determine at least suboptimal partitions. One such formulation, for that matter the most used in practice, is the sum-of-squared-error distances or minimum variance criterion representing each of C clusters by their mean (the so-called centroid $\mathbf{v}_i \in \mathbb{R}^n$, $i = 1, \dots, C$, of its points):

$$Q = \sum_{i=1}^C \sum_{j=1}^N u_{ij} D_{ji}^2(\mathbf{x}_j, \mathbf{v}_i), \quad (3)$$

where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ denotes a set of feature vectors (or patterns) in the \mathbb{R}^n space. $D_{ji}(\mathbf{x}_j, \mathbf{v}_i)$ is a measure of the distance from \mathbf{x}_j to the i th cluster prototype. The elements $u_{ij} \in \{0, 1\}$, $i = 1, \dots, C$ and $j = 1, \dots, N$, form a matrix designated as the *partition matrix*, which maps the patterns to a cluster. If $u_{ij} = 1$, the pattern j belongs to cluster i ; otherwise if $u_{ij} = 0$, the pattern j is not accounted as a member of cluster i . This formulation is appealing because it still favors sets of well-separated clusters with small intracluster distances, while replaces all the pairwise distances computation by a single cluster representative. Thus the computation of the objective function becomes linear in N and it is now feasible the application of an iterative optimization process aiming at gradual improvements of the built clusters.

The c -means algorithm (also referred in the literature as k -means or hard c -means) [14, 15] is the best known squared error-based example of such a process. For a given initialization of the C centroids the heuristic approach consists of two-step major iterations that follow from the first-order optimality conditions of (3): first, reassign all the points to their nearest cluster, thus updating the partition matrix, and then recompute the centroids, \mathbf{v}_j (its coordinates are the arithmetic mean, separately for each dimension, over all the points in the cluster), of the newly assembled groups. This iterative procedure continues until a stopping criterion is achieved (usually until no reassessments happen). In spite of its simplicity and speed this algorithm has some major drawbacks. It is much dependent on the initial centroids assignment (frequently, in practice, it is run for a number of times with different random assignments and the best resulting partition is taken), does not ensure that the result has a global minimum of variance, is very sensitive to outliers, and lacks scalability.

Another not so obvious disadvantage is related to the binary nature of the elements of the partition matrix and consequently of the induced partitions. This kind of partition matrix is based on classical set theory, requiring that an object either does or does not belong to a cluster. The partitioning of the data into a specified number of mutually exclusive subsets is usually referred as *hard clustering*. In many situations this is not an adequate representation. Consider for instance the two-dimensional data set depicted in Figure 7.2.

Two different clusters are immediately perceived and a possible hard-partition matrix could be the one below:

$$U = \begin{pmatrix} a & b & c & d & e & f & g & h & i & j \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \mathbf{1} & \mathbf{1} \end{pmatrix}.$$

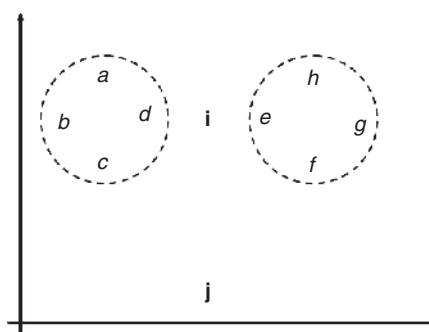


Figure 7.2 Two-dimensional data set with two cluster structures, a borderline point and an outlier

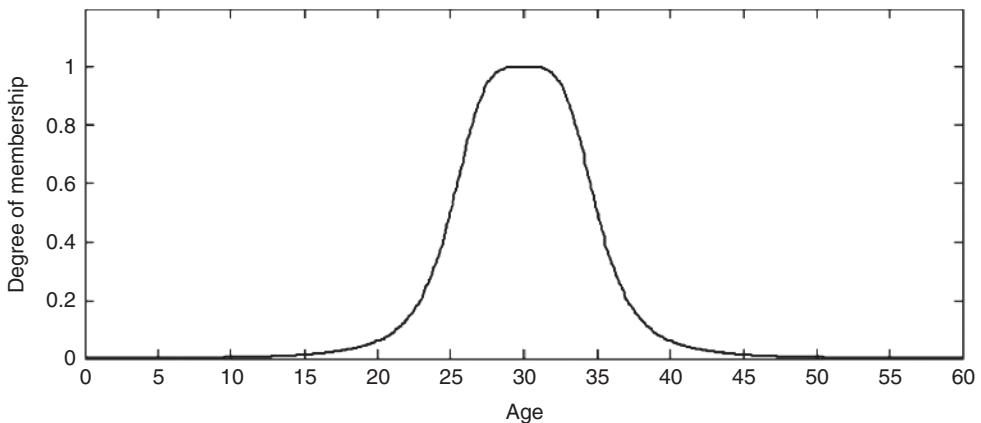


Figure 7.3 Possible materialization of the fuzzy set ‘around 30 years’

However if we take a closer look at patterns **i** and **j**, we may be tempted to question the allocation of these points to a particular cluster. As a matter of fact, **i** (a borderline point) is located in the boundary between the two clusters and we can say that **j** (an outlier) is almost as typical of the first cluster as it is of the second one. In these frequent situations a more natural partition would be one which allowed the objects to belong to several clusters simultaneously, with different degrees of membership. This is precisely the central concept behind *fuzzy clustering* methods with foundations in the fuzzy sets theory [16]. A *fuzzy set* is characterized by a *membership function* that maps each point of the universe \mathbf{X} to a number in the interval $[0, 1]$. (1 represents full degree of inclusion and 0 non-membership at all.) For instance, consider the universe $\mathbf{X} = \mathbb{R}^+$ as the set of admissible ages for a human being. A feasible membership function for the set of ages ‘around 30 years’ is graphically represented in Figure 7.3. The fuzzy set can be perceived as a more expressive generalization of the conventional set.

We can say that the relaxation of the constraint imposed on the partition matrix to $u_{ij} \in [0, 1]$ is more realistic and able to provide a richer insight of the data structure, especially when in presence of ambiguous data or clusters without sharp boundaries. Indeed, referring back to the example of Figure 7.2 a more appealing and intuitive partition matrix could be the following one, with membership degrees between 0 and 1:

$$U = \begin{pmatrix} 1 & 1 & 1 & 0.9 & 0.1 & 0 & 0 & 0 & \mathbf{0.5} & \mathbf{0.5} \\ 0 & 0 & 0 & 0.1 & 0.9 & 1 & 1 & 1 & \mathbf{0.5} & \mathbf{0.5} \end{pmatrix}.$$

Notice that patterns **i** and **j** (located at the boundaries between two classes) are no longer forced to fully belong to one of the classes, but rather exhibit equal partial membership to the different clusters. Also patterns **d** and **e** begin to show the effect of the influence of the neighboring cluster. In the remaining of this section we review some different algorithmic approaches that allow the construction of fuzzy partitions, i.e., algorithms which represent a cluster as a fuzzy set.

7.2.2 The Fuzzy C-Means Clustering Algorithm

Fuzzy clustering was introduced as early as 1969 by Ruspini [17]. Fuzzy c-means (FCM) is a simple and widely used clustering algorithm. The algorithm results from an optimization problem that consists in the minimization, with respect to \mathbf{V} , the set of prototypes, and \mathbf{U} , the fuzzy membership matrix, of the

following index (objective function) [18]:

$$Q_{FCM} = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m D_{ji}^2(\mathbf{x}_j, \mathbf{v}_i), \quad (4)$$

where $m > 1$ is the so-called fuzziness parameter ($m = 2$ is a common choice) that controls the influence of membership grades or in other words how much clusters may overlap, (cf. [19]), and D stands for a norm distance in \mathbb{R}^n , under the following conditions on the partition matrix elements:

$$u_{ij} \in [0, 1] \quad \text{for all } i = 1, \dots, C \quad \text{and } j = 1, \dots, N, \quad (5)$$

$$\sum_{i=1}^C u_{ij} = 1 \quad \text{for all } j = 1, \dots, N, \quad (6)$$

$$\sum_{j=1}^N u_{ij} > 0 \quad \text{for all } i = 1, \dots, C. \quad (7)$$

Condition (6) induces a fuzzy partition in the strict sense and assures that every datum has a similar global weight on the data set. Constraint (7) guarantees that none of the C clusters is empty, thus implying a cluster partition with no less than C clusters. Notice the similarity between (3) and (4). As a matter of fact they are coincident apart from a fixed transformation (the introduction of the fuzzifier, m) introduced as a means to prevent that under condition (6) the same minimum as the one obtained by the crisp standard formulation was reproduced.

For this constrained non-linear optimization problem there is no obvious analytical solution. Therefore the most popular and effective method to minimize the constrained objective function consists in resorting to a technique known as alternating optimization. This means that one set of parameters is kept fixed, while the other is being optimized, and next they exchange roles. The prototype \mathbf{V} and membership \mathbf{U} update equations are obtained from the necessary conditions of a minimum:

$$\frac{\partial J_{FCM}}{\partial \mathbf{V}} = 0 \quad (\text{assuming } \mathbf{U} \text{ to be constant}); \quad (8)$$

$$\frac{\partial J_{FCM}}{\partial \mathbf{U}} = 0 \quad (\text{assuming } \mathbf{V} \text{ to be constant}). \quad (9)$$

Additionally, the consideration of (6) in the original objective function (4) by means of Lagrange multipliers converts the constrained problem into its constrained-free version. Some straightforward computations lead to the update formula of the partition matrix:

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{D_{ji}(\mathbf{x}_j, \mathbf{v}_i)}{D_{jk}(\mathbf{x}_j, \mathbf{v}_k)} \right)^{\frac{2}{(m-1)}}}. \quad (10)$$

This formula does not depend on the chosen distance function; however, the determination of the prototypes is more complicated since many distance norms do not lead to a closed-type expression. A common practical choice is to use the Euclidean distance or L_2 norm (for a generalization to L_p , $p > 0$, the interested reader is referred to [20]) leading to the following prototype update equation:

$$\mathbf{v}_i = \frac{\sum_{j=1}^N u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^N u_{ij}^m}. \quad (11)$$

The alternate optimization of \mathbf{U} and \mathbf{V} proceed iteratively until no significant change of the objective function is registered. It has been proved that the generated sequence of solutions for fixed $m > 1$ always converge to local minima or saddle points of (4) [21]. Informally, what the resulting algorithm will do is

to search for the clusters that minimize the sum of the intracluster distances. In general, the performance of fuzzy algorithms, when compared with the corresponding hard-partitioning ones, is superior and they are less prone to be trapped in local minima [18]. However, like its hard counterpart the FCM algorithm shares the problem of high sensitivity to noise and outliers, something that is common to the generality of the least-squares approaches and that can drastically distort the optimal solution or facilitate the creation of additional local minima. Next, we discuss an alternative formulation, specifically designed to tackle this problem.

7.2.3 The Possibilistic C-Means Clustering Algorithm

The influence of noise points can be reduced if the memberships associated with them are small in all clusters. However, as can be seen from the probabilistic-like constraint (6), the memberships generated by the FCM are relative numbers expressing the concept of sharing of each pattern between clusters rather than the concept of typicality of a given pattern to a given cluster. This means that noise points and outliers will also have significantly high membership values. A more general form of fuzzy partition, the *possibilistic* partition, can be obtained by relaxing the constraint (6) in order to address this problem. Referring back to the example in Figure 7.2 a possible partition matrix expressing typicality could be something like this:

$$U = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0.5 & 0.1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0.5 & 0.1 \end{pmatrix}.$$

In this case the sum of the membership of the outlier \mathbf{j} no longer has to be 1, and a low membership value in both clusters implies that the effect of this pattern is now confined.

In order to assign to noise points low membership in each cluster, the normalization condition (6) must be dropped, leading to possibilistic instead of fuzzy partitions. To avoid the trivial solution (i.e., a matrix with null elements) Krishnapuram and Keller [22] added to (4) a punishment term for low memberships resulting in the augmented possibilistic c -means (PCM) objective function:

$$Q_{\text{PCM}} = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m D_{ji}^2(\mathbf{x}_j, \mathbf{v}_i) + \sum_{i=1}^C \eta_i \sum_{j=1}^N (1 - u_{ij})^m, \quad (12)$$

where the distance parameters $\eta_i > 0$ ($i = 1, \dots, C$) are specified by the user. Notice that the second term expresses the desire to have strong assignments of data to clusters. Because of the nature of the membership constraint, we call *possibilistic clustering algorithm* (PCM) to a fuzzy clustering algorithm which minimizes (12) under the constraint (7). The partition matrix update equations, as before for the FCM case, are obtained by setting the derivative of the objective function equal to zero while holding the prototype parameters fixed:

$$u_{ij} = \frac{1}{1 + \left(\frac{D_{ji}^2(\mathbf{x}_j, \mathbf{v}_i)}{\eta_i} \right)^{\frac{1}{(m-1)}}}. \quad (13)$$

This update expression clearly emphasizes the typicality interpretation of the membership function. Unlike the FCM formulation, the degree of membership of one point to a cluster depends exclusively on its distance to the center of that cluster. For the same cluster, closer points obtain higher membership than the ones farther away from it. Moreover (13) shows that η_i determines the distance of the ‘definite’ assignment ($u_{ij} > 0.5$) of a point to a cluster. (Simply considering $m = 2$ and substituting η_i by $D_{ji}^2(\mathbf{x}_j, \mathbf{v}_i)$ results in $u_{ij} = 0.5$.) So it is useful to choose each η_i separately, according to the individual geometrical features of each cluster. Unfortunately, these are not always available, so Krishnapuram and Keller recommend several methods to determine η_i [22, 23]. Using the fuzzy intracluster distance a sound

probabilistic estimation of these weight factors can be obtained:

$$\eta_i = \frac{\sum_{j=1}^N u_{ij}^m D_{ji}^2(\mathbf{x}_j, \mathbf{v}_i)}{\sum_{j=1}^N u_{ij}^m}. \quad (14)$$

The update formula for the prototypes is the same as the one used in the FCM method since the second term in (12) simply vanishes when computing the derivative of the objective function with respect to the prototype parameters. If we take a closer look at (13), we see that the membership degree of a pattern to a cluster depends only on the distance of the pattern to that cluster, and not on its distance to other clusters. So it happens that in some situations this algorithm can originate coincident clusters (converging to the same local optimal point), thus disregarding clusters with lower density or less points, or even present stability problems due to sensitivity to initialization [23]. Thus to overcome these drawbacks of the probabilistic approach it is a common practice to initialize PCM with a prior run of the probabilistic FCM.

7.2.4 Other Approaches to Fuzzy Clustering

The literature on fuzzy clustering is remarkably rich (cf. [24]), and in a broad sense it reflects the attempts made to surpass the problems and limitations of the FCM and PCM algorithms. In the two former sections we reviewed FCM and PCM and their prototypes' update equations, assuming the Euclidean distance as the standard metric. However when combined with a squared error-based objective function this distance induces hyperspherical clusters. To overcome this geometrical constraint imposed by clustering algorithms based on a fixed-distance metric several algorithms using adaptive distance measures have been proposed. Two of the most well known are the Gustafson–Kessel algorithm [25], which replaces the Euclidean distance by the Mahalanobis distance (an interesting generalization of the Euclidean distance) with a specific covariance matrix for each cluster, and the unsupervised Gath–Geva algorithm [26], where the distance is based on the fuzzification of the maximum likelihood estimation method. Both these algorithms are well fitted to find ellipsoidal clusters with varying size and orientation. (There are also axis-parallel variants of these algorithms and to some extent they can also be used to detect lines.)

In the field of image processing and recognition the geometry of the fuzzy clusters is a key aspect for image analysis tasks. Both FCM and PCM use point prototypes. If we are interested in finding particular cluster shapes, algorithms based on hyperplanar or functional prototypes, or prototypes defined by functions, are a good choice. The distance is no longer defined between two patterns (i.e., a datum and a prototype); instead, it is measured between a pattern and a more complex geometric construct. This class of algorithms includes the fuzzy c -varieties [27] for the detection of linear manifolds (lines, planes, or hyperplanes), fuzzy c -elliptotypes [28] for objects located in the interior of ellipses, fuzzy shell clustering for the recognition of object boundaries (e.g., fuzzy c -shells [29] in the detection of circles, hyperquadric shells [30], and fuzzy c -rectangular shells [31]), and fuzzy regression models [32]. The interested reader may follow a comprehensive explanation of these branch of methods in [33].

In addition to PCM other methods have been proposed in order to improve the robustness of the FCM algorithm to noisy data points and outliers while maintaining the constraint (6) (thus circumventing the problem of cluster coincidence of the PCM approach). For instance the technique presented in [34] and [35] consists in the introduction of an additional *noise cluster* aiming at grouping the points with low probability of belonging to the remaining clusters. This probability depends on the mean value of the squared distances between patterns and the prototypes of the normal clusters. Latter on, this technique was extended in order to accommodate different noise probabilities per cluster [36].

The great majority of the algorithms presented hitherto result from alternating the optimization of the membership functions and prototype locations in an iterative process. Therefore the clustering model constrains (and is constrained to) the particular shapes of the membership functions and the positions of the prototypes to those determined by the updating equations derived from the objective function. However, the user might be interested in the use of a certain type of membership function with more adequate shapes to the problem in question or in certain cluster prototypes satisfying some application-specific needs. The *alternating cluster estimation* (ACE) framework [37] is able to provide, when required, this

extra flexibility. In applications such as extraction of fuzzy rules from data, where each fuzzy set should have a clear semantic meaning (for instance, associated to linguistic labels like ‘high’ temperature or ‘about 80° degrees), a convex fuzzy set with limited support may be more preferable than the non-convex membership functions generated by FCM or PCM. Notwithstanding that ACE embodies FCM and PCM as particular instances of the framework, the requirement that the updating equations for the membership function and the prototypes should result from the necessary conditions for local extrema is now rejected and the user is free to choose the pair of updating equations which is better fitted for the problem at hand. At first sight this generalization may seem to be lacking mathematical soundness; however, it has proved its usefulness in practical examples.

In many practical applications the data sets can be heavily contaminated by noise points which promote the proliferation of local minima. In these cases, the probability of the alternate optimization getting stuck at local optimal values is far from being negligible. To obviate this problem, stochastic algorithms have been used in cluster analysis, many of them inspired on biological paradigms, such as the natural evolution of species or swarm-based behavior. Examples of such approaches to fuzzy clustering include the use of genetic algorithms [38–43], evolutionary programming [44], evolutionary strategies [45], ant colony optimization [46], and particle swarm optimization [47]. Notwithstanding that these attempts do not guarantee optimal solutions, demand the definition of a set of problem-specific parameters (e.g., population size) and are very computationally time consuming, they can undoubtedly contribute to avoid local extrema and reduce the sensitivity to initialization.

As concluding remark of this section we would like to quote [37] in a sentence which applies to every clustering endeavor:

The quality of clusters [...] is, as always, a function of parametric choices, user skill and determination, and most importantly, the data being clustered.

7.2.5 Determination of the Number of Fuzzy Partitions

In the great generality of the partitional algorithms the number of clusters C is the parameter having greater influence on the resulting partition. The chosen clustering algorithm searches for C clusters, regardless of whether they are really present in the data or not. So when there is no prior knowledge about the structure of the data a natural question arises: what is the right number of clusters for a particular data set? This question is known in the literature as the cluster validity problem and distinct validity measures have been proposed in order to find an answer (cf. [9, 8, 48–51]). However, in spite of a greater practical adhesion to some of them, due to the subjective and application-dependent character of the problem there is no consensus on their capability to provide a definitive answer to the foregoing question. For partitional fuzzy clustering it is advisable that the validity indices account for both the data set (e.g., their variance) and the resulting membership degrees. An example of such class of validity indices, exhibiting good behavior when matched against a set of other indices [52], is the Xie–Beni index [53], also known as the compactness and separation index, computed as the ratio of the compactness of the fuzzy partition of a data set to its separation:

$$\text{XB} = \frac{\sum_{i=1}^C \sum_{j=1}^N u_{ij}^m D_{ji}^2(\mathbf{x}_j, \mathbf{v}_i)}{N \min_{i \neq j} D_{ij}^2(\mathbf{v}_i, \mathbf{v}_j)}. \quad (15)$$

The interested reader is referred to [54] for further examples and properties of hard/fuzzy validation indices. The effectiveness of a particular choice of C is verified *a posteriori* by *cluster validity analysis*, performed by running the clustering algorithm for different values of C several times with different initializations. However, since different validity measures may produce conflicting results (even runs with different initializations may introduce some distortion for the same measure), it is advisable that they should be used only as guidelines to find a plausible range for the correct number of clusters.

The cluster validity problem was also tackled by unsupervised techniques with no a priori assumption on the number of clusters. Many of these approaches (e.g., [55–57]) take advantage of the fact that (4)

is minimized when the number of clusters is equal to the cardinality of the data set (when prototypes and data coincide) by adding to the cost function (4) a regularization term which is minimized when all the patterns are assigned to one cluster. These algorithms start with a large number of clusters which is progressively reduced until convergence. Regretfully, in practice, the problem of cluster validity is replaced by the determination in advance of another user-supplied parameter, with major influence in the clustering outcome and dictating which clusters are discarded.

An interesting blending between fuzzy partitional clustering techniques and hierarchical algorithms was presented in [58]. The objective is to exploit the advantages of hierarchical clustering while overcoming its disadvantages in dealing with overlap between clusters. At every new recursive agglomerative step the proposed algorithm adaptively determines the number of clusters in each bifurcation by means of a weighted version of the unsupervised optimal fuzzy clustering algorithm [26]. The final outcome of the clustering is the fuzzy partition with the best validity index value. Needless to say, the algorithm presents sensitivity to the adopted validity index.

Unsupervised stochastic techniques have also been applied to cluster validity analysis. In [59] a genetic fuzzy clustering algorithm is used for the classification of satellite images into different land cover regions. The objective function is replaced directly by a validity index (in this case the Xie–Beni index) and a variable chromosome length (depending on the number of clusters represented by each individual) allows the simultaneous evolution of solutions with a different number of clusters. The outcome is the best (in the Xie–Beni sense) of the evaluated fuzzy partitions.

7.3 Fuzzy Clustering as a Means to Information Granulation

Information granules are simultaneously a means and an objective. Because of the limited capability of human mind and sensory organs to deal with complex information its decomposition into manageable chunks of information is essential. The aggregation of similar or nearby objects into information granules (class abstraction) and the encapsulation of functional commonalities are fundamental skills for a successful approach to the great majority of problems that we face every day. This granulation may be crisp or fuzzy.

Crisp granules are derived with the apparatus of the classical set theory and are common components in various methods of information analysis, e.g., decision trees, interval analysis, or rough set theory. Fuzzy granules found their inspiration in the human capability to reason in an uncertain and imprecise environment and are supported by the theory of fuzzy information granulation [2], a part of the fuzzy sets and fuzzy logic armamentarium. Furthermore, the fuzzy logic approach relies on the notion of (fuzzy) set, opposite to the member of a classical set, to represent uncertain and imprecise knowledge. This last facet is the point of departure to the model identification with different levels of descriptive precision and granularity, viz., (fuzzy) granulation (cf. [60, 61]). In this setting, typically, a information granule is a fuzzy set and the process of information granulation consists in describing a crisp or fuzzy object as a collection of fuzzy granules (or eventually as relationships between them).

The concept of *linguistic variable* [62, 63] plays a pivotal role in this task. Informally, a linguistic variable is a granulated variable whose granular values are words or phrases represented by fuzzy sets (altogether with their connectives, modifiers, and negation). These linguistic characterizations are, usually, less specific than the numeric ones, but in compensation are safer. Thus the linguistic variable can be viewed as a way to accomplish (lossy) compression of information. Moreover the linguistic variable provides a descriptive means for complex or poorly understood systems and, more important, offers a bridge between linguistics and computation, (cf. [61]). As Zadeh [3] sharply pointed out, *the fuzziness of granules, their attributes, and their values is a central characteristic of the ways in which human concepts are formed, organized, and manipulated.*

This observation supports what seems to be one of the most human-centric approaches to discover structure in data: fuzzy clustering. The fuzzy logic approach to clustering differs from the conventional set theory approach mainly because a generic datum may belong to more than one cluster with a different degree of membership (usually a value between 0, non-membership, and 1, full degree of inclusion). Hence the data points near the core of a given cluster exhibit a higher degree of membership than

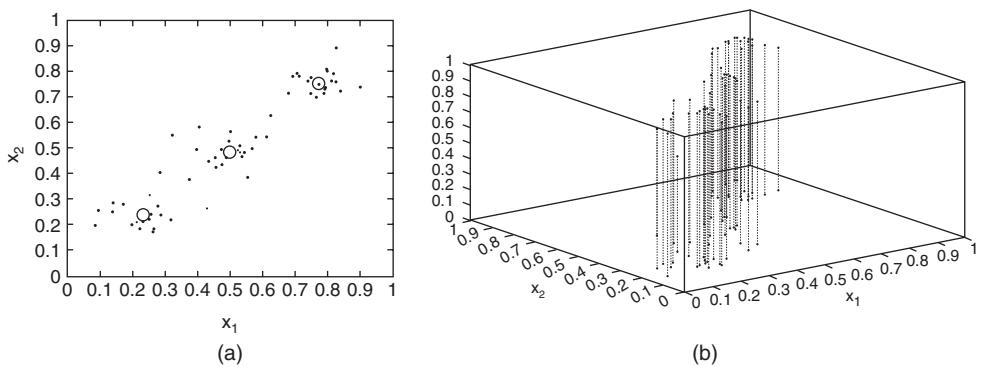


Figure 7.4 Simple data set in \mathbb{R}^2 and clustering results of the FCM algorithm. (a) Represents the data points (dots) and the clusters' centers (unfilled circles), while (b) depicts the maximum membership value of each data point

those lying farther away (near its border). With this framework it is possible to capture the uncertainty, vagueness, and flexibility inherent to the data set and to the concepts being formed.

Almost all of the (fuzzy) clustering algorithms are predominantly data driven. Given a set of numeric data, a measure of distance or similarity is established and an objective function is specified. The algorithm assigns data to clusters in a manner which guarantees minimal value for the objective function. The result of the algorithm is usually expressed in the form of a set of clusters' centers (also designed prototypes) and, in the case of fuzzy clustering, a partition matrix storing the degrees of membership of all the data points to each cluster. Usually this matrix has as many rows as the number of clusters and as many columns as the cardinality of the data set; hence, each row stores the membership degrees to a particular cluster.

In Section 7.2 we reviewed the standard FCM, its assets, and common alternatives to overcome its shortcomings. Next with the help of a visually appealing example the path leading from raw data to information granules is briefly explained. To facilitate the visualization we consider a synthetic data set defined in \mathbb{R}^2 , as depicted in Figure 7.4a.

It is composed of three visually separable clusters resulting from a normal distribution of 20 elements around three distinct points. (Table 7.1 presents the details of the distribution.)

Suppose that the clusters' centers, marked as unfilled circles in Figure 7.4a, were found by an adequate fuzzy clustering method (in this case FCM). The purpose here is to describe those clusters invoking simple fuzzy granules. Let us assume that the clustering algorithm has produced the partition matrix where each data point is characterized by a set of membership values, one per each cluster: the closer the point is to the cluster's center, the higher the membership value of that point. This relation can be perceived in Figure 7.4b, where only the maximum value of membership for each data point is shown (in the Z-axis). Each one of the resulting clusters may be conceived as a multidimensional granule; however,

Table 7.1 Details of the normal distribution of the example data set

Cluster	#Points	Mean	Standard deviation
1	20	(0.25; 0.25)	(0.05; 0.05)
2	20	(0.50; 0.50)	(0.05; 0.05)
3	20	(0.75; 0.75)	(0.05; 0.05)

Note: The three clusters are depicted in Figure 7.4a.

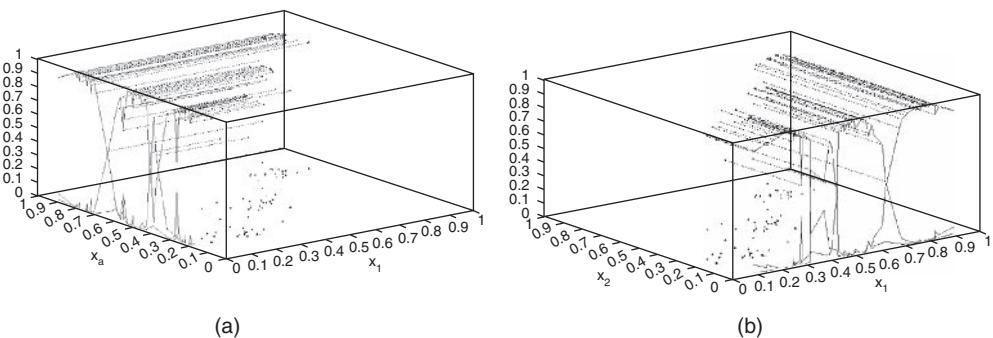


Figure 7.5 Dimensionality reduction by projection to the coordinate spaces: (a) projection to x_2 ; (b) projection to x_1

to be clearly understandable and subject to human communication it has to be expressed in terms of simpler qualitative attributes defined for each feature.

To accomplish this, first the dimensionality of this fuzzy relation is reduced by a simple operation of projection to the corresponding coordinate spaces. For every two-dimensional granule G defined on $\mathbf{X}_1 \times \mathbf{X}_2$, there are two projections $G_{\text{proj } \mathbf{X}_1}$ and $G_{\text{proj } \mathbf{X}_2}$ with the following membership functions: (For discrete sets \sup is replaced by \max .):

$$G_{\text{proj } \mathbf{X}_1}(a) = \sup_{y \in \mathbf{X}_2} G(a, y), \quad a \in \mathbf{X}_1; \quad (16)$$

$$G_{\text{proj } \mathbf{X}_2}(b) = \sup_{x \in \mathbf{X}_1} G(x, b), \quad b \in \mathbf{X}_2. \quad (17)$$

Computing the correspondent projections, each cluster induces a one-dimensional *discrete* fuzzy set per feature. Figures 7.5a and 7.5b depict these projections. (Notice that for ease of visualization the individual fuzzy sets are depicted as piecewise linear functions when, in fact, they are composed of discrete elements.)

To extend this fuzzy set to the whole one-dimensional domain an adequate enveloping fuzzy set (convex completion) or a suitable parameterized fuzzy set approximation is usually necessary. Obviously, this approximation implies some loss of information. In the given example each one-dimensional fuzzy set was approximated by a Gaussian membership function distributed around the prototype's projection (see Figure 7.6a) and altogether they form a fuzzy partition across each single domain. Finally, a last step must be performed if one wishes to describe each multidimensional granule in an human-friendly way: if possible each one-dimensional fuzzy set must be associated to a linguistic value with a clear semantic meaning. (In Figure 7.6b, S stands for small, M for medium, and L stands for large.)

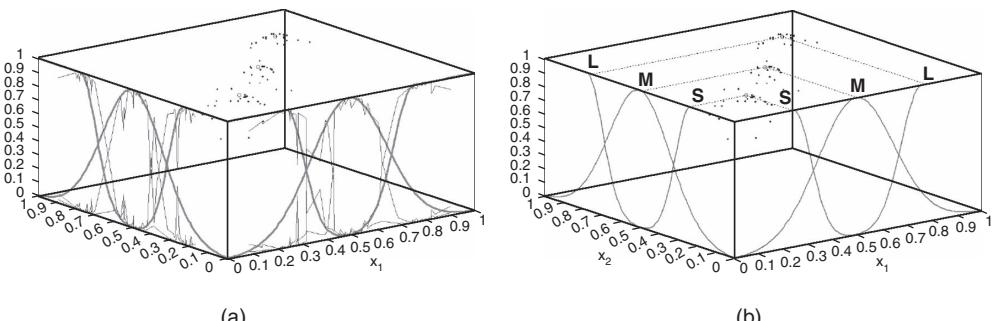


Figure 7.6 Synthesis of interpretable information granules. In (a) a functional approximation was performed followed by a semantic conversion (b) conveying meaning to each one-dimensional fuzzy set

The multidimensional granule is thus defined as a combination of one-dimensional fuzzy sets encoding linguistic labels relevant for the problem at hand. For each cluster the one-dimensional fuzzy sets where its prototype's projection attains the maximum value are chosen as the clusters' representatives. Hence each multidimensional cluster may be expressed as cartesian products of simpler granules.

Referring back to Figure 7.6b the overall data set may be entirely described in this concise form:

$$S_1 \times S_2 + M_1 \times M_2 + L_1 \times L_2, \quad (18)$$

where $+$ represents disjunction and X_1 and X_2 play the role of linguistic variables assuming the values small (S), medium (M), and large (L), necessarily with different concretization in each domain.

7.3.1 Elicitation of the Information Granules

Fuzzy clusters are information granules represented by fuzzy sets or, more generally, by fuzzy relations in some multidimensional data space. However, as was emphasized above, in order to take full advantage of their expressive power they should be able to be described as propositions in a natural language. This translation is highly dependent on the semantic soundness of the fuzzy sets in the distinct feature spaces. In this respect the set of semantic properties postulated in [64], in the context of fuzzy modeling, can be adopted as useful guidelines. These properties emerged as a means to clarify the meaning of a linguistic term (a fuzzy set) when matched against other linguistic terms in the same universe of discourse. The proposed set of properties includes a moderate number of membership functions, coverage, normality, natural zero positioning, and distinguishability.

Three of these properties seem to have an inherent interest for the clustering endeavor:

1. A moderate number of membership functions since although this number is clearly application dependent, if one intends to describe the structure of the data in a human-friendly way, there are strong reasons for imposing an upper bound on the number of clusters (in the limit, when the number of membership functions approaches the cardinality of data, a fuzzy system becomes a numeric system). This constraint makes sense not only in the feature space, where the typical number of items efficiently handled at the short-term memory (7 ± 2) [65] can be adopted has the upper limit of linguistic terms, but also in the data space since a high number of clusters result in information granules with a high granularity level.
2. Coverage which states that membership functions should cover the entire universe of discourse, so that every datum may have a linguistic representation.
3. Distinguishability since this property is clearly related with cluster separation. (Membership functions should be distinct enough from each other.)

In opposition to our previous oversimplified example (Figures 7.4–7.6) there are many situations posing several difficulties to the adequate elicitation of a *semantic mapping* between data space and feature space. Just to give an example, consider a data set with four well-separable clusters in \mathbb{R}^2 and centers in the vicinity of the vertices of a square with sides parallel to the coordinate axes. In this case the correspondent projections into the one-dimensional spaces would result in two pairs of very close fuzzy sets per feature and consequently almost undiscernible between them. The approaches to develop semantically sound information granules as a result of the fuzzy clustering process range from purely *prescriptive* methods to purely *descriptive* techniques (cf. [66]). In the prescriptive characterization of the fuzzy sets the meaningful granules are expressed intuitively by an observer in such a way that they capture the essence of the problem. The descriptive design involves the detailed computation of the membership functions based on the available numeric data.

The work presented in [67] is a good example of this latter approach complemented with some semantic concerns. The overall technique can be summarized in three steps. First, a cluster analysis is performed on the data set. The clustering algorithm (e.g., FCM) induces C information granules and this number of clusters has a major effect on the information granularity. Second, the prototypes are projected into each

dimension, their projections being further clustered in order to obtain a prespecified number of clusters, i.e., one-dimensional granules. The final step consists in quantifying the resulting one-dimensional prototypes as fuzzy sets in the feature space by means of Gaussian membership functions with a desired level of overlap. The second step of this double-clustering technique is not computationally demanding (the number of prototypes is much lesser than the number of data elements) and promotes the fusion of projections, which otherwise would result in undiscernible data sets, into one single representative, thus permitting the representation of granules via highly comprehensible fuzzy propositions.

The prescriptive approach can be illustrated by the interesting technique of context clustering [7] (see also [4]). In essence, the algorithm results from an extension to the FCM algorithm replacing the standard normalization constraint (6) by a conditioning constraint dictated by the context under which the clustering is being performed. The context is specified by the user and can assume the form of an information granule (linguistic term) defined in a peculiar feature or a logical combination between granules in the same feature space or even a composite context resulting from the Cartesian product of fuzzy sets defined in different feature spaces. Informally, we can say that the (fuzzy) linguistic context acts as a data window focusing the clustering effort on particular subsets of the data or regions of interest, thus enabling a deeper insight into the internal structure of those information granules.

The technique reported in [66] tries to present a balanced trade-off between the prescriptive and descriptive approaches. The descriptive component is represented by the clustering algorithm (the experiments report to the standard FCM) performed in the multidimensional data space. Given two different runs of the clustering algorithm, searching for a different number of clusters, the resulting granules necessarily present a different granularity level. The distinct granularity of the resulting information granules (the mixture of coarser and finer granules) can be turned into an advantage. The prescriptive component task is to conciliate different granular representations by means of specific operations of generalization (information granules combined *or*-wise) and specialization (information granules refined *and*-wise) of the fuzzy relations. The logic operators (*s*-norms and *t*-norms) are defined in advance then, if we intend to decrease the granularity of the finer result, the algorithm finds the coarser granule and the respective generalization (selected among the possible pairwise generalizations of the finer granular set) with optimal similarity (based on the overall difference between membership values of each datum in the original set and in the given generalization). On the other hand, when we intend to increase the granularity of the collection of information granules a similar process is performed, viz., the optimal replacing of a granule by the pair of granules forming its closest specialization.

The representation of information granules via multidimensional hyperboxes with sides parallel to the coordinates greatly simplifies their transparent expression as decomposable relations of classical sets in the corresponding feature spaces. In [68] the standard FCM was modified through a gradient-based technique in order to accommodate the Tchebyshev distance. This distance induces a hyper-box-shaped geometry of the clusters; however, because of the interaction between clusters there exists a deformation of the hyperboxes which need to be reconstructed in an approximate manner. When compared with FCM with Euclidean distance, the algorithm equipped with Tchebyshev distance exhibited less sensitivity to the size of the data groupings, being able to identify smaller clusters. The algorithm produces a description of the data consisting of hyperboxes (whose sizes depend on a given threshold) which encompass the core of the data and a residual portion of the data described by the standard FCM membership expression. Another interesting approach to hyperbox granulation combined with fuzzy clustering was presented in [69]. The proposed measure of information density (the ratio between cardinality and specificity of a set) is maximized in a recursive manner departing from the numeric data that are progressively mixed with the produced granular data. As a result of this granulation process the data are compressed, while the number of information granules in the high-data-density areas is reduced. Next, the information granules are clustered using the FCM algorithm combined with a parametric method of representation of the hyperboxes. This results in a collection of cluster prototypes interpretable in the original data space as hyperboxes altogether with a fuzzy partition matrix representing the membership of data into clusters. Because of the reduction of the number of information granules in high-density areas the FCM problem of underrepresenting smaller groups of data is thus obviated. Moreover, the hyperbox representation of the prototypes has direct transposition as fuzzy decomposable relations in the feature space enabling a transparent interpretation of the information granules.

As a concluding remark it is worth stressing that independently of the followed approach (descriptive, prescriptive, or both) the elicitation of information granules in a human-comprehensible way is very dependent on the characteristics of the application at hand and on the judicious decisions of the data analyst.

7.4 Conclusion

There is an increasing awareness that the modeling of vague concepts and consequent capability to reasoning about them is crucial for data analysis tasks. In this chapter we tried to present a glimpse of the process leading from the raw data to fuzzy information granules. The point of departure of our study was the broad definition of GrC in [5]:

We propose that Granular Computing is defined as a structured combination of algorithmic abstraction of data and non-algorithmic, empirical verification of the semantics of these abstractions. This definition is general in that it does not specify the mechanism of algorithmic abstraction nor it elaborates on the techniques of experimental verification. Instead, it highlights the essence of combining computational and non-computational information processing.

Consciously, as the organization of the chapter shows, we have approached the distinct facets of the definition, viz., algorithmic and semantic, inside the specific framework of fuzzy sets. From the algorithmic point of view, the fuzzy clustering mechanism has a number of well-established techniques with given proofs in myriad practical applications. Moreover the membership gradation of the data in a given cluster favors a richer insight into the data structure, especially when different clusters overlap each other. In Section 7.2 we provided a brief survey on data-driven fuzzy clustering algorithms, emphasizing their applicability as modeling tools of information granules. On the other hand, the process of instilling the real-world interpretation into data structures and its consequent validation is much more subjective. In this respect, fuzzy clustering, because of the seamless transposition of its outputs into fuzzy granules representable as linguistic propositions, seems to detain an important competitive advantage against other clustering techniques. In Section 7.3 we illustrated the elicitation of the information granules and, more important, discussed some of the techniques available to overcome interpretability problems while providing some clues to perform the semantic mapping from the fuzzy information granules into the physical world.

References

- [1] Y. Yao. Three perspectives of granular computing. In: *Proceedings of the International Forum on Theory of GrC from Rough Set Perspective, Journal of Nanchang Institute of Technology*, Nanchang, China, Vol. 25, no. 2, 2006, pp. 16–21.
- [2] L. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.
- [3] L. Zadeh. Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. *Soft Comput.* 2(1) (1998) 23–25.
- [4] K. Hirota and W. Pedrycz. Fuzzy computing for data mining. *Proc. IEEE* 87(9) (1999) 1575–1600.
- [5] A. Bargiela and W. Pedrycz. The roots of granular computing. In: *Proceedings of the 2006 International Conference on Granular Computing*, Allanta, USA, May 10–12, 2006, pp. 806–809.
- [6] Y. Yao. Perspectives of Granular Computing. In: *Proceedings of 2005 IEEE International Conference on Granular Computing*, Beijing, China, July 25–27, 2005, Vol. 1, pp. 85–90.
- [7] W. Pedrycz. Conditional fuzzy clustering in the design of radial basis function neural network. *IEEE Trans. Neural Netw.* 9 (1998) 601–612.
- [8] A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM Comput. Surv.* 31(3) (1999) 264–323.
- [9] R. Xu and D. Wunsch II. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* 16(3) (2005) 645–678.
- [10] R. Sibson. Slink: an optimally efficient algorithm for a complete link method. *Comput. J.* 16 (1973) 30–34.
- [11] D. Defays. An efficient algorithm for a complete link method. *Comput. J.* 20 (1977) 364–366.

- [12] E. Voorhees. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. *Inf. Process. Manage.* 22(6) (1986) 465–476.
- [13] G. Liu. *Introduction to Combinatorial Mathematics*. McGraw-Hill, New York, 1968.
- [14] E. Forgy. Cluster analysis of multivariate data: efficiency vs. interpretability of classifications. *Biometrics* 21 (1965) 768–780.
- [15] J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In: *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, University of California Press, Berkeley, 1967, pp. 281–297.
- [16] L. Zadeh. Fuzzy sets. *Inf. Control* 8 (1965) 338–353.
- [17] E. Ruspini. A new approach to clustering. *Inf. Control* 15 (1969) 22–32.
- [18] J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
- [19] F. Klawonn and F. Hoppner. What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier. *Lect. Notes Comput. Sci.* 2810 (2003) 254–264.
- [20] R. Hathaway, J. Bezdek, and Y. Hu. Generalized fuzzy c -means clustering strategies using L_p norm distances. *IEEE Trans. Fuzzy Syst.* 8(5) (2000) 576–582.
- [21] J. Bezdek, R. Hathaway, M. Sabin, and W. Tucker. Convergence theory for fuzzy c -means: counterexamples and repairs. *IEEE Trans. Syst. Man Cybern.* 17 (1987) 873–877.
- [22] R. Krishnapuram and J. Keller. A possibilistic approach to clustering. *IEEE Trans. Fuzzy Syst.* 1 (1993) 98–110.
- [23] R. Krishnapuram and J. Keller. The possibilistic C -means algorithm: insights and recommendations. *IEEE Trans. Fuzzy Syst.* 4 (1996) 385–393.
- [24] J. Valente de Oliveira and W. Pedrycz (eds.). *Advances in Fuzzy Clustering and Its Applications*, Wiley, New York, 2007.
- [25] D. Gustafson and W. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In: *Proceedings of the IEEE Conference on Decision Control*, San Diego, CA, 1979, pp. 761–766.
- [26] I. Gath and A. Geva. Unsupervised optimal fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 11 (1989) 773–781.
- [27] J. Bezdek, C. Coray, R. Gunderson, and J. Watson. Detection and characterization of cluster substructure, I: Linear structure: fuzzy c -lines. *J. Appl. Math.* 40(2) (1981) 339–357.
- [28] J. Bezdek, C. Coray, R. Gunderson, and J. Watson. Detection and characterization of cluster substructure, II Fuzzy c -varieties and convex combinations thereof. *J. Appl. Math.* 40(2) (1981) 358–372.
- [29] R. Dave. Fuzzy shell clustering and application to circle detection in digital images. *Int. J. Gen. Syst.* 16 (1990) 343–355.
- [30] R. Krishnapuram, H. Frigui, and O. Nasroui. Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation. *IEEE Trans. Fuzzy Syst.* 3(1) (1995) 29–60.
- [31] F. Hoppner. Fuzzy shell clustering algorithms in image processing fuzzy c -rectangular and two rectangular shells. *IEEE Trans. Fuzzy Syst.* 5 (1997) 599–613.
- [32] R. Hathaway and J. Bezdek. Switching regression models and fuzzy clustering. *IEEE Trans. Fuzzy Syst.* 1 (1993) 195–204.
- [33] F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis*. John Wiley, Chichester, England, 1999.
- [34] Y. Ohashi. Fuzzy clustering and robust estimation. In: *Proceedings of 9th Meeting SAS User Group Int*, FL, 1984.
- [35] R. Dave. Characterization and detection of noise in clustering. *Pattern Recognit. Lett.* 12 (1991) 657–664.
- [36] R. Dave and S. Sen. On generalising the noise clustering algorithms. In: *Proceedings of the 7th IFSA World Congress, IFSA'97*, Prague, Czech Republic, June 25–29, 1997, pp. 205–210.
- [37] T. Runkler and J. Bezdek. Alternating cluster estimation: A new tool for clustering and function approximation. *IEEE Trans. Fuzzy Syst.* 7 (1999) 377–393.
- [38] M. Egan. Locating clusters in noisy data: a genetic fuzzy c -means clustering algorithm. In: *Proceedings of the 1998 Conference of the North American Fuzzy Information Processing Society*, IEEE, 1998, pp. 178–182.
- [39] M. Egan, M. Krishnamoorthy, and K. Rajan. Comparative study of a genetic fuzzy c -means algorithm and a validity guided fuzzy c -means algorithm for locating clusters in noisy data. In: *Proceedings of the International Conference on Evolutionary Computation*, 1998, Anchorage, USA, May 4–9, pp. 440–445.
- [40] L. Hall, B. Ozyurt, and J. Bezdek. Clustering with a genetically optimized approach. *IEEE Trans. Evol. Comput.* 3 (2) (1999) 103–112.
- [41] F. Klawonn and A. Keller. Fuzzy clustering with evolutionary algorithms. *Int. J. Intell. Syst.* 13(10/11) (1998) 975–991.
- [42] J. Liu and W. Xie. A genetics-based approach to fuzzy clustering. In: *Proceedings of the 1995 IEEE International Conference on Fuzzy Systems*, Vol. 4, Yokohama, Japan, March 20–24, 1995, pp. 2233–2240.

- [43] C.-H. Wei and C.-S. Fahn. A distributed approach to fuzzy clustering by genetic algorithms. In: *Proceedings of the 1996 Asian Fuzzy Systems Symposium*, IEEE, 1996, pp. 350–357.
- [44] M. Sarkar. Evolutionary programming-based fuzzy clustering. In: *Proceedings of the Fifth Annual Conference on Evolutionary Programming*. MIT Press, MA, 1996, pp. 247–254.
- [45] B. Yuan, G. Klir, and J. Swan-Stone. Evolutionary fuzzy c-means clustering algorithm. In: *Proceedings of 1995 IEEE International Conference on Fuzzy Systems*, Yokohama, Japan, March 20–24, Vol. 4, 1995, pp. 2221–2226.
- [46] T. Runkler. Ant colony optimization of clustering models. *Int. J. Intell. Syst.* 20(12) (2005) 1233–1261.
- [47] T. Runkler and C. Katz. Fuzzy clustering by particle swarm optimization. In: *Proceedings of the 2006 IEEE International Conference on Fuzzy Systems*, Vancouver, Canada July 16–21, 2006, pp. 601–608.
- [48] C. Fralley and A. Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. *Comput. J.* 41(8) (1998) 578–588.
- [49] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering algorithms and validity measures. In: *Proceedings of the 13th International Conference on Scientific and Statistical Database Management*, Fairfax, USA, July 18–20, 2001, pp. 3–22.
- [50] U. Maulik and S. Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *IEEE Trans. Pattern Anal. Mach. Intell.* 24(12) (2002) 1650–1654.
- [51] G. Milligan and C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika* 50(2) (1985) 159–179.
- [52] N. Pal and J. Bezdek. On cluster validity for the fuzzy c-means model. *IEEE Trans. Fuzzy Syst.* 3(3) (1995) 370–379.
- [53] X. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 13 (1991) 841–847.
- [54] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *J. Intell. Inf. Syst.* Kluwer Publishers, Dordrecht, 17(2/3) (2001) 107–145.
- [55] H. Frigui and R. Krishnapuram. Clustering by competitive agglomeration, *Pattern Recognit.* 30(7) (1997) 1109–1119.
- [56] H. Frigui and R. Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *IEEE Trans. Pattern Anal. Mach. Intell.* 21(5) (1999) 450–465.
- [57] A. Lorette, X. Descombes, and J. Zerubia. Fully unsupervised fuzzy clustering with entropy criterion. In: *Proceedings of the 15th International Conference on Pattern Recognition*, Barcelona, Spain, September 3–8, Vol. 3, 2000, pp. 986–989.
- [58] A. Geva. Hierarchical unsupervised fuzzy clustering. *IEEE Trans. Fuzzy Syst.* 7(6) (1999) 723–733.
- [59] U. Maulik and S. Bandyopadhyay. Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. *IEEE Trans. Geosci. Remote Sens.* 41(5) (2003) 1075–1081.
- [60] L. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Syst. Man Cybern.* SMC-3(1) (1973) 28–44.
- [61] L. Zadeh. Fuzzy logic = computing with words. *IEEE Trans. Fuzzy Syst.* 4(2) (1996) 103–111.
- [62] L. Zadeh. The concept of linguistic variable and its application to approximate reasoning. *Inf. Sci.* 8 (1975) 199–249 (part I), 301–357 (part II).
- [63] L. Zadeh. The concept of linguistic variable and its application to approximate reasoning. *Inf. Sci.* 9 (1976) 43–80 (part III).
- [64] J. Valente de Oliveira. Semantic constraints for membership function optimization. *IEEE Trans. Syst. Man Cybern. Part A Syst. Man* 29(1) (1999) 128–138.
- [65] G. Miller. The magic number seven, plus or minus two: some limits on our capacity for processing information. *Psychol. Rev.* 63 (1956) 81–97.
- [66] W. Pedrycz and G. Vukovich. Abstraction and specialization of information granules. *IEEE Trans. Syst. Man Cybern. Part B* 31(1) (2001) 106–111.
- [67] G. Castellano, A. Fanelli, and C. Mencar. Generation of interpretable fuzzy granules by a double-clustering technique. *Arch. Control Sci. Spec. Issue Granular Comput.* 12(4) (2002) 397–410.
- [68] A. Bargiela and W. Pedrycz. A model of granular data: a design problem with the Tchebyshev FCM. *Soft Comput.* 9 (2005) 155–163.
- [69] A. Bargiela, and W. Pedrycz. Recursive information granulation: aggregation and interpretation issues. *IEEE Trans. Syst. Man Cybern.* 33(1) (2003) 96–112.

8

Encoding and Decoding of Fuzzy Granules

Shounak Roychowdhury

8.1 Introduction

The term *granular information* is typically used to express semantical expressions, concepts at higher levels of abstraction, and contextual communication at different levels of clarification. It is usually represented by ‘chunks’ of data or granules, also known as information granules. This type of computation is often useful for building collective knowledge to enhance intelligent decision making. Recent computing advances and availability of abundance of data have generated a tremendous interest to harness granular information from data. Information granules can be understood as collective entities that satisfy properties of functional similarity, spatial proximity and orientation, structural equivalences, or even collection of simple rules, and hierarchy of information content. Very often granularity can be realized in terms of the amount of raw data content and the amount of aggregated information that is often described as a hierarchy of grain sizes, either starting from fine grain to coarse grain or vice versa. The study of information granules and their interactions and transformations form the core of granular computing [1]. The computational representation of information granules is usually done using a variety of existing uncertainty modeling concepts like singleton, interval-valued set, rough set, and fuzzy set to capture different levels of abstraction.

In this chapter we will explore transformation from imprecise domain into precise domain and vice versa. Particularly, we will focus on these transformations from the perspective of fuzzy sets. Typically, a crisp encoding is defined as a process of associating an external symbol (input vector) to an internal symbol (code) belonging to an alphabet and an inverse process for crisp decoding. Fuzzy encoding (\mathcal{F}) generalizes the crisp coding by associating each fuzzy set (in the universe of discourse) to all internal symbols with different degrees of membership. Therefore, fuzzy encoding realizes a mapping as shown below:

$$\mathcal{F} : \mathbf{X} \rightarrow [0, 1]^n,$$

where \mathbf{X} is the domain input variable or a symbol set, and n is the number of fuzzy sets. Similarly, fuzzy decoding (\mathcal{F}^{-1}) is an inverse process of associating each fuzzy set with all possible number of symbols belonging to the symbol set:

$$\mathcal{F}^{-1} : [0, 1]^n \rightarrow \mathbf{X}.$$

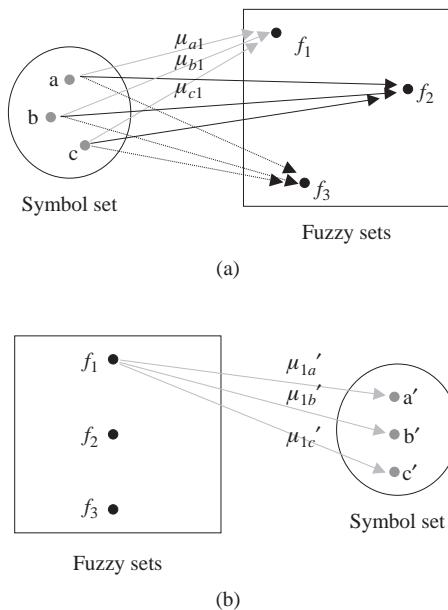


Figure 8.1 (a) Fuzzy encoding: it is a mapping from all elements of the symbol set to every fuzzy set in the universe of discourse. (b) Fuzzy decoding: it is a mapping from the all fuzzy sets in the universe of discourse to all elements of the symbol set. Each mapping has membership value between 0 and 1

In the above sense the encoding and decoding are complementary processes. Figure 8.1 shows the fuzzy encoding and fuzzy decoding. In Figure 8.1a, we assumed for simplicity that the symbol set has three elements **a**, **b**, and **c**. There are three fuzzy sets f_1 , f_2 , and f_3 with two support elements. The symbol **a** is associated to fuzzy set f_1 with degree μ_{a1} , **b** is associated to fuzzy set f_2 with degree μ_{b1} , and **c** is associated to fuzzy set f_1 with degree μ_{c1} . Similarly, the mapping for other symbols to fuzzy sets can be defined. Thus, in fuzzy encoding and decoding a mapping between every symbol in the symbol set and every fuzzy set is defined in the domain of discourse. It is easy to notice that the symbol set and domain of fuzzy sets form a complete bipartite graph where the edges have weights given by the membership values.

Fuzzy encoding and fuzzy decoding form a general framework for transformation among information granules. Within this framework the choice of associating a symbol with a fuzzy set is performed by the fuzzification operator $FUZ(\cdot)$ and defuzzification operator $DEFUZ(\cdot)$. In the literature, $FUZ(\cdot)$ is also accepted as a fuzzy granulation operator, while $DEFUZ(\cdot)$, being a complementary operator, is thought of as a degranulator. These operators have been described as fuzzy interfaces in making of fuzzy systems. In [2, 3], Oliveira used these interfaces for converting numerical variables to linguistic variables and vice versa in construction of fuzzy control systems. Subsequently, Pedrycz and Oliveira [4] used fuzzy interfaces in construction of optimal fuzzy models.

The fuzzification operator is given by $FUZ(\cdot)$,

$$FUZ : x \rightarrow A(x),$$

where $A = \{\frac{\mu_1}{x_1}, \dots, \frac{\mu_i}{x_i}, \dots, \frac{\mu_n}{x_n}\}$ is a discrete fuzzy set and its membership function is given by $\mu(\cdot)$. For each value of x_i , the corresponding membership value is $\mu(x_i) = \mu_i$. In early fuzzy literature it was considered as a transformation from precise to imprecise domains.

Similarly, on converse, the transformation from imprecise into precise domain has been known as defuzzification. The defuzzification operator is given by $DEFUZ(\cdot)$,

$$DEFUZ : A(x) \rightarrow x.$$

At this point it is important to realize that even though intuitively it easy to believe that fuzzification and defuzzification as complementary operators, i.e., $DEFUZ(FUZ(x)) = x$, however that viewpoint is not particular always true. Furthermore, this view of the operators being complementary to each other might be too restrictive. Naturally, this has lead us to seek a broader understanding of fuzzification and defuzzification in terms of fuzzy encoding and decoding mechanisms.

The Figure 8.1 illustrates how fuzzification and defuzzification form a special case of fuzzy encoding and fuzzy decoding mechanisms. Let us assume that $\mu_{a1} \leq \mu_{b1} \leq \mu_{c1}$, then the $FUZ(\cdot)$ operator can be characterized as a process that chooses the mapping with maximum membership value. For instance, in this case we can associate f_1 with **c** because the latter has maximum membership value $\max(\mu_{a1}, \mu_{b1}, \mu_{c1})$ with the former. Thus a symbol or singleton gets bound to a fuzzy set. Choosing the association with maximum membership value is one such strategy, and perhaps there may be other mapping strategies for binding singleton to fuzzy set that could be explored. In this chapter we would use the maximum membership value of the association between a symbol and a fuzzy set. Similarly, in case of fuzzy decoding, as shown in Figure 8.1b, f_1 is associated with the symbol set $\{a', b', c'\}$. Likewise, we assume that $\mu_{1a'} \leq \mu_{1b'} \leq \mu_{1c'}$, then the $DEFUZ(\cdot)$ operator can be characterized as a process that chooses the mapping with maximum membership value $\mu_{1c'}$, which leads to the selection of symbol **c'** as the defuzzified value. Typically for all practical engineering problems the symbol set is the real line.

We will discuss the methods available for fuzzification in Section 8.2 and defuzzification in Section 8.3. The intent of this chapter is to provide an overview of these two interesting and essential operators that are often very useful in granular computing.

8.2 Fuzzification

Essentially the concept of fuzzification means to generate a membership function for a fuzzy set in the universe of discourse from a singleton value. In other words, it is to find the collection of data that not only encapsulates the singleton, but also captures information about its neighborhood in a graded and meaningful fashion. This process of capturing approximate information is quite subjective, depends on who is making the decision, and is often problem- and domain dependent. Dubois and Prade pointed out in [5], there is no uniformity in the interpretation of what a membership grade means. Perhaps because of this subjectivity there is no standard algorithm, no sound analytical approach, or a theoretical explanation that can fully characterize the process of fuzzification. Perhaps the answer lies in the subjective interpretation of the designer who creates the fuzzy sets. Since fuzzification is problem dependent, quantification of fuzzification research has been a slow progress, and unfortunately there are limited number of works in the fuzzy research in this regard. Turksen [6] under the framework of measurement theory summarized most of the earlier works based on vague and imprecise concepts [7].

Perhaps we can broadly separate the process of fuzzification into two categories – functional approach and data-driven approach. In the functional approach, a functional generator is used to generate the membership functions, whereas the data-driven approach generates the membership functions from the data either using unsupervised learning techniques like clustering or using expert's characterization of the data. However, it should be noted that there is no sharp distinction between these two categories. In some cases, a small amount of preliminary data analysis may required in order to derive optimal parameters for the functional operators.

8.2.1 Functional Approach

Fuzzy sets can be generated analytically by identifying the type of membership functional. There are six well-known membership function generators and they are listed below. In practice, triangular membership functions and Gaussian membership functions are mostly used.

- *Triangular membership:*

$$\mu(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{m-a} & \text{if } x \in [a, m] \\ \frac{b-x}{b-m} & \text{if } x \in [m, b] \\ 0 & \text{if } x > b \end{cases}, \quad (1)$$

where m is a modal value and a , and b are lower and upper bounds.

- *Trapezoidal membership:*

$$\mu(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{m-a} & \text{if } x \in [a, m] \\ 1 & \text{if } x \in [m, n], \\ \frac{b-x}{b-m} & \text{if } x \in [n, b] \\ 0 & \text{if } x > b \end{cases} \quad (2)$$

where m and n are modal values, and a and b are lower and upper bounds.

- *Gaussian membership:* This membership generates the bell-curve-like membership centered around the modal value m :

$$\mu(x) = e^{-k(x-m)^2}, \quad (3)$$

where $k \geq 0$.

Among other less used membership functions they are as follows:

- *Gamma membership (Γ):*

$$\mu(x) = \begin{cases} 0 & \text{if } x < a \\ 1 - e^{-k(x-a)^2} & \text{if } x \geq a, \end{cases} \quad (4)$$

where $k \geq 0$.

- *S-membership:*

$$\mu(x) = \begin{cases} 0 & \text{if } x < a \\ 2\left(\frac{x-a}{b-a}\right)^2 & \text{if } x \in [a, m] \\ 1 - 2\left(\frac{x-b}{b-a}\right)^2 & \text{if } x \in [m, b] \\ 0 & \text{if } x > b \end{cases}, \quad (5)$$

where $m = \frac{a+b}{2}$ and is called crossover point.

- *Exponential membership:* This membership generates the bell-curve-like membership centered around the modal value m :

$$\mu(x) = \frac{1}{1 + k(x - m)^2}, \quad (6)$$

where $k \geq 0$.

8.2.2 Data-Driven Approach

There are data-driven methods through which one can generate fuzzy sets. They include estimation made by experts who know the characteristics of the data to automatic generation of membership functions using non-parameteric statistical techniques like nearest-neighbor algorithms, variants of unsupervised learning algorithms like clustering algorithms such as k -means or fuzzy- c means or its variants, and neural networks.

8.2.2.1 Expert's Estimation

In practice, the fuzzy sets are often modeled by experts who have a good understanding of the data under investigation. However, this approach may not be possible in many cases, as it is difficult to find experts for each and every data set. However, we will briefly mention a few methods that are used by the experts – (1) polling, (2) direct estimation, and (3) experience- and intuition-based estimation. The experts typically either use the polling method or estimate directly using statistical measurements. They also use their experience and intuition to guide and fine-tune estimated values.

8.2.2.2 Automatic Membership Generation

The approach is to computationally understand the nature of the data through repeated experiments and observations. The statistical results can be interpreted as linguistic variables. A survey by Medasani *et al.* [8] describes automatic generation of membership functions from the data using a variety of techniques like heuristics, histograms, nearest-neighbor techniques, etc.

We briefly outline the possible ways to estimate the membership function in our context, where we are given classes and respective training data. We would like to see the options that we can use in our case.

- *Histograms:* Fuzzy sets and their relationship to histograms have been studied as early as 1985 by Devi and Sarma [9]. A normally scaled histogram can be interpreted as a membership function. Recently there is a surge in interest in the modeling of membership functions from histograms [10] and statistical data.
- *Nearest neighbors:* There are two approaches here:
 1. We can estimate the degrees of membership of the labeled data by measuring the distance $d(\cdot)$ (l^p -distance) from the prototype. The prototype is an average of the collection of a single class. Let \mathbf{x}_f be the farthest data from the prototype, x_p , in the class and x_i be the i th data point in the given class. The membership of each data point is thus given by

$$\mu(x_i) = \frac{1 - \frac{d(x_i - x_p)}{d(x_f - x_p)}}{\max_i\{d(x_i - x_p)\}}. \quad (7)$$

Apply normalization to ensure that we have a normal fuzzy set because x_p need not be part of the data.

2. The second method uses k nearest neighbors to generate the membership, which is given by

$$\mu(x_i) = \frac{k_i}{k},$$

where k_i is the number of elements among k closest neighbors having the i th class.

- *Clustering:* Assuming that there are c clusters of data with centroid $\bar{u}_1, \bar{u}_2, \dots, \bar{u}_c$, then membership function in a cluster C_j can be written as

$$\mu_{C_j}(\mathbf{x}) = \frac{(x_j - \bar{u}_j)^{-2}}{\sum_{i=1}^c (x_i - \bar{u}_i)^{-2}}. \quad (8)$$

Chen and Wang [11] used fuzzy clustering analysis for optimizing fuzzy membership functions using cluster validation technique. They used their method for the problem of fuzzy identification. Liao *et al.* [12] also used a variant of fuzzy c -means (FCM) algorithm for the generation of fuzzy term sets and optimized the membership functions.

- *End-data approximation:* Usually, this method is applicable to one-dimensional cases where the end data points are computed with respect to the *max* and *min* operators based on the value. Then we can define the parameters to generate either a triangular or a trapezoidal membership function.
- *Fuzzy neural networks:* There are a couple of interesting works that use fuzzy neural network to understand the process of fuzzification. Buckley [13] has tried to explain fuzzifying operator on real numbers by having functional constraints, such as $y = f(x)$. This functional constraint could use the extension principle or otherwise use α -cuts and interval arithmetic. He has used fuzzy neural network to emulate input–output relationship as crisp–fuzzy and fuzzy–fuzzy relationship. Bortolan and Pedrycz [14] have addressed the problem of reconstructing the fuzzy set by analyzing the input–output relationship. They tried to address the reconstruction mechanism from the perspective of mean square error (MSE) optimization. They implemented their theory in terms of fuzzy neural network.
- *Self-organizing maps:* Yang and Bose [15] have proposed a novel and robust scheme to generate fuzzy membership functions with unsupervised learning using self-organizing feature maps (SOMs).
- *Probability–possibility transformation:* We know statistical data generate histograms and as mentioned earlier researchers have been interested in using the theory of probability and statistical approaches. Recently, Dubois and Prade have been exploring the relationship between possibility theory and statistical reasoning [16]. Using the probability–possibility transformation of Dubois and Prade, Masson and Deneux [17] have devised a simple algorithm to infer possibility distributions from empirical data.

In this section we showed a number of fuzzification approaches that are being used in practice. In the next section we will explore the inverse process of fuzzification called defuzzification.

8.3 Defuzzification

Defuzzification does degranulation of a fuzzy set – a process contrary to fuzzification. This operator has been studied extensively [18] with respect to performance of fuzzy rulebases of fuzzy controllers [19]. The classical relationship between fuzzy controllers and standard defuzzification operations and defuzzifier components is well mentioned in [20]. Since the beginning of the fuzzy research the role of the defuzzifier was defined as to transform a fuzzy set into a numeric value. This is partially because fuzzy controllers used rules having consequents that were described by fuzzy sets whose support sets were real intervals. Therefore the feedback parameter was a real value. Therefore, from an engineering and practical perspective the conversion of a fuzzy set into a numeric value was easily accepted and the definition has continued in practice till date. However, many researchers have attempted to provide their insight and proposed alternative solutions.

8.3.1 Properties

Runkler and Glesner [21] provided a set of 13 features that are observed by most of the defuzzification methodologies. Here again, they have assumed that defuzzification operation takes a fuzzy set and transforms it into a numeric value. We will summarize 13 of their features into 5 core properties.

1. *Defuzzification operator generates a singleton.*
2. *Concentration and dilation operator affect the defuzzified value.* Repeated use of concentration operator on a fuzzy set monotonically leads to the defuzzified value. Similarly, repeated use of dilation operator on a fuzzy set monotonically leads the defuzzified value away from the normal of a fuzzy set.
3. *Defuzzification is not affected by scaling of membership values.*
4. *Translation of a fuzzy set also translates the defuzzified value.*
5. *The defuzzified value of a fuzzy set derived from union (T-conorm) or intersection (T-norm) of two fuzzy sets is always contained within the defuzzified values of those two fuzzy sets.* If A and B are fuzzy sets, then for both T-norm (T) or T-conorm (T^*) of a fuzzy set, $\text{Def}(A) \leq \text{Def}(C) \leq \text{Def}(B)$.

8.3.2 Standard Methods

This section discusses four most commonly used defuzzification methods still used in practice. They are mean of maxima, center of means, center of gravity, and midpoint of area. Since long these methods have been used so often in fuzzy control and other applications that they have become as de facto standard methods for defuzzification. Moreover, these methods are simple and have been so popular. There are several others like height defuzzification, first of maxima, last of maxima, etc. [22], but we would like to concentrate here on the standard ones:

- *Mean of maxima (MOM):* This method was first proposed by Mamdani in his fuzzy controller design. It computes the center of gravity of the area under the maxima of the fuzzy set. Then,

$$\bar{x} = \frac{\sum_{i=1}^n (x_i)}{\{i | \mu_i = \max\{\mu_1, \dots, \mu_n\}\}}. \quad (9)$$

This method is fast, but it is also a poor approximator of optimal defuzzified value. It has been shown that the MOM method generates poor steady-state performance and yields a less smooth response curve compared with the COG method.

- *The center of maxima (COM):* This procedure computes a defuzzified value of a fuzzy set,

$$\bar{x} = x_h,$$

where

$$\sum_{i=\min M}^h \mu(x_i) = \sum_{i=h}^M \mu(x_i)$$

and

$$M = \{i | \mu_i = \max\{\mu_1, \dots, \mu_M\}\}. \quad (10)$$

- *Midpoint of area (MOA):* This method is given by

$$\bar{x} = x_h,$$

where

$$\sum_{i=h}^n \mu(x_i) = \sum_{i=1}^h \mu(x_i)$$

and

$$M = \{i | \mu_i = \max\{\mu_1, \dots, \mu_n\}\}. \quad (11)$$

- *Center of gravity (COG):* For most practical purposes we think that the minimization of a simple cost function provides us with a solution that we admire. The COG defuzzification method is simple, elegant, and rationally convincing. The COG method generates a value that is the center of gravity of a fuzzy set. It actually minimizes the membership graded weighted mean of the square of the distance. Let us consider a cost function that is weighted by its membership function,

$$K(\bar{x}) = \sum_{i=1}^n (x_i - \bar{x}_i)^2 \mu(x_i). \quad (12)$$

Minimization of $\mu(\cdot) = \mu(x_i)$ by differentiation, we have

$$\frac{K(\bar{x})}{d\bar{x}} = 2 \sum_{i=1}^n (x_i - \bar{x}_i) \mu(x_i) = 0, \quad (13)$$

generates the COG, and it is given by

$$\bar{x} = \frac{\sum_{i=1}^n (x_i) \mu(x_i)}{\sum_{i=1}^n \mu(x_i)}. \quad (14)$$

COG is computationally expensive; therefore, there have been attempts to seek faster algorithms that would nearly approximate the COG method. Patel and Mohan [21] have performed numerical experimentations for faster COG algorithms. Recently, a similar effort was noted in [24].

Many researchers have proposed variants of these methods as in [22, 24, 25]. Most of these variants have some flavor of geometrical intuitions behind them and are somewhat ad hoc methods, and till date there are no convincing reasons behind these methods except for COG. However, these methods remain popular and they have been successfully implemented in many fuzzy applications.

At same time there have been other attempts to understand the logic and the workings of the defuzzification process, which we will discuss in the next section.

8.3.3 Alternative Defuzzification Approaches

In this section we will browse through some of the alternative and recent approaches that have been proposed by researchers over the years.

Thiele [26] tried to develop an axiomatic theory of defuzzification based on theory of groups, theory of partially ordered sets, and Galois connection.

8.3.3.1 Unit Hypercube Defuzzification

- *Nearest vertex defuzzification:* Kosko in [27] has described fuzzy set as a point in a unit hypercube defined by the Cartesian product of a unit interval $I^n = [0, 1]^n$. The vertices of the cube I^n define non-fuzzy sets. Therefore, there are 2^n non-fuzzy sets. Furthermore, Kosko has also defined the l^p distance between two fuzzy sets A and B in the unit hypercube as $l^p(A, B) = \sqrt[p]{\sum_{i=1}^n |\mu_A(x_i) - \mu_B(x_i)|^p}$, where $M(A) = \sum_{i=1}^n \mu(x_i)$ is the σ -count (cardinality of the fuzzy set). Out of that there are only n vertices that have one element that has a unit indicator function. Just to mention that the fuzzy set at the center of the unit hypercube has maximum fuzziness, and membership for all the elements is 0.5. See Q in the Figure 8.2.

From this geometrical viewpoint we can interpret defuzzification as a selection process where it should select anyone of n vertices. Each of the n vertices has one element and is just a singleton. If we can find the nearest singleton to a fuzzy set, then we can associate (decode) it as the defuzzified value. Thus defuzzification process can be understood as finding the nearest vertex, or whose distance from the fuzzy set is minimum. The Figure 8.2 illustrates the point more clearly. Here a fuzzy set denoted by

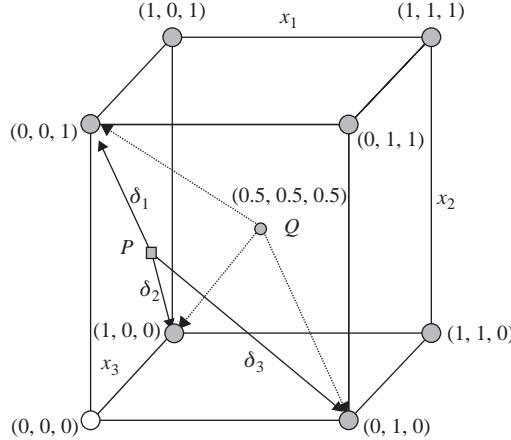


Figure 8.2 From a geometric viewpoint, defuzzification can be interpreted as a selection of the nearest vertex of the unit hypercube to the fuzzy set. Each vertex is a singleton

P in the unit hypercube is shown by a square. The closest vertex to P is $\{1, 0, 0\}$, which is computed by finding the minimum of $\delta_1 = l^p(P, \{0, 0, 1\})$, $\delta_2 = l^p(P, \{1, 0, 0\})$, and $\delta_3 = l^p(P, \{0, 1, 0\})$. The distance δ_2 being smallest we choose $\{1, 0, 0\}$ as the defuzzified representative.

Figure 8.3 shows four fuzzy sets and their positions in the unit hypercube I^3 . The first fuzzy set (i) has maximum fuzzy entropy (see [27] for details). As a fuzzy set gradually moves toward a vertex in its own unit hypercube, that gradually reduces the fuzzy entropy and finally at the vertex the fuzzy set is reduced to a singleton $\{0, 0, \dots, 1, \dots, 0\}$. In other words, we have x_i as a selected or defuzzified value. This method only performs distance comparisons with respect to all the vertices to find a selected element. This is a geometric interpretation for performing the $\max(\cdot)$ operation. Similar interpretations can be drawn for other methods as well.

- **Subsethood defuzzification:** Oliveira in [28] based on Kosko's subsethood theorem [27] also tried to explain the process of defuzzification. The subsethood defined by Kosko is degree to which fuzzy set X is a subset of fuzzy set Y is also given by

$$S(X, Y) = \frac{M_p(X \cap Y)}{M_p(X)},$$

where σ -count of a fuzzy set X is defined as $M_p(X) = (\sum_{i=1}^n \mu_X^p(x_i))^{1/p}$. For any given fuzzy set A , the averaging defuzzification strategy is expressed in terms of set measures as shown below:

$$\bar{x} = \frac{M_p(A \cap \mathcal{M}_A)}{M_p(A)},$$

where \mathcal{M}_X is the mirror support set of X and is expressed as $\sum_{i=1}^n \mu(x_i)/x_i$. As shown in his paper,

$$\hat{x}_{SD} = \frac{M_p(A \cap \mathcal{M}_A)}{M_p(A)}, \quad (15)$$

$$= \frac{\left(\sum_{i=1}^n T(\mu_A^p(x_i), x_i) \right)^{1/p}}{\left(\sum_{i=1}^n \mu_A^p(x_i) \right)^{1/p}}, \quad (16)$$

and T is a t -norm operator. It can be easily shown that COG and MOM are special cases of equation (16).

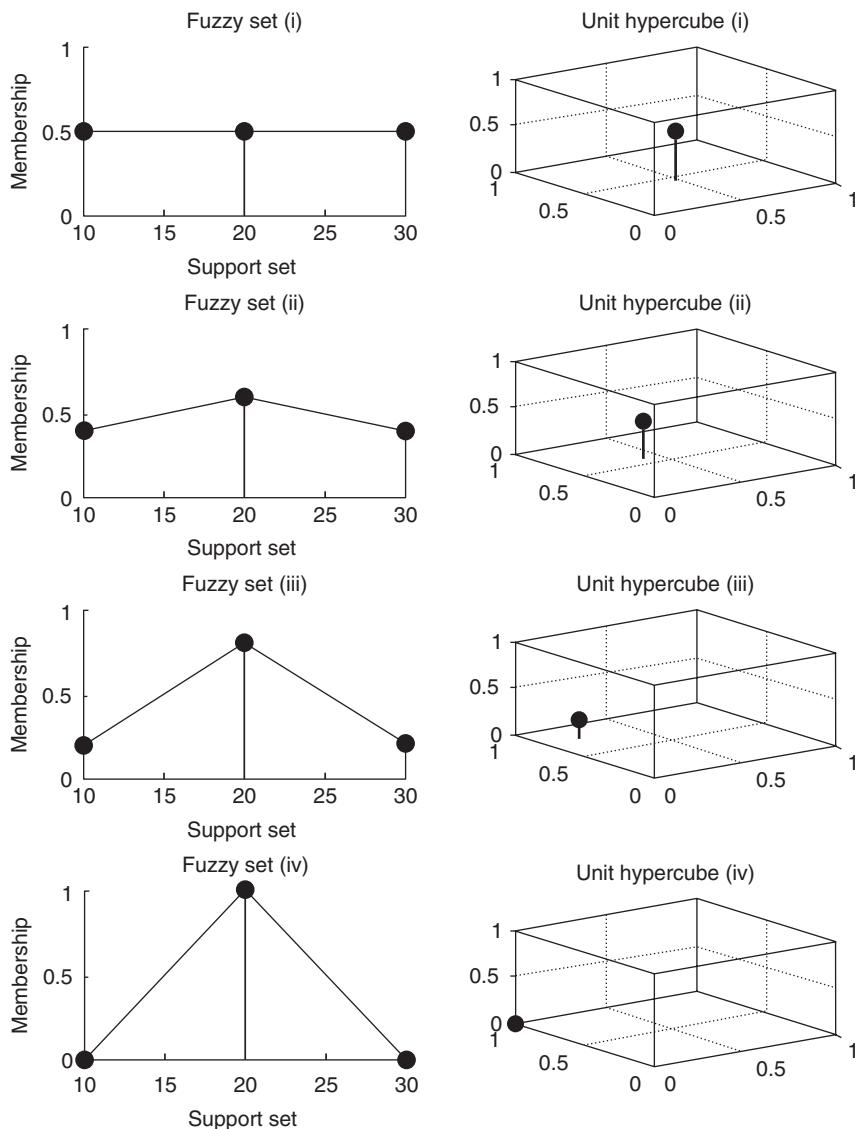


Figure 8.3 The transformation of fuzzy set (i) with maximum fuzzy entropy to a fuzzy set with only element having non-zero membership value. Note the point in the unit hypercube settles to a vertex which can be treated as a defuzzified value

Kim *et al.* [29] critiqued the above approach with some counterexamples. In their paper they mentioned that Oliveira used a product as t -norm operator, and when other t -norms (such as \min operator) were used the subsethood defuzzification did not produce expected results.

8.3.3.2 Probability–Possibility Defuzzification

Filev and Yager [30] proposed methods based on the basic defuzzification distribution (BADD) approach. Their main idea has been to transform a possibility distribution to a probabilistic distribution based on Klir's principle of uncertainty invariance [31] (also see [32]). BADD transforms a concentration or

dilation operator to a desired degree depending on δ . The BADD transformation is $F_2(x_i) = (F_1(x_i))^\delta$, where $\delta \in (0, \infty)$ and F_1 and F_2 are the original and transformed fuzzy sets, respectively.

The BADD method converts the possibility distribution of a normalized fuzzy set into a probability distribution, by using a transformation $\omega(x_i) = K(\mu(x_i))^\delta$. Earlier, this mapping was proposed and was used by Klir to provide a conversion framework between probability and possibility distributions. The probability distribution is given by

$$p_i = \frac{\mu(x_i)^\delta}{\sum_{i=1}^n \mu(x_i)^\delta} \quad (17)$$

after imposing the convexity conditions on the above transformation. They were able to show that the expected value $E(p_i)$ is given by

$$\bar{x} = \frac{\sum_{i=1}^n x_i \mu(x_i)^\delta}{\sum_{i=1}^n \mu(x_i)^\delta}. \quad (18)$$

The BADD method generates COG and MOM for $\delta \rightarrow 1$ and $\delta \rightarrow \infty$, respectively.

They also proposed SLIDE (semilinear defuzzification) and an adaptive method that used a linear transformation to transform an original fuzzy set into another fuzzy set that is given by

$$T_{\alpha,\beta} : F_1 \rightarrow F_2, \quad (19)$$

where

$$\mu_i = \begin{cases} \eta_i & \text{if } \eta_i \geq \alpha \\ (1 - \beta)\eta_i & \text{if } \eta_i \leq \alpha \end{cases} \quad (20)$$

and η_i and μ_i are the membership values of fuzzy sets F_1 and F_2 for a particular alpha-cut α , respectively. See [33] and [34] for details. Here are some of the transformations that have been proposed in the literature. It is clear that the transformation $T_{\alpha,\beta} : F_1 \rightarrow F_2$ preserves the shape and the values of the membership function η_i for values of η_i that are equal to or higher than α . Thereafter, the authors construct a probability distribution by normalizing the sequence of μ_i . The probability distribution is given by

$$P_i = \frac{\mu_i}{\sum_{i=1}^n \mu_i}, \quad (21)$$

such that $\sum_{i=1}^n P_i = 1$, and $P_i \geq 0$, $i = (1, n)$.

The properties of SLIDE are proved in [33]. The expected value defined by SLIDE transformation is given by

$$\bar{x} = \frac{(1 - \beta) \sum_{i \in L} x_i \mu(x_i) + \sum_{i \in H} x_i \mu(x_i)}{(1 - \beta) \sum_{i \in L} \mu(x_i) + \sum_{i \in H} \mu(x_i)}, \quad (22)$$

where $L = \{i | \eta_i < \alpha, i = (1, n)\}$ and $H = \{i | \eta_i > \alpha, i = (1, n)\}$. Another extension of SLIDE is called modified-SLIDE (M-SLIDE), and the interested reader can see [33] and [35]. Apparently, there is not much of a significant difference of concepts in both SLIDE and M-SLIDE. In [33], Yager and Filev discuss the multiplicative transformation, which is given by

$$T_{\alpha,\beta} : \mu_i = \eta_i^\alpha s_i^\beta, \quad (23)$$

where $s_i \geq 0$ and is the scaling factor for η_i , $i \in [1, m]$, and $\alpha, \beta \geq 0$. Refer to [33] for details.

Yager in [35] proposed the random generation (RAGE) method. Similar to the earlier methods of BADD and SLIDE a fuzzy set is transformed into a probability distribution. Divide the support set of the fuzzy set in such a way that each of the elements falls into a given probability range. In addition, perform a random experiment to choose the defuzzified element. The viability of this method has been shown in fuzzy control experiments where inferred fuzzy sets exhibit prohibitive information.

8.3.3.3 Neighborhood Defuzzification

- *Cooperative neighbors:* In [36], Roychowdhury and Wang attempted to explain the concept of defuzzification using the ideas of evolutionary biology. In evolutionary biology, the ecology is functionally dependent on the interaction of the different species and is guided by the principle of natural selection. They proposed a variety of variations functions, $v(\cdot)$, that were derived by modeling the biological concepts of interaction among the neighboring elements in the set. All the neighbors interact with each other to reach a consensus. As described in the paper, we have

$$v(x_i) = \frac{f(\mu(x_1), \dots, \mu(x_n))}{\sum_{i=1}^n f(\mu(x_1), \dots, \mu(x_n))}, \quad (24)$$

and finally by using the quadratic minimization they were able to show

$$\bar{x} = \frac{\sum_{i=1}^n v(x_i)x_i}{\sum_{i=1}^n v(x_i)}. \quad (25)$$

The above model generalizes COG and BADD for $v(x_i) = \mu_i$ and $v(x_i) = \mu_i^\alpha$. This is a very generic model that does not need to convert a possibility distribution into a probability distribution and so on; rather, it uses an alternative approach to understand the problem of defuzzification.

- *Radial defuzzification:* In another paper [37] Roychowdhury *et al.* extended the above concept of applying cooperative elements of a fuzzy set to understand the defuzzification process. The method is called the radial defuzzification method. In short, the basic idea is to use a Cartesian product of interaction $A_f \times A_f$ called the i -space of a fuzzy set A_f , such that

$$\mu_{ij} = A_f \times A_f \rightarrow [0, 1]. \quad (26)$$

This means that the ordered pair (i, j) of the i -space consists of interaction between the i th and the j th element of the same fuzzy set. The interaction value on the ordered pair is denoted by μ_{ij} . It is computed by using membership values of the i th and the j th element of the fuzzy set and is given by

$$\mu_{ij} = g(\mu_i, \mu_j). \quad (27)$$

The paper discusses the effect of interacting neighbors with a radial zone of the i -space. Here, we only provide the results of the simplest case when neighbors do not interact and when only nearest neighbors interact. In the first case, we observe that

$$\mu_{ij} = g(\mu_i, \mu_j) = \mu_i \mu_j. \quad (28)$$

Therefore, the variation function is

$$v(x_i) = \frac{\mu(x_i)^2}{\sum_{i=1}^n \mu(x_i)^2}. \quad (29)$$

Additionally, the defuzzified value is thus given by

$$\bar{x} = \frac{\sum_{i=1}^n \mu_i^2 x_i}{\sum_{i=1}^n \mu_i^2}. \quad (30)$$

We find that the above equation is similar to the BADD method. Notice that the value of α is 2. In another case when nearest neighbors interact, we have the following defuzzification result where the variation function $v(x)$ is given by

$$v(x_i) = \frac{\mu_i(\mu_i + \frac{1}{2}(\mu_{i-1} + \mu_{i+1}))}{\sum_{i=1}^n \mu_i(\mu_i + \frac{1}{2}(\mu_{i-1} + \mu_{i+1}))}. \quad (31)$$

In addition, therefore, the defuzzified value is

$$\bar{x} = \frac{\sum_{i=1}^n \mu_i^2 x_i}{\sum_{i=1}^n \mu_i^2}. \quad (32)$$

8.3.3.4 Fuzzy Clustering in Defuzzification

Genther *et al.* [38] proposed another alternative to understand the process of defuzzification. Their approach is based on fuzzy clustering. Fuzzy clustering enables us to partition the data into subclasses by generating fuzzy sets. In other words, fuzzy membership is assigned to each cluster. The FCM algorithm is a well-known fuzzy clustering algorithm. It is an optimization problem dealing with the minimization of the following objective functional;

$$J(U, v) = \sum_{i=1}^n \sum_{j=1}^c u_{ij}^m \|x_i - v_j\|^2, \quad (33)$$

where $\|x_i - v_j\|$ is a distance function between the cluster center v_j and datum x_i . The membership is calculated by

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - v_k\|^2}{\|x_i - v_j\|^2} \right)^{\frac{1}{m-1}}} \quad (34)$$

and $1 \leq j \leq c$ and $1 \leq i \leq n$. The centers of the clusters are computed by

$$v_j = \frac{\sum_{i=1}^n u_{ij}^m x_i}{\sum_{i=1}^n u_{ij}^m}. \quad (35)$$

This is a standard procedure to compute the FCM algorithm. Genther *et al.* [38] modified the distance function between a datum x_i and the associated membership value $\mu(x_i)$ to $d_{ij} = \sqrt{\alpha \frac{|x_i - v_j|}{(x_i - v_j)_{\max}} + (1 - \alpha) \frac{|x_i - v_j|}{(x_i - v_j)_{\min}}}$, and

$$w_j = \frac{\sum_{i=1}^n u_{ij}^m \mu(x_i)}{\sum_{i=1}^n u_{ij}^m \mu(x_i)}. \quad (36)$$

Finally on obtaining c -cluster centers with associated membership values (v_j, w_j) , they chose the cluster center k with $w_k = \max_j(w_j)$ and the defuzzified value was its center v_k . Interestingly, by adjusting the parameter α they were able to achieve acceptable defuzzification results for fuzzy sets having prohibitive information.

8.3.3.5 Neural Networks for Defuzzification

It has been found in many cases that the choice of defuzzification method can be critical in designing fuzzy systems. Usually, trial-and-error methods are employed to find out the suitable defuzzifier for a given fuzzy control system. Tuning defuzzification strategy to an application is a desirable option. In [39], Halgamuge and Glesner describe a customizable BADD (CBADD) defuzzifier that uses a transparent neural network to tune defuzzification to different applications. Song and Bortolon [40] proposed some properties of neural networks that learn the defuzzification process. Song and Leland [41] have used a three-layer neural network to study adaptive learning for defuzzification. Halgamuge *et al.* [42] describe the CBADD neural network and the learning algorithm that is based on the gradient descent learning and that has proved the associated convergence algorithm. By using a learning network in CBADD, it acts as a universal defuzzification function approximator for a fuzzy rulebase.

8.3.3.6 Recent Advances

In this section we will briefly mention four interesting and different types of works that are being pursued in the area of defuzzification. The works mentioned below show that there is a growing interest to formalize and deeply understand the problem of defuzzification. Furthermore, it also confirms that the researchers have begun to address this problem from other viewpoints. An interested reader may look up the mentioned reference for more details:

- *Evolutionary defuzzification*: In recent years, the use of evolutionary algorithms [43] to tune the parameters used in the defuzzification algorithms has gained popularity as such methods contribute to higher accuracy. Such methods are known as evolutionary adaptive defuzzification (EAD) methods. See [44–47] for more details.
- *Area of compensation*: Fortemps and Roubens [48] have tried to explain the defuzzification process by estimating the mean value of a fuzzy number by computing the area of compensation. They have used the arithmetic of the expected values of the possibility and necessity distribution with respect to fuzzy numbers.
- *Steiner point defuzzification*: Recently, for a multidimensional feature vectors in R^n , Vetterlien and Navara [49] have seriously tried to understand the defuzzification in terms of Steiner points of the convex fuzzy sets.
- *Distribution averaging*: In [50], Roventa and Spircu have also used multidimensional fuzzy sets following an integral over the level sets. They call it defuzzification through distribution averaging. Their algebraic methods are close to the method of ranking of random variables in probability theory. Finally, their averaging procedure transforms a fuzzy set into a crisp set and then replaces the crisp set by a single value.

8.4 Conclusion

In this chapter we revisited the concepts of fuzzy encoding and decoding terms of fuzzification and defuzzification. Fuzzification and defuzzification have been difficult problems in fuzzy research, as there are no convincing reasoning of granulation and degranulation. Mostly researchers have devised their own home-grown tricks and methods to construct their fuzzifier and defuzzifier in order to solve their bigger problems.

In the fuzzification section we showed some functional operators and also discussed data-driven approaches that have been used in the literature. Regarding defuzzification we discussed standard approaches and other alternative methods that have been proposed. Since this is an overview we tried to cover most of related research till date for both the topics.

Acknowledgments

The author expresses gratitude to anonymous reviewers and editors for their constructive comments.

References

- [1] W. Pedrycz (ed.). *Granular Computing: An Emerging Paradigm*. Springer-Verlag, New York, 2001.
- [2] J.V. de Oliveira. On optimal fuzzy systems I/O interfaces. In: *Proceedings of Second IEEE International Conference on Fuzzy Systems*, San Francisco, CA, Vol. 2, 1993, pp. 851–856.
- [3] J.V. de Oliveira. A design methodology for fuzzy system interfaces. *IEEE Trans. Fuzzy Syst.* 3 (1995) 404–414.
- [4] W. Pedrycz and J.V. de Oliveira. Optimization of fuzzy models. *IEEE Trans. Syst. Man Cybern. Part B* 28 (1996) 627–636.
- [5] D. Dubois and H. Prade. The three semantics of fuzzy sets. *Fuzzy Sets Syst.* 90 (1997) 141–150.
- [6] I.B. Turksen. Measurement of membership functions and their acquisition. *Fuzzy Sets Syst.* 40 (1991) 5–38.
- [7] E. Hisdal. Are grades of membership probabilities? *Fuzzy Sets Syst.* 25 (1988) 325–348.

- [8] S. Medasani, J. Kim, and R. Krishnapuram. An overview of membership function generation techniques for pattern recognition. *Int. J. Approx. Reason.* 19 (1998) 391–417.
- [9] B.B. Devi and V.V.S. Sarma. Estimation of fuzzy memberships from histograms. *Inf. Sci.* 35 (1985) 43–59.
- [10] R. Viertl. Univariate statistical analysis with fuzzy data. *Comput. Stat. Data Anal.* 51 (2006) 133–147.
- [11] M.S. Chen and S.W. Wang. Fuzzy clustering analysis for optimizing membership functions. *Fuzzy Sets Syst.* 103 (1999) 239–254.
- [12] T.W. Liao, A.K. Celmins, and R.J. Hammell II. A fuzzy c -means variant for the generation of fuzzy term sets. *Fuzzy Sets Syst.* 135 (2003) 241–256.
- [13] J.J. Buckley. Fuzzify. *Fuzzy Sets Syst.* 73 (1995) 245–248.
- [14] G. Bortolan and W. Pedrycz. Reconstruction problem and information granularity. *IEEE Trans. Fuzzy Syst.* 5 (1997) 234–248.
- [15] C.C. Yang and N.K. Bose. Generating fuzzy membership function with self-organizing feature map. *Pattern Recognit. Lett.* 27 (2006) 356–365.
- [16] D. Dubois and H. Prade. Possibility theory and statistical reasoning. *Comput. Stat. Data Anal.* 51 (2006) 47–69.
- [17] M.H. Masson and T. Deneux. Inferring a possibility distribution from empirical data. *Fuzzy Sets Syst.* 157 (2006) 319–340.
- [18] S. Roychowdhury and W. Pedrycz. A survey of defuzzification strategies. *Int. J. Intell. Syst.* 16 (2001) 679–695.
- [19] D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer-Verlag, New York, 1993.
- [20] C.C. Lee. Fuzzy logic in control systems: Fuzzy logic controllers – Part I & II, *IEEE Trans. Syst. Man Cybern. SMC-20* (2) (1990) 404–435.
- [21] T.A. Runkler and M. Glesner. A set of axioms for defuzzification strategies towards a theory of rational defuzzification operators. *Proceedings of the Second IEEE International Conference Fuzzy Systems*, San Francisco, CA, 1993, pp. 1161–1166.
- [22] W. van Leewijck and E.E. Kerre. Defuzzification: criteria and classification. *Fuzzy Sets Syst.* 108 (1999) 159–178.
- [23] A. Patel and B. Mohan. Some numerical aspects of center of area defuzzification method. *Fuzzy Sets Syst.* 132 (2002) 401–409.
- [24] E. Van Broekhoven and B. De Baets. Fast and accurate center of gravity defuzzification of fuzzy system outputs defined on trapezoidal fuzzy partitions. *Fuzzy Sets Syst.* 157 (2006) 904–918.
- [25] S.S. Lancaster and M.J. Wierman. Empirical study of defuzzification. In: *Proceedings of 22nd International Conference of the North American Fuzzy Information Processing Society*, Chicago, IL: July 24–26, 2003, pp. 121–126.
- [26] H. Thiele. Towards axiomatic foundations for defuzzification theory. *Reihe Comput. Intell.* 31 (1998) 1–14.
- [27] B. Kosko. *Neural Networks and Fuzzy Systems*. Prentice Hall, Englewood Cliffs, NJ, 1992.
- [28] J.V. de Oliveira. A set theoretical defuzzification method. *Fuzzy Sets Syst.* 76 (1994) 63–71.
- [29] J.K. Kim, C.H. Cho, and H.L. Kwang. A note on the set-theoretical defuzzification. *Fuzzy Sets Syst.* 98 (1998) 337–341.
- [30] D.P. Filev and R.R. Yager. A generalized defuzzification method via BAD distributions. *Int. J. Intell. Syst.* 6 (1991) 687–697.
- [31] G.J. Klir. A principle of uncertainty and information invariance. *Int. J. Gen. Syst.* 17 (1990) 249–275.
- [32] D. Dubois and H. Prade. Fuzzy sets and probability: misunderstandings, bridges, and gaps. In: *Proceedings of the Second IEEE International Conference on Fuzzy Systems*, San Francisco, CA, Vol. 2 1993, pp. 1059–1068.
- [33] R.R. Yager and D.P. Filev. SLIDE: a simple adaptive defuzzification method. *IEEE Trans. Fuzzy Syst.* 1 (1993) 69–78.
- [34] R.R. Yager and D.P. Filev. *Essentials of Fuzzy Modeling*. John Wiley & Sons, New York, 1994.
- [35] R.R. Yager and D.P. Filev. Constrained defuzzification. In: *Proceedings of the Fifth IFSA World Congress*, Seoul, Korea, 1993, pp. 1167–1170.
- [36] S. Roychowdhury and B.H. Wang. Cooperative neighbors in defuzzification. *Fuzzy Sets Syst.* 78 (1994) 37–49.
- [37] S. Roychowdhury, B.H. Wang, and S.K. Ahn. Radial defuzzification. *Int. J. Gen. Syst.* 28 (1999) 201–225.
- [38] H. Genther, T.A. Runkler, and M. Glesner. Defuzzification based on fuzzy clustering. In: *Proceedings of the Third IEEE Conference on Fuzzy Systems*, Vol. 3, Orlando, FL, June 26–29, 1994, pp. 1645a, 1646–1648.
- [39] S.K. Halgamuge and M. Glesner. Neural networks in designing fuzzy systems for real world application. *Fuzzy Sets Syst.* 65 (1994) 1–12.
- [40] Q. Song and G. Bortolan. Some properties of defuzzification neural networks. *Fuzzy Sets Syst.* 61 (1994) 83–89.
- [41] Q. Song and R.P. Leland. Adaptive learning defuzzification techniques and applications. *Fuzzy Sets Syst.* 81 (1996) 321–329.

- [42] S.K. Halgamuge, T.A. Runkler, and M. Glesner. On neural defuzzification networks. *Proceedings of the IEEE International Conference on Fuzzy Systems*, New Orleans, LA, 1996, pp. 463–469.
- [43] T. Back, D. Fogel, and Z. Michalewicz. *Handbook of Evolutionary Computation*. Oxford University Press, Oxford, 1997.
- [44] O. Cordon, F. Herrera, F.A. Márquez, and A. Peregrin. A study on the evolutionary adaptive defuzzification methods in fuzzy modelling. *Int. J. Hybrid Intell. Syst.* 1 (2004) 36–48.
- [45] D. Kim, Y. Choi, and S. Lee. An accurate COG defuzzifier design using Lamarckian co-adaptation of learning and evolution. *Fuzzy Sets Syst.* 130 (2002) 207–225.
- [46] C.Z. Hong. Handwritten numeral recognition using a small number of fuzzy rules with optimized defuzzification parameters. *Neural Netw.* 8 (1995) 821–827.
- [47] T. Jiang and Y. Li. Generalized defuzzification strategies and their parameter learning procedures. *IEEE Trans. Fuzzy Syst.* 4 (1996) 64–71.
- [48] P. Fortemps and M. Roubens. Ranking and defuzzification methods based on area compensation. *Fuzzy Sets Syst.* 82 (1996) 319–330.
- [49] T. Vetterlien and M. Navara. Defuzzification using steiner points. *Fuzzy Sets Syst.* 157 (2006) 1455–1462.
- [50] E. Roventa and T. Spircu. Averageing process in defuzzification process. *Fuzzy Sets Syst.* 136 (2003) 375–385.

9

Systems of Information Granules

Frank Höeppner and Frank Klawonn

9.1 Introduction

Collecting large amounts of data has become a routine in business, industry, and science nowadays. However, in order to handle large amounts of raw data and extract and present the inherent information, techniques are required that provide compressed and compact representations of the aspects of interest. In the extreme case a collection of measurements might be represented by a single value, for instance, the mean or the median. Although at least one of the crucial characteristics of the data can be covered by a mean or median, such simplified concepts loose by far too much of the information contained in the data to be of practical use in applications.

The two cases – the raw data and a single value in the form of the mean or median – can be viewed as two extremes of granularities to represent the data. By treating all data items as single and separate objects, the finest possible granularity, loss of information is avoided for the price of difficulties in handling the data and the loss of interpretable compressed representations. On the other hand, a mean or median is easy to understand, but loses most of the information contained in the data. The truth lies somewhere in between these extremes. In order to handle, represent, and manage the data in an understandable and efficient way, information granules are introduced. An information granule – or simply granule – is a conceptual notion that has instances expressed in a data set [1–3]. In the simplest case, a granule is a subset of the data, for instance, described by an interval. A typical form of such crisp granules results from a discretization of a continuous variable. Instead of considering exact values, ranges are introduced, i.e., the domain, for instance, the interval $[a, b]$ partitioned into k subintervals $[a, x_1], [x_1, x_2], \dots, [x_{k-1}, b]$, where $a < x_1 < x_2 < \dots < x_{k-1} < b$.

Although in some cases, there might be a canonical way to choose the boundaries x_i for the intervals, such a discretization often causes problems, since values close to a boundary of one of the intervals, but lying on different sides of the boundary, belong to different granules, although their difference might be extremely small. In order to avoid such problems, a reasonable approach is to give up the idea that an object must either belong or not belong to a granule. This leads to granules in the form of fuzzy sets or probability distributions. Also overlapping crisp granules might be considered, leading to rough sets [4] – an approach that will not be considered in this chapter.

An information granule is more than just a collection or set of elements. It should be describable by a specific property or concept as in the example of discretization where a granule is defined by the lower and upper bound of the corresponding interval.

The choice of appropriate granules strongly depends on the specific purpose and application. In the fields of business intelligence and data warehouses, online analytical processing (OLAP) [5] exploits

granules in a canonical way. OLAP provides views on a data set based on different levels of refinements. Most attributes handled within OLAP are categorical with an additional hierarchical structure. This means that the attribute can take values from a finite, refinable domain. For instance, an attribute representing a location might have different levels of refinement, like country, state, region, and town. Or an attribute for time might be refinable from year to quarters, months, and days. As in this example, for categorical attributes there is often an obvious way to refine or coarsen them, defining granules directly. This does not usually apply to attributes with a continuous domain. Notions like magnitude are good candidates for granules in the case of continuous attributes. The crucial problem is to define the appropriate granules suited best to the data and to the underlying task or purpose. Therefore, this chapter will focus exclusively on continuous attributes.

The purpose of the granules can be simply to describe important or interesting patterns or substructures in the data. A typical application is subgroup discovery [6], where the aim is to find granules in which a certain class of objects is significantly over- or underrepresented compared with the overall distribution of the class in the data set. However, in most applications it is important to cover more or less the whole data set by granules. In order to avoid redundant information, the granules should not overlap or at least not too much. In this sense, the granules should roughly represent a partition, not necessarily in the strict mathematical sense.

With this motivation in mind, the chapter is organized as follows. Section 9.2 provides an overview of the possible contexts and purposes in which granularization is of interest. Section 9.3 discusses useful and important properties of partitions induced by granules. Section 9.4 is devoted to a short introduction to cluster analysis, one of the main techniques to construct granules from data in an unsupervised context. Many similarity-driven approaches to granularization, which are described in Section 9.5, rely on clustering methods. Finding granules in a supervised learning context is outlined in Section 9.6 before the summary in the conclusion section.

9.2 Purpose of Granularization

The general problem of finding granules in the context considered here can be described as follows. Given a data set or a universe of discourse $D \subset \mathbb{R}^p$, how can we cover or partition this domain by a finite set of meaningful concepts?

First of all, the overall purpose of the granularization should be specified. In the case of exploratory data analysis, where one of the main aims is to discover interesting or meaningful structures in the data set, the granules are used for compact descriptions, groupings, or segmentations of objects or data. Typically, the granules are found on the basis of a similarity or distance measure. The identification of meaningful substructures without specifying in advance which objects should belong to a certain substructure is called unsupervised learning.

Supervised learning refers to a context where the prediction of an attribute based on other attributes is the final goal. Regression refers to the case when the attribute to be predicted is a numerical one, whereas the term classification is used for problems where the attribute to be predicted is of categorical nature. In other words, in supervised learning we want to describe a classification or regression function $f : D \rightarrow Y$, where Y is a finite set in the case of classification and $Y \subseteq \mathbb{R}$ in the case of regression. The granules are involved in the description of the function f . Nevertheless, the task of finding suitable granules is only indirectly supervised in the sense that the granules should support the definition of a suitable function f as much as possible. The granules are found by optimizing f . In this sense, the identification of the granules is task driven, guided by the specific classification or regression problem.

It should be noted that, even in the context of supervised learning, it is sometimes popular to identify granules in an unsupervised manner, ignoring the attribute to be predicted. After the granules are identified in this purely unsupervised way, they are then used for the purpose of supervised learning. However, there is no reason why granules, derived for descriptive purposes, should also perform well in a specific prediction task.

Granules can be defined by a human expert or determined automatically on the basis of an available data set. When the granularization is carried out completely by a human expert, a suitable formal language,

model, or tool to describe the granules is required. This will not be the focus of this contribution. We will only provide guidelines that should be taken into account when designing granules as well as an overview of techniques to determine granules automatically on the basis of data.

9.3 Properties of Partitions

As mentioned before, we are interested not only in isolated granules, but in granules that can cover the domain of interest roughly in terms of a partition. There are many ways of organizing a partition; some of the aspects that may be taken into account will be addressed in this section. All these aspects or properties influence both the suitability of the partition for a given task and the interpretability, so it depends on the application and the purpose of the granularization (see Section 9.2) which of these properties a partition should possess.

9.3.1 Degree of Uncertainty

The traditional notion of a partition is the following: Given a data set $D = \{x_1, \dots, x_n\}$ and a number of subsets $C_i, i = 1 \dots c$, the system of sets $\{C_i | i = 1 \dots c\}$ is called a partition if the following properties hold: the elements of a partition are non-empty, pairwise disjoint, and cover the whole data set:

$$\forall i \in \{1, \dots, c\} \quad C_i \neq \emptyset \quad (1)$$

$$\bigcup_{i=1 \dots n} C_i = D; \quad (2)$$

$$\forall i, j \in \{1, \dots, c\} \quad C_i \cap C_j = \emptyset. \quad (3)$$

Any element of D belongs to exactly one element of the partition; there is no ambiguity about the data–granule relationship. Although convenient from a mathematical point of view, many concepts in the real world are not that strict. Sometimes an object may belong to two granules equally likely. In this case, condition (3) is removed and we speak of an *overlapping partition*.

As soon as uncertainty comes into play, one may want to express the degree of uncertainty in a more expressive way. We reformulate the above-mentioned properties by a characteristic function $\chi_{C_i} : D \rightarrow \{0, 1\}$, where $\chi_{C_i}(x_j) = 1 \Leftrightarrow x_j \in C_i$. Rather than considering a binary decision $\chi_{C_i}(x_j) \in \{0, 1\}$ we then consider the belongingness to C_i as a matter of degree – as a fuzzy set determined by a membership function $\mu_{C_i} : D \rightarrow [0, 1]$ with the unit interval as its range.

The definition of a *fuzzy partition* is not as obvious as the generalization of the notion of a crisp set to the concept of a fuzzy set. In most practical applications, where fuzzy sets are used as granules to describe a domain, these fuzzy sets satisfy the condition that they are disjoint (cf. (3)) with respect to the Lukasiewicz t -norm. The Lukasiewicz t -norm is defined by $\alpha \odot \beta = \max\{\alpha + \beta - 1, 0\}$ and can be viewed as one of many possible choices for a $[0, 1]$ -valued conjunction or intersection operation. Two fuzzy sets μ_1 and μ_2 are disjoint with respect to the Lukasiewicz t -norm if $\mu_1(x) + \mu_2(x) \leq 1$ holds for all x in the considered domain.

The covering property (2) is very often satisfied with respect to the dual t -conorm to the Lukasiewicz t -norm, the bounded sum defined by $\alpha \oplus \beta = \min\{\alpha + \beta, 1\}$. In other words, the sum of membership degrees should add up to at least 1 for all elements in the considered domain.

Figure 9.1 shows a typical fuzzy partition that satisfies the disjointness property with respect to the Lukasiewicz t -norm and the covering property with respect to the bounded sum. These two conditions will be satisfied by any fuzzy partition on an interval where the membership degrees of two neighboring fuzzy sets always add up to 1 and at most the supports of two fuzzy sets overlap in every point.

Fuzzy partitions can be defined either on the domain or on the considered data. In the latter case, the membership degrees of the data objects x_1, \dots, x_n to the granules C_1, \dots, C_c can be defined by a membership matrix $U = [u_{i,j}]_{j=1 \dots n, i=1 \dots c}$. The above-mentioned disjointness and covering conditions translate in this case to the condition $\sum_{i=1}^c u_{i,j} = 1$ for all $j = 1, \dots, n$. This equation can also be used

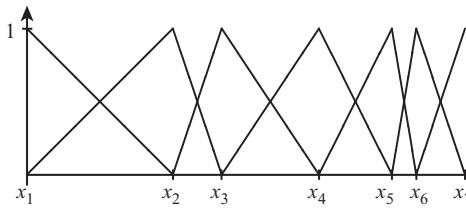


Figure 9.1 A typical fuzzy partition

to replace the two conditions (2) and (3) for crisp partitions on finite domains, where only $u_{i,j} \in \{0, 1\}$ is allowed. In this case we assume $u_{i,j} = 1 \Leftrightarrow x_j \in C_i$.

There might be various reasons for introducing uncertainty in the partition. Uncertainty may already be contained in the domain, for instance, in the form of noisy measurements or vague concepts that are used to describe the granules. Another reason might be to avoid sharp boundaries between the granules, when the belongingness of data to granules is used in further processing. For instance, when each granule is associated with a function in terms of a local model, then switching from one granule to another one will result in discontinuous behavior in the output function unless the corresponding functions fit to each other at the boundaries of the granules. In order to avoid this effect and achieve smooth switching between the local models, membership degrees can be taken into account to combine the local models according to the membership degrees.

The condition $\sum_{i=1}^c u_{i,j} = 1$ can be seen as a probabilistic constraint for the partition. The membership degree $u_{i,j}$ could be interpreted as the probability that object x_j belongs to the granule c_i . However, even if this probabilistic interpretation is intended, granules are seldomly interpreted in terms of probability distributions. Mixture models [7] are a very popular probabilistic approach in clustering and classification. But the resulting distributions might look as shown in Figure 9.2. The two Gaussian distributions might be used for classification in order to distinguish between two classes. Both distributions might have the same a priori probability .5. In this case an object in the form of a real number would be assigned to the Gaussian distribution that yields the higher likelihood. This means that values far away from zero will be assigned to the flatter Gaussian distribution drawn with a dotted line and values closer to zero will be assigned to the other Gaussian distribution. It is obvious that these two Gaussian distributions do not represent useful granules, since they overlap too much. Therefore, we will not consider probability distributions as granules in this chapter.

Even though the probabilistic constraint is also very common for fuzzy partitions, it can be dropped and generalized to a possibilistic framework [8]. In this case, additional steps have to be taken into account in order to guarantee a partition-like set of granules.

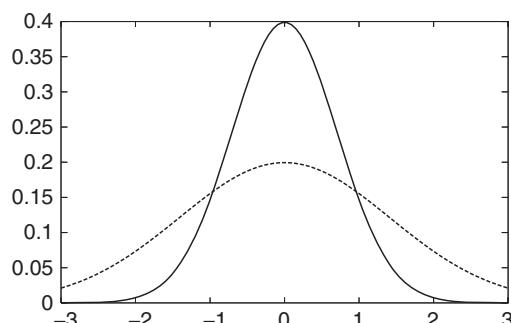


Figure 9.2 Two Gaussian distributions

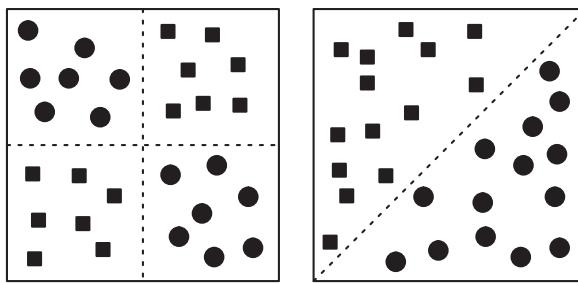


Figure 9.3 Cases for two-dimensional granules

9.3.2 Multidimensional and Hierarchical Granules

The considered domain for which the granules have to be defined is usually a multidimensional space, typically a subset of \mathbb{R}^n . A crucial problem is the description of multidimensional granules. A very intuitive approach consists in defining granules or partitions for the single dimensions and then combining these granules to describe multidimensional granules. In this case an implicit independence assumption is made. Independence should not be interpreted in the sense of probability theory, but in more general terms. When the granules are based on similarity concepts, independence means that there is no interaction between the similarities. In order to understand this effect, let us consider the two simple examples shown in Figure 9.3. The aim is in both cases a classification problem where the circles should be separated from the squares. In both cases, looking at the projections to the single dimensions will result in a complete loss of information for the separation of the two classes circle and square. Nevertheless, the example given on the left side in Figure 9.3 can still be treated easily by defining suitable granules on the single dimensions. When each of the dimensions is partitioned into two granules as indicated by the dotted lines, then it is easy to use these one-dimensional granules to define four two-dimensional granules that will either contain only circles or contain only squares.

However, the example on the right side in Figure 9.3 is difficult to treat with combinations of one-dimensional granules. Of course, with a sufficient number of one-dimensional granules the dotted separating line can be approximated with combinations of one-dimensional granules. But this is a typical case where an independent consideration of the single dimension does not provide a helpful insight into the two-dimensional problem.

Another problem in granularization is the number of granules or how coarse or how fine the granularization should be chosen. In principle, it is possible to consider independently different levels of granularization. However, this will make it more difficult to switch between the different levels of granularization. Therefore, it is recommended to choose hierarchical granules or partitions when different levels of granularization are required. Such hierarchical partitions might be derived in a canonical way as for instance in the case of OLAP applications or they might be found by hierarchical clustering techniques [9].

Another popular approach is the concept of rough sets, where granules are organized hierarchically [4, 10]. Having the notion of *granule inclusion* available, the human way of focusing and generalizing can then be emulated quite naturally. The inclusion in a lower approximation of a granule is defined formally as the satisfaction of additional constraints. Approaches to learn hierarchical granules are often quite similar to hierarchical clustering algorithms.

9.4 Brief Introduction to Clustering

The problem of clustering is that of finding a partition that captures the similarity among data objects by grouping them accordingly in the partition. Data objects within a group (element of the partition) should be similar, while data objects from different groups should be dissimilar. It is immediately clear

Table 9.1 Types of clustering algorithms

Type	Approach	Algorithm
Linkage	Iteratively merge data objects with their respective closest cluster	Hierarchical clustering [9, 11]
Density	Identify connected regions where the data density exceeds some thresholds	DBScan [12], Clique [13]
Objective function	Perform clustering by minimizing an objective function that, e.g., minimizes the sum of within-cluster distances	c-Means [14], FCM [15, 16]

that clustering requires an explicit notion of similarity, which is often provided by means of a distance (or dissimilarity) function $d : D \times D \rightarrow \mathbb{R}_+$ or matrix $\mathcal{D} \in \mathbb{R}_+^{n \times n}$, $|D| = n$. A small value indicates that two objects are similar, while a large value indicates that they are dissimilar.

There are several types of clustering algorithms and many algorithms of each type available in the literature. (Table 9.1 shows only some representatives.) The choice of the clustering algorithm depends in the first place on how the dissimilarity information is provided. If only a dissimilarity matrix is given, relational clustering algorithms have to be used. Usually, this approach can be used only if the entities we are going to cluster are only a few, because the space (and time) complexity is at least quadratic. (The distance matrix contains a distance value for each pair of entities.) Whenever the data are numerical in nature, dissimilarity *functions* such as the Euclidean distance can be utilized and there is no need for storing n^2 distances explicitly. Most clustering algorithms assume that such a distance function is given.

Here, we briefly discuss only the *c*-means and fuzzy *c*-means (FCM) clustering algorithm; for the other techniques we refer to the literature [9, 17–19]. The *c*-means algorithm [14] partitions the available data into c groups by finding c *prototypical data objects* that best represent the whole data set. Given a prototype, all data objects that are closer to this prototype than to any other form the cluster or group of this prototype. The algorithm iteratively performs the following two steps: First, for all data objects, find the closest prototype (which is initialized randomly at the beginning). Second, adjust each of the prototypes to better represent the group by moving it to the center of gravity of the group. If we encode the belongingness of data object x_j to prototype p_i in a binary variable $u_{i,j}$, which is 1 if and only if x_j is closest to p_i (0 otherwise), then the *c*-means algorithm actually minimizes

$$J_1(U, P; X) = \sum_{j=1}^n \sum_{i=1}^c u_{i,j} \|x_j - p_i\|^2 \quad (4)$$

by alternately minimizing J with respect to $U = [u_{i,j}]$ and $P = (p_1, \dots, p_c)$.

The algorithm considers the belongingness of a data object to a prototype as a binary decision, giving it the flavor of a combinatorial problem. The fuzzy *c*-means variant of *c*-means [15, 16] turns the problem into a continuous one by making the cluster membership a matter of degree ($u_{i,j} \in [0, 1]$ rather than $u_{i,j} \in \{0, 1\}$; cf. Section 9.3.1). Despite the relaxation, the minimization of (4) would still lead to crisp membership degrees; therefore, an exponent m for the membership degrees is introduced, the so-called *fuzzifier*. (For a detailed discussion of the effect of the fuzzifier and alternative approaches, see [20].) Thus, the fuzzy *c*-means algorithm minimizes

$$J(U, P; X) = \sum_{j=1}^n \sum_{i=1}^c u_{i,j}^m \|x_j - p_i\|^2 \quad (5)$$

subject to the constraints $\sum_{i=1}^c u_{i,j} = 1$ and $\sum_{j=1}^n u_{i,j} > 0$. The necessary conditions for a minimum of (5) w.r.t. p_i (assuming $u_{i,j}$ to be constant) yields the same update equations as in the *c*-means

Table 9.2 The FCM algorithm

Choose $m > 1$ (typically $m = 2$)
 Choose termination threshold $\varepsilon > 0$
 Initialize prototypes p_i (randomly)
 Repeat
 update memberships using (7)
 update prototypes using (6)
 until change in memberships drops below ε

algorithm: the prototypes have to be shifted to the (weighted) center of all data points in the group:

$$\forall 1 \leq i \leq c : \quad p_i = \frac{\sum_{j=1}^n u_{ij}^m x_j}{\sum_{j=1}^n u_{ij}^m}. \quad (6)$$

The minimization w.r.t. $u_{i,j}$ (assuming p_i to be constant) delivers

$$u_{ij} = \begin{cases} \frac{1}{\sum_{l=1}^c \left(\frac{\|x_j - p_l\|^2}{\|x_j - p_l\|^2} \right)^{\frac{1}{m-1}}} & \text{in case } I_j = \emptyset \\ \frac{1}{|I_j|} & \text{in case } I_j \neq \emptyset, i \in I_j, \\ 0 & \text{in case } I_j \neq \emptyset, i \notin I_j \end{cases} \quad (7)$$

where $I_j = \{k \in \mathbb{N}_{\leq c} \mid x_j = p_k\}$. The FCM algorithm is depicted in Table 9.2. For a more detailed discussion of FCM and examples we refer to the literature (e.g., [19]).

If a hierarchical partition is desired, it can be obtained quite easily by recursively applying a clustering algorithm to each of its clusters. In the case of a fuzzy partition the objective function-based clustering algorithms can easily be modified to consider the degree of belongingness to the clusters by an additional weight (see, e.g., [21]). In [22] granules are represented by hyperboxes and these granules are then themselves clustered in a hierarchical fashion.

Almost all clustering algorithms deliver a partition regardless of the data set provided; i.e., c -means yields c prototypes even in case of a uniform distribution without any cluster substructure. Thus, once a partition has been obtained by a clustering algorithm, the question is whether it represents significant structural information or just random fluctuations. In the literature, one can find many validity measures (see, e.g., [23]), that address global properties of the partition (e.g., degree of ambiguity) or local properties of each individual cluster (e.g., separation and compactness). Validity measures are also used to determine the number of clusters in c -means or FCM clustering.

There are two important properties of the FCM algorithm: (1) it derives fuzzy partitions (i.e., allows overlapping granules, which is much more realistic in many applications) and (2) the possibility of modifying the objective function allows us to force FCM quite easily to consider additional aspects in the resulting partition. If the purpose of granularization is purely descriptive in nature, granularization and clustering become almost indistinguishable. However, as we have seen in Section 9.2, quite often the partition must also fit in a certain task or application and/or must be easily interpretable. The fact that FCM can be tailored to specific applications by changing the objective function makes FCM a good candidate also for task-driven granularization purposes.

9.5 Similarity-Induced Partitions

In this section we consider the problem of deriving a system of granules in an unsupervised setting; e.g., the only information available to develop the granules is the data itself; we cannot utilize any feedback in the form of a classification or an approximation error to guide the process. If such information is

available, Section 9.6 should be consulted. In an unsupervised context we cannot measure the fitness for a specific purpose, so the granules should at least be representative for the data from which the granules have been induced.

9.5.1 One-Dimensional Granules with Crisp Boundaries

Let us first consider the case of a single numerical attribute. It is a common practice in many data-mining applications that the set of values is discretized in an equiwidth or equifrequency fashion, thereby obtaining a reduced number of c intervals rather than (up to) n individual values. In the equiwidth approach, all intervals share the same width. To define such a partition we only need to know the minimum and maximum value and set the number of intervals c a priori. Such a set of intervals hardly qualifies as a system of granules, because it does not reflect the underlying data distribution. In the equifrequency approach each granule covers the same number of data objects: having sorted all instances we introduce a new interval every $\frac{n}{c}$ values. Figure 9.4 shows the effect of both approaches in an example data set of 20 values. An equiwidth partition ($c = 5$) is shown in Figure 9.4a, the intervals do not reflect the data distribution; the fourth interval does not contain any instances. Figure 9.4b shows the case of an equifrequency partition ($c = 4$). Again, the partition does not characterize the underlying data: The values near the middle are separated into two different intervals, although they form a compact group of similar values. In contrast, the results of clustering algorithms lead to much more meaningful partitions, as shown in Figures 9.4c and 9.4d. Density-based approaches identify those instances that do not reach a desired data density as outliers; therefore, in Figure 9.4c the granules do not cover the complete range of values. Clustering algorithms that identify prototypical values, e.g., c -means, can be used to define granules, too, by introducing boundaries halfway between the prototypes, as shown in Figure 9.4d.

If the attribute under consideration is of a categorical scale, there is no information whatsoever to group the data by similarity – except in the case that an expert provides additional information by means of a similarity matrix. If such a matrix is given, relational clustering algorithms such as the single-linkage hierarchical clustering algorithm [11] can be used to obtain a partition. If the scale is ordinal, we have two options: either to provide a distance matrix and use relational clustering or to transform the attribute into a numerical one by assigning numbers to the ordinal values and proceed as before with numerical values.

9.5.2 One-Dimensional Granules with Fuzzy Boundaries

The systems of information granules shown in Figures 9.4c and 9.4d also illustrate the unsatisfying consideration of outliers by crisp partitions. The two instances between the leftmost partitions may be considered as outliers (not belonging to any granule) or as equally poor representatives of both the leftmost granules. It may, however, easily happen that these two instances are assigned to different granules if a crisp assignment is required (as in Figure 9.4d). The uncertainty in the assignment to a granule is better

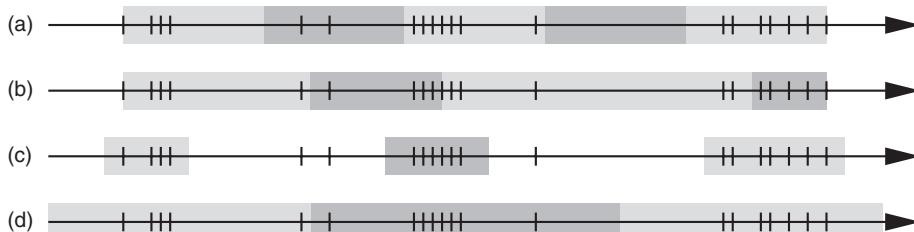


Figure 9.4 Interval partitions (shaded regions) for a numerical attribute obtained through (a) equiwidth partitioning, (b) equifrequency partitioning, (c) density-based clustering, and (d) prototype-based clustering

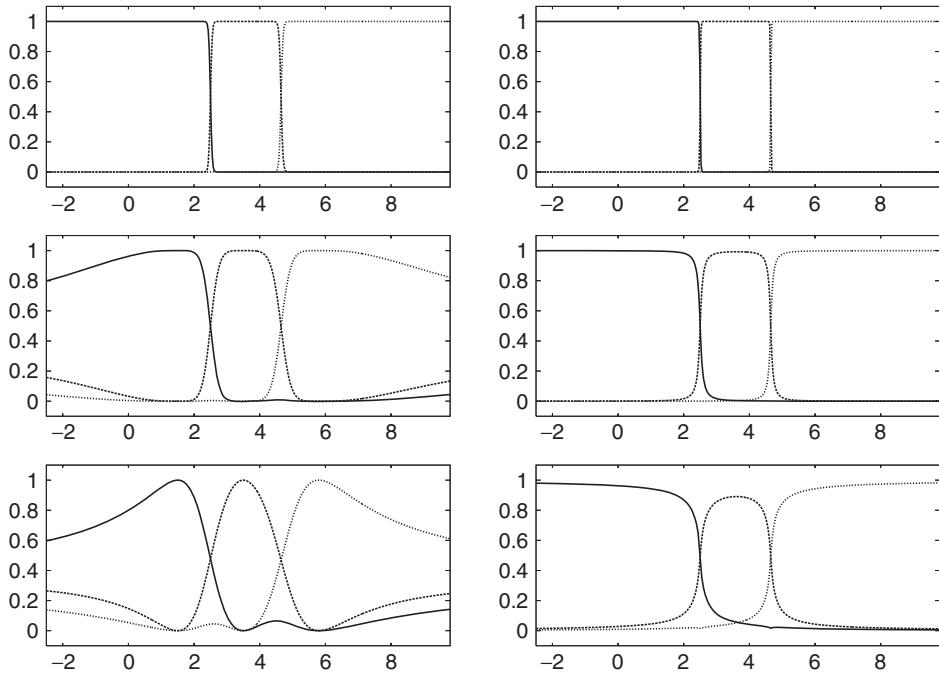


Figure 9.5 Fuzzy membership functions. From top to bottom the fuzziness is increased (m resp. η); left column: original FCM and right column: modified FCM

captured when fuzzy memberships are used, because then a smaller membership degree indicates a less confident assignment to a granule.

The most simple way to turn the interval granules from Figure 9.4d into fuzzy granules is by replacing the c -means algorithm with the FCM algorithm. The fuzzy membership functions of FCM are given analytically by (7) and the only parameters are the cluster prototypes. Once the prototypes are determined, the membership functions can serve as information granules. The left column of Figure 9.5 shows the fuzzy granules for three different degrees of fuzziness. The fuzzier the granules get, the faster the membership functions approach the value $\frac{1}{c}$ at the boundaries of the domain. This is because the farther we are from the prototypes, the less certain we can be that a value belongs to a particular prototype. On the other hand, if we regard the introduction of fuzzy memberships as a fuzzification of the crisp intervals in Figure 9.4d, the degradation near the borders seems counterintuitive.¹ From that point of view, granules like those on the right-hand side of Figure 9.5 are preferable.

This kind of membership function can be obtained by a modification of the FCM objective function. The motivation for the modification proposed in [24] is to avoid the unexpected local maxima (cf. bottom-left plot in Figure 9.5) and the completely fuzzy membership degrees near the borders by rewarding crisp membership degrees. Only where we have to switch between the cores of the sets, fuzziness is pertained (cf. right-hand side of Figure 9.5). We introduce a number of parameters $a_j \in \mathbb{R}_{\geq 0}$, $1 \leq j \leq n$, and consider the following modified objective function:

$$J(U, P; X, A) = \sum_{j=1}^n \sum_{i=1}^c u_{i,j}^2 d^2(x_j, p_i) - \sum_{j=1}^n a_j \sum_{i=1}^c \left(u_{i,j} - \frac{1}{2} \right)^2. \quad (8)$$

¹ In case of a single attribute, this problem affects two granules at most; however, this is no longer true when seeking for granules in high-dimensional spaces (Section 9.5.3).

Only the second term is new in the objective function; the first term is identical to the objective function of FCM with $m = 2$. If a data object x_j is clearly assigned to one prototype p_i , then we have $u_{i,j} = 1$ and $u_{k,j} = 0$ for all other $k \neq i$. For all these cases, the second term evaluates to $-\frac{a_j}{4}$. If the membership degrees become more fuzzy, the second term increases. Since we seek to minimize (8), this modification rewards crisp membership degrees. The maximal reward we can give to obtain positive membership degrees is then

$$a_j = \min d_{*,j}^2 = \min\{d_{i,j}^2 \mid i \in \{1, \dots, c\}\} - \eta, \quad (9)$$

where $\eta > 0$ is a constant that takes a similar role as the fuzzifier. The resulting membership functions are

$$u_{i,j} = \frac{1}{\sum_{k=1}^c \frac{d_{i,j}^2 - \min d_{*,j}^2}{d_{k,j}^2 - \min d_{*,j}^2}}. \quad (10)$$

Since the notion of a cluster prototype itself is not affected by this modification, the prototype update equations remain the same. The resulting algorithm finds the fuzzy partition by performing a three-stage alternating optimization, consisting of

1. the calculation of the membership degrees via (10), assuming prototypes and a_j as being constant,
2. the calculation of the prototypes via (6), assuming memberships and a_j as being constant, and
3. the calculation of the a_j via (9), assuming prototypes and memberships as being constant.

Figure 9.5 illustrates the resulting membership functions in the one-dimensional case. Interestingly, even the multidimensional membership functions can be interpreted in a very intuitive manner. Given a set of prototypes, the Voronoi cell of a prototype p is the set of all points that are closer to p than to any other prototype. It turns out [24] that the distance term $d_{i,j}^2 - \min d_{*,j}^2$ in the membership functions corresponds to a (scaled) distance to the Voronoi cell of the respective prototype. This is illustrated for the two-dimensional case shown in Figure 9.6. The fact that the original FCM uses a *squared* Euclidean distance to the prototype, whereas this modification uses a scaled, but *unsquared*, distance to the Voronoi cell of the prototype, makes it less sensitive to outliers.

If the application requires the same number of instances per granules just as with the equifrequency partitions, FCM can also be biased toward uniformly sized clusters [25].

9.5.3 Multidimensional Granules

If we have to deal with multiple variables, they may be granularized individually and independently as described in the previous section. Will the granules obtained from the marginal data distributions

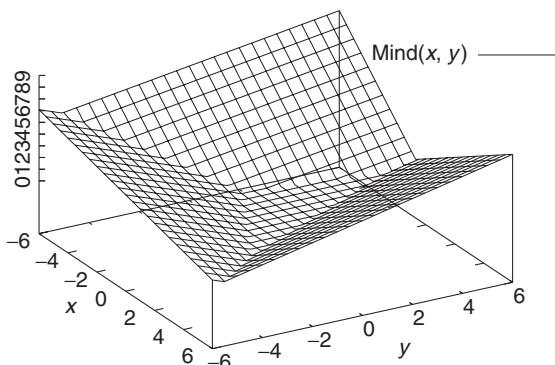


Figure 9.6 Distance to Voronoi cell

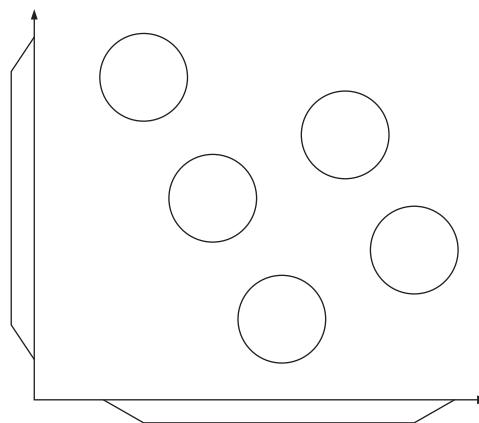


Figure 9.7 Granularization of single attributes does not provide a good starting point for granularization in the two-dimensional space

automatically be qualified to compose the granules in the higher dimensional space? Figure 9.7 shows that this is not true in general (see also Section 9.3.2). The circles in the two-dimensional plane shall represent regions of similar data; each circle may be considered as a granule we seek to discover. In this particular example, the data regions slightly overlap in the projection on each individual variable, such that an almost uniform marginal distribution is perceived (indicated along the axes). Given such a uniform distribution, we will obtain arbitrary one-dimensional granules and it is not very likely that we can describe the two-dimensional granules by a cross-product of one-dimensional granules. In this particular case, a clustering-based partition may even detect that there are no subclusters at all and thus a single granule will be sufficient. Although we have seen that the equiwidth partitioning (Section 9.5.1) hardly qualifies as a system of granules, in this particular example equiwidth partitioning may actually outperform clustering-based partitioning. Some approaches to granularization therefore take a multidimensional equiwidth partition as an initial solution and merge adjacent intervals if their (multidimensional) data distribution is identical [26]. In this bottom-up approach an initial fine granulated grid is iteratively coarsened, unless no more simplifications are eligible.

From a clustering perspective, the example shown in Figure 9.7 suggests to invert the approach: rather than developing the granules in the low-dimensional space and trying to approximate the high-dimensional granules by a product of low-dimensional granules, we could find the high-dimensional granules and extract low-dimensional granules from them (top-down). This can be easily done by applying clustering algorithms directly to the high-dimensional space.² The disadvantage of this approach is that different multidimensional granules may lead to *similar* projections in the one-dimensional space. These heavily overlapping granules should be identified and merged to obtain an understandable one-dimensional granularization (e.g., [27]). In such an approach, where only the projection of multidimensional granules will be processed further, it is advantageous to support merging by appropriate clustering algorithms that align the clusters along the main axes only [28] or seek for clusters in the shape of hyperboxes [22].

An alternative solution to this problem can also be found in a modified FCM algorithm [29]: rather than seeking for c independent, multidimensional granules, we may look for one-dimensional partitions of c_i granules each, such that the multidimensional space is tessellated into $\prod_i c_i$ granules. The prototypes are no longer optimized individually, but the grid of regularly distributed prototypes is optimized as a whole. During optimization the aim is to find optimal multidimensional granules, but in order to adjust the multidimensional granules we are limited to the modification of the one-dimensional partitions.

² We are simplifying at this point, because many clustering algorithms themselves depend on the set of attributes. If different subsets of attributes are used, different clusters will be found.

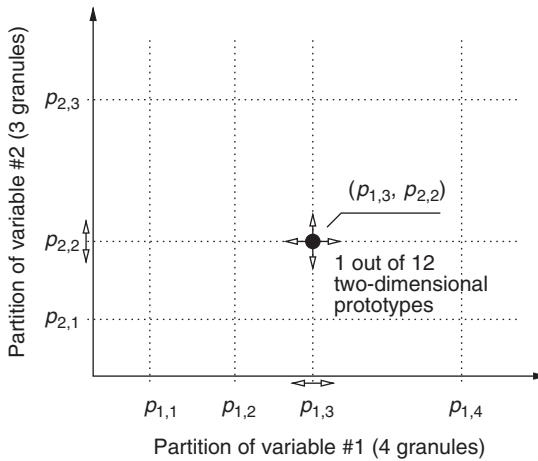


Figure 9.8 Illustration of the notation used in Section 9.5.3: $d = 2$ dimensions, first dimension consists of $c_1 = 4$ granules and second of $c_2 = 3$ granules. The multidimensional granules are composed of one-dimensional granules and are denoted by a tuple

Given d input variables, suppose we divide the domain of variable v_i into c_i granules, induced by representatives $p_{i,j}$, $i \in \{1, \dots, d\}$, $j \in \{1, \dots, c_i\}$, as illustrated by the example in Figure 9.8. We construct multidimensional granules by combining the one-dimensional granules $p_{i,j}$ (one for each dimension i) and denote them by a tuple $(p_{1,i_1}, p_{2,i_2}, \dots, p_{d,i_d})$, where $i_k \in \{1, \dots, c_k\}$.

The granules of interest are the multidimensional ones, so our cluster prototypes live in the multidimensional space and their total number is $c = \prod_{k=1}^d c_k$. But in contrast to the traditional FCM algorithm, where each of the prototypes is optimized individually, our prototypes are constrained to lie on the grid defined by the one-dimensional partitions. So the set of prototypes is given by

$$P = \{(p_{1,i_1}, p_{2,i_2}, \dots, p_{d,i_d})^\top \mid i_k \in \{1, \dots, c_k\}\}.$$

The standard FCM objective function remains unchanged, but has to take the construction of the prototypes into account:

$$J = \sum_{j=1}^n \sum_{i_1=1}^{c_1} \sum_{i_2=1}^{c_2} \dots \sum_{i_d=1}^{c_d} u_{(i_1, i_2, \dots, i_d), j}^m \left\| \begin{pmatrix} x_1 - p_{1,i_1} \\ x_2 - p_{2,i_2} \\ \vdots \\ x_k - p_{k,i_k} \end{pmatrix} \right\|^2 \quad (11)$$

(still subject to the same constraints on the membership degrees as before). The multidimensional granules are parameterized by the one-dimensional ones, so to optimize this objective function we have to solve for each of the $p_{i,j}$ individually. That is, the large set of $c = \prod_{k=1}^d c_k$ multidimensional granules is parameterized by a much smaller set of $\sum_{k=1}^d c_k$ parameters. The necessary conditions for a minimum of the objective function (11) are given by [29]:

$$p_{l,r} = \frac{\sum_{j=1}^n \sum_{(i_1, i_2, \dots, i_k), i_l=r} u_{(i_1, i_2, \dots, i_k), j}^m \cdot x_l}{\sum_{j=1}^n \sum_{(i_1, i_2, \dots, i_k), i_l=r} u_{(i_1, i_2, \dots, i_k), j}^m}. \quad (12)$$

Since the derivation of the necessary condition arrives at a unique solution, convergence of the iterative alternating optimization scheme is guaranteed [30]. Figure 9.9 shows the resulting cluster centers for a two-dimensional data set: the six parameters (three positions on both axes) define nine prototypes, forming a regular grid that approximates the regions of high data density appropriately.

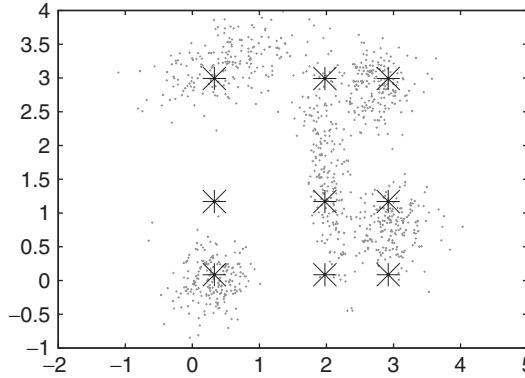


Figure 9.9 Clustering with nine prototypes that share their coordinates (three x values and three y values)

9.6 Task-Driven Partitions

If the purpose of granularization is not only descriptive, the incorporation of application-specific information, i.e., the tailoring of the granules toward the application at hand, always leads to better results. Additional information might be a class label in the case of a classification tasks or an output value or error for regression tasks.

There are two ways to exploit such additional information in the granularization. One way is to use this information to minimize an error like the misclassification rate or the mean squared error directly by choosing or modifying the granules. Another possibility is to use other measures that are related to the actual goals, but do not guarantee an optimal choice with respect to the goal.

9.6.1 One-Dimensional Granules

There is a large variety of techniques to find granules for supervised learning from which we can only present a selection.

A typical example for a granularization based on an indirect strategy is the way in which decision trees are usually constructed. In order to build a classifier, decision trees partition the domains of the single attributes in order to construct the classification rules. We briefly recall a method to partition the domain of a numerical attribute proposed by Elomaa and Rousu [31], an extension of the algorithm described by Fayyad and Irani [32] for finding a partition splitting the domain into two sets only. The partition of the domain of the numerical attribute will be based on another categorical attribute that we intend to predict later on.

We consider a single numerical attribute j whose domain should be partitioned into a predefined number t of intervals. Therefore, $t - 1$ cut points T_1, \dots, T_{t-1} have to be specified. These cut points should be chosen in such a way that the entropy of the partition is minimized. Assume that T_0 and T_t are the left and right boundary of the domain, respectively.

Assume, that we have n data objects and n_i ($i = 1, \dots, t$) of these fall into the interval $[T_{i-1}, T_i]$. Let k_ℓ denote the number of the n_i data that belong to class ℓ . The entropy in this interval is given by

$$E_i = - \sum_{\ell=1}^c \frac{k_\ell}{n_i} \log \left(\frac{k_\ell}{n_i} \right). \quad (13)$$

The entropy of the partition induced by these cut points is simply the weighted sum of the single entropies

$$E = \sum_{i=1}^t \frac{n_i}{n} E_i. \quad (14)$$

The weights are the probability or fractions for the corresponding intervals. The aim is to choose the cut points in such a way that (14) is minimized.

Sorting the data with respect to the values of the j th attribute, it was proved in [31] that it is sufficient to consider boundary points only for finding an optimal partition. If the class label changes directly before or after a value, this value becomes a boundary point. The following example illustrates the concept of boundary points that are marked by lines.

Value:	0	1	2	2	3	4	4	5	5	6	7	7	8	9	10	10	11
Class:	<i>c</i>	<i>c</i>	<i>a</i>	<i>a</i>	<i>a</i>	<i>b</i>	<i>b</i>	<i>a</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>b</i>	<i>a</i>	<i>a</i>	<i>a</i>	

Note that a boundary point occurs after 4 and also before 6 in the attribute j . Since at least one object with value 5 in attribute j belongs to class *a* and at least one object with value 6 in attribute j belongs to another class, namely, class *c*, it is necessary to introduce a boundary point after the second value 5 printed in boldface style.

Once the boundary points are computed, the optimal partition minimizing (14) for a fixed number t of intervals can be constructed. A recursive search can be carried out to find the best partition among the $\binom{b}{t-1}$ possible partitions. Of course, a complete search can be carried out only when $\binom{b}{t-1}$ is reasonably small. Otherwise, heuristic strategies as proposed in [33] might be used to find a suboptimal solution.

Similar ideas can be applied to regression problems. Instead of entropy, the variance in the numerical attribute to be predicted can be taken into account.

Instead of using entropy other measures are also possible. The minimum description length (MDL) principle [34, 35] is proposed in [36] to find good partitions. MDL is a technique for choosing the proper complexity of a model for a given data set. The underlying idea is as follows. The aim is to encode the given data set in such a way that a minimum amount of bits is required to transfer the data over a channel. When a model to represent or approximate the data has been selected, this model is used to compress the data. The total amount of bits to be transferred is the compressed data as well as the description of the model. A very complex model might lead to a very compact representation of the data, resulting in a very effective compression. However, the amount of bits needed to transfer the complex model as well might result in a larger total number of bits to be transferred. On the other hand, a very simple model will need only a few bits to be transferred, but it will usually not provide a good compression for the data.

So far, the granularization or partition was considered to be crisp. But the same strategies can be applied to obtain fuzzy partitions [33]. Once the cut points for the crisp partition are determined, a suitable fuzzy partition can be defined, as shown in Figure 9.10. In [37] an MDL-based approach to find good fuzzy partitions is introduced.

MDL- and entropy-driven techniques for granularization are indirect approaches in the sense that they do not explicitly aim to optimize the actual objective function like the misclassification rate or the mean squared error. There are numerous specific techniques to optimize partitions in order to minimize an objective function like the classification or the mean squared error directly. These techniques include local linear regression models, gradient descent methods, neural networks, or evolutionary algorithms, especially in combination with fuzzy systems [38, 39]. However, to discuss the granularization for such models in detail cannot be within the scope of this contribution, since the methods strongly depend on the underlying model.

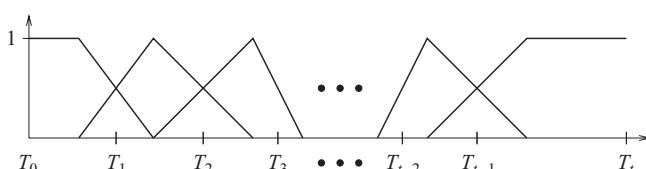


Figure 9.10 A fuzzy partition induced by cut points

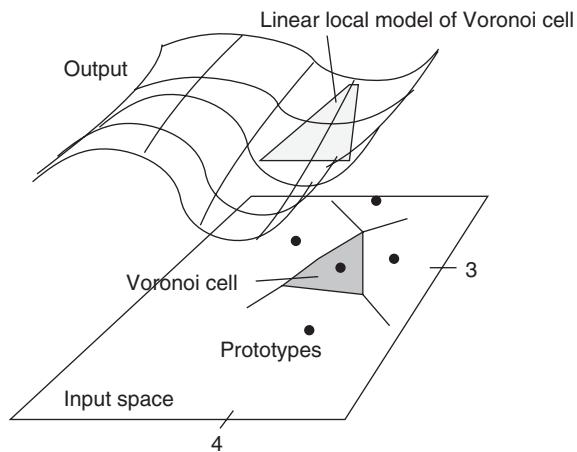


Figure 9.11 Interaction between partitioning task (Voronoi cell) and regression task (local model)

9.6.2 Multidimensional Granules

The techniques mentioned in the previous section focus on granularizations of single attributes. The indirect methods consider the single attributes in an isolated manner. The methods that try to minimize an error measure for the prediction directly take the interaction of different attributes into account, although they still focus on partitions for the single attributes. Although multidimensional granules are more flexible, they are usually difficult to interpret. Therefore, most of the techniques focusing on multidimensional dependencies still try to maintain granules for single attributes, but try to construct them on the basis of their interaction with respect to the variable to be predicted. In [33] techniques are proposed that construct partitions on single variables based on entropy minimization where the interaction of the attributes is taken into account in order to find the best partitions or to simplify partitions. Other methods try to find a multidimensional grid [40] in order to find simple granules.

We have seen in Section 9.5 that clustering-based methods are well suited to find systems of granules. Such unsupervised techniques can also be altered to reflect additional information. This is, e.g., helpful in case of partially labeled data, e.g., when class label information is available only for some data objects (see, e.g., [41]). The clustering techniques can be modified to additionally reflect either class label information or the approximation error in a regression task. In both cases, the objective function of the FCM algorithm is supplemented with an additional penalty term that promotes *pure* granules [42] (classification problem) or granules with a *good fit* [29] (regression problem). After that, we no longer have a clustering algorithm. Since both, partitioning and classification/regression error, are considered in the *same* objective function, an interdependency between the granules and the models is established. A poor classification or fit (cf. Figure 9.11) can be resolved in the next iteration step by a better model *and/or* by modified granules. Therefore, the partitioning of the input space indirectly influences the classification or regression error and vice versa. To ease the interpretation of the granules, the prototypes may additionally be organized on a regular grid, as discussed in Section 9.5.3, leading to Voronoi cells in the shape of hyperboxes.

9.7 Conclusion

Finding useful information granules is closely related to finding good partitions, where a partition should not be understood in a strict mathematical sense. The methods for defining and finding the partitions and the kind of partitions strongly depend on the context in which the granules will be used. This chapter has provided an overview of different purposes for the use as well as various techniques for the construction of granules. The choice of the granularization depends highly on the specific application.

Nowadays, data are usually collected and stored in the finest granularity available to prevent loss of any information. This is, however, completely different from the way humans perceive and understand the world. Depending on the level of abstraction, humans focus on the data at different resolutions or granularities. It is very often the case that the structure inherent in the data becomes evident only at a specific level of abstraction. Exploiting *the right* granularity for an application at hand is therefore essential in human problem solving and consequently in the design and implementation of intelligent systems. For the case of data mining it is quite often not the choice of the learning algorithm but the choice of (knowledge-based or data-driven) granularization that decides about success or failure.

References

- [1] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer, Dordrecht, 2002.
- [2] L. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.
- [3] J. Mill and A. Inoue. Granularization of machine learning. In: *Proceedings of the Eleventh International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Paris, France, July 2–7, 2006, pp. 1907–1912.
- [4] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer, Dordrecht, 1991.
- [5] E. Thomsen. *OLAP Solutions: Building Multidimensional Information Systems*. Wiley, Chichester, England, 2002.
- [6] S. Wrobel. An algorithm for multi-relational discovery of subgroups. In: *Proceedings of the 1st European Conference on Principles of Data Mining and Knowledge Discovery*, Trondheim, Norway, June 24–27, 1997, pp. 78–87.
- [7] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, Chichester, England, 2000.
- [8] D. Dubois and H. Prade. Possibility theory, probability theory and multiple-valued logics: a clarification. *Ann. Math. Artif. Intell.* 32 (2001) 35–66.
- [9] L. Kaufman and P. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, Chichester, England, 2005.
- [10] A. Skowron, J. Stepaniuk, J. Peters, and R. Swiniarski. Calculi of approximation spaces. *Fundam. Inf.* 72 (2006) 363–378.
- [11] P. Sneath. The application of computers to taxonomy. *J. Gen. Microbiol.* 17 (1957) 201–206.
- [12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xiaowei. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the 2nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, United States, August 2–4, 1996, pp. 226–331.
- [13] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high-dimensional data for data mining applications. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Seattle, Washington, United States, June 1–4, 1998, pp. 94–105.
- [14] J.B. McQueen. Some methods of classification and analysis of multivariate observations. In: *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, California, United States, June 21–July 18, 1965, and December 27, 1965 – January 7, 1966, pp. 281–297.
- [15] J. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact, well-separated clusters. *J. Cybern.* 3 (1973) 32–57.
- [16] J.C. Bezdek. A convergence theorem for the fuzzy ISODATA clustering algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* 2 (1980) 1–8.
- [17] A.K. Jain, N.M. Murty, and P.J. Flynn. Data clustering: a review. *ACM Comput. Surv.* 31 (1999) 264–323.
- [18] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ, 1988.
- [19] F. Höppner, F. Klawonn, R. Kruse, and T.A. Runkler. *Fuzzy Cluster Analysis*. Wiley, Chichester, England, 1999.
- [20] F. Klawonn and F. Höppner. What is fuzzy about fuzzy clustering? – understanding and improving the concept of the fuzzifier. In: *Proceedings of the International Conference on Intelligent Data Analysis*, Berlin, Germany, August 28–30, 2003, pp. 254–264.
- [21] A.B. Geva. Feature extraction and state identification in biomedical signals using hierarchical fuzzy clustering. *Med. Biol. Eng. Comput.* 36 (1998) 608–614.
- [22] W. Pedrycz and A. Bargiela. Granular clustering: a granular signature of data. *IEEE Trans. Syst. Man Cybern. Part B* 32 (2002) 212–224.
- [23] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *J. Intell. Inf. Syst.* 17 (2001) 107–145.

- [24] F. Höppner and F. Klawonn. Improved fuzzy partitions for fuzzy regression models. *Int. J. Approx. Reason.* 32 (2003) 85–102.
- [25] F. Klawonn and F. Höppner. Equi-sized, homogeneous partitioning. In: *Proceedings of the International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, Bournemouth, United Kingdom, October 9–11, 2006, pp. 70–77.
- [26] S.D. Bay. Multivariate discretization for set mining. *Knowl. Inf. Syst. J.* 3 (2001) 491–512.
- [27] M. Setnes, R. Babuska, U. Kaymak, and H.R. van Nauta Lemke. Similarity measures in fuzzy rule base simplification. *IEEE Trans. Syst. Man Cybern. Part B* 28 (1998) 376–386.
- [28] F. Klawonn and R. Kruse. Constructing a fuzzy controller from data. *Fuzzy Sets Syst.* 85 (1997) 177–193.
- [29] F. Höppner and F. Klawonn. Learning fuzzy systems – an objective-function approach. *Mathware Soft Comput.* 11 (2004) 143–162.
- [30] F. Höppner and F. Klawonn. A contribution to convergence theory of fuzzy c-means and derivatives. *IEEE Trans. Fuzzy Syst.* 11 (2003) 682–694.
- [31] T. Elomaa and J. Rousu. *Finding Optimal Multi-Splits for Numerical Attributes in Decision Tree Learning*. Technical Report NC-TR-96-041. Department of Computer Science, University of Helsinki, Finland, 1996.
- [32] U. Fayyad and K. Irani. On the handling of continues-valued attributes in decision tree generation. *Mach. Learn.* 8 (2001) 87–102.
- [33] F. Klawonn and D. Nauck. Automatically determine initial fuzzy partitions for neuro-fuzzy classifiers. In: *Proceedings of the IEEE International Conference on Fuzzy Systems*, Vancouver, Canada, July 16–21, 2006, pp. 7894–7900.
- [34] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Ann. Stati.*, 11 (1983) 416–431.
- [35] J. Rissanen. *Stochastic Complexity and Statistical Inquiry*. World Scientific, Singapore, 1989.
- [36] U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In: *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Chambery, France, August 28–September 3, 1993, pp. 1022–1027.
- [37] A. Klose. *Partially Supervised Learning of Fuzzy Classification Rules*. Ph.D. Thesis. Otto von Guericke University Magdeburg, Germany, 2004.
- [38] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena. Ten years of genetic fuzzy systems: current framework and new trends. *Fuzzy Sets Syst.* 141 (2004) 5–31.
- [39] D. Nauck, F. Klawonn, and R. Kruse. *Neuro-Fuzzy Systems*. Wiley, Chichester, England, 1997.
- [40] F. Klawonn and A. Keller. Fuzzy clustering with evolutionary algorithms. *Int. J. Intell. Syst.* 13 (1998) 975–991.
- [41] W. Pedrycz and J. Waletzky. Fuzzy clustering with partial supervision. *IEEE Trans. Syst. Man Cybern. Part B* 27 (1997) 787–795.
- [42] F. Höppner. Objective function-based discretization. In: *Proceedings of the 29th Annual Conference of the Gesellschaft für Klassifikation*, Magdeburg, Germany, March 9–11, 2006, pp. 438–445.

10

Logical Connectives for Granular Computing

Erich Peter Klement, Radko Mesiar, Andrea Mesiarová-Zemánková
and Susanne Saminger-Platz

10.1 Introduction

The concept of granularity had been introduced by Zadeh [1–5] and further developed by many authors, e.g., [6–8]. Recall that, following Zadeh [5], a ‘*granule is a clump of elements drawn together by indistinguishability, similarity, proximity or functionality.*’

Not going into details (these are discussed deeply in other chapters of this volume), recall that the concept of granularity is closely related to the concept of *properties*, on the basis of which we may obtain a clump of objects forming a granule in some possible world (i.e., in some model). A typical example is image processing [9], where we do not focus our attention on individual pixels and process them as such but we group them together into semantically meaningful constructed familiar objects (features) we deal with in everyday life. Such features involve regions that consist of pixels or categories of pixels drawn together because of their proximity in the image, similar texture, color, etc.

An essential characteristic of most granules is their unsharpness and, thus, granular computing is heavily related to existing formalisms of set theory (using, e.g., intervals, random sets, fuzzy sets, or rough sets), exhibiting some fundamental commonalities of these theories. Quite often the granules are expressed linguistically, and their processing is then referred to as *computing with words* [2, 4, 10, 11]. Typically, such linguistically described quantities possess at least an ordinal structure, so they can be represented by some discrete (finite) ordinal scale. These situations will be discussed in Sections 10.3 and 10.5.

Evidently, classical predicate logic, though very powerful in many situations, has no means to model the vagueness phenomenon linked to the unsharpness. An elegant tool to fulfil this task is the fuzzy logic in its narrow sense as discussed, e.g., in [7]. For more details about fuzzy logic we refer to [12, 13].

The aim of this chapter is to discuss deeply the logical connectives depending on the bounded lattice L of truth values modeling the possible unsharpness (only the extremal elements $\mathbf{0}$ and $\mathbf{1}$ correspond to full sharpness.) Note that we will deal with a sharp concept of truth only. Although there are already several models generalizing this approach, e.g., taking special fuzzy sets as possible truth values [14, 15], the processing of logical connectives in such cases can be derived from the models with crisp values. Main attention will be paid to conjunction and negation operators. Operators for other logical connectives like disjunction and implication can mostly be derived from conjunction and negation, and thus we will mention them only briefly.

The chapter is organized as follows. In the next section we recall necessary preliminaries introducing logical operators on a general bounded lattice L . The following sections will be dedicated to logical operations (negation, conjunction, disjunction, and implication) on different types of bounded lattices. In Section 10.3 we discuss the case when L is a finite chain, i.e., we have a finite linearly ordered set of truth values. Section 10.4 deals with the most common carrier of truth values in fuzzy logics, the real unit interval $L = [0, 1]$. After a discussion of logical connectives on infinite discrete chains in Section 10.5, we turn to interval-valued truth values in Section 10.6 and to some other ranges of truth values in Section 10.7.

10.2 Conjunction Operators on Bounded Lattices and Related Logical Operators

Throughout this chapter, L will be a bounded lattice of truth values with top element $\mathbf{1}$ and bottom element $\mathbf{0}$, equipped with a partial order \leq and the operations \wedge and \vee denoting meet and join (or infimum and supremum), respectively. Logical operations on L are extensions of the corresponding Boolean logical operations on $\{\mathbf{0}, \mathbf{1}\}$ characterized by some sets of axioms. Observe that while conjunction and disjunction operators mostly have unique sets of characterizing axioms, there is a variety of approaches to the implication operators. Even negation admits, in general, two approaches.

Negation

Definition 1.

1. A mapping $N: L \rightarrow L$ is called a *negation* if $N(\mathbf{0}) = \mathbf{1}$, $N(\mathbf{1}) = \mathbf{0}$, and N is order reversing, i.e., $N(x) \geq N(y)$ whenever $x \leq y$.
2. A negation $N: L \rightarrow L$ is called a *strong negation* if it is involutive, i.e., $N(N(x)) = x$ for all $x \in L$.

On each truth-value lattice L there is the strongest negation $N^*: L \rightarrow L$ given by

$$N^*(x) = \begin{cases} \mathbf{0} & \text{if } x = \mathbf{1}, \\ \mathbf{1} & \text{otherwise,} \end{cases}$$

and the weakest negation $N_*: L \rightarrow L$ given by

$$N_*(x) = \begin{cases} \mathbf{1} & \text{if } x = \mathbf{0}, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

Evidently, $N^* = N_*$ (and they are strong negations) if and only if $L = \{\mathbf{0}, \mathbf{1}\}$ is the trivial Boolean lattice. Note that there are bounded lattices not admitting strong negations (e.g., $L_{(\infty)} = \{0, 1, 2, \dots, \infty\}$, see Section 10.5).

Conjunction

Conjunction operators on L are defined either directly by means of the axioms (C1)–(C4), see Definition 2, or in the framework of BL-logics introduced by Hájek [12], see Definition 3. For recent advances on this topic we refer to the edited volume [16].

Definition 2. A binary operation $C: L^2 \rightarrow L$ is called a *conjunction operator* on L if it satisfies, for all $x, y, z \in L$, the following four axioms:

- (C1) $C(x, y) = C(y, x)$ (commutativity),
- (C2) $C(x, C(y, z)) = C(C(x, y), z)$ (associativity),
- (C3) $C(x, y) \leq C(x, z)$ whenever $y \leq z$ (monotonicity),
- (C4) $C(x, \mathbf{1}) = x$ (neutral element).

Observe that the only conjunction operator C on the trivial lattice $L = \{\mathbf{0}, \mathbf{1}\}$ is just the Boolean conjunction. In all other cases there are at least two different conjunction operators on L , namely the strongest one $C^*: L^2 \rightarrow L$ given by

$$C^*(x, y) = x \wedge y,$$

and the weakest one $C_*: L^2 \rightarrow L$ given by

$$C_*(x, y) = \begin{cases} x & \text{if } y = \mathbf{1}, \\ y & \text{if } x = \mathbf{1}, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

As mentioned before, conjunctions can also be introduced in the framework of BL-logics and BL-algebras. Their introduction by Hájek [12, definitions 2.2.4 and 2.3.3] marked an important step forward in the framework of fuzzy logics (and many-valued logics, in general).

Definition 3. A logic $(A, \rightarrow, \&, \mathbf{0})$, with A some appropriate set of atomic symbols, which has as axiom system the following schemata

- (BL1) $(\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi)),$
- (BL2) $\varphi \& \psi \rightarrow \varphi,$
- (BL3) $\varphi \& \psi \rightarrow \psi \& \varphi,$
- (BL4) $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow (\varphi \& \psi \rightarrow \chi)$
- (BL5) $(\varphi \& \psi \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \rightarrow \chi))$
- (BL6) $\varphi \& (\varphi \rightarrow \psi) \rightarrow \psi \& (\psi \rightarrow \varphi),$
- (BL7) $((\varphi \rightarrow \psi) \rightarrow \chi) \rightarrow (((\psi \rightarrow \varphi) \rightarrow \chi) \rightarrow \chi),$
- (BL8) $\mathbf{0} \rightarrow \varphi$

and has as its (only) inference rule the rule of detachment (or modus ponens – with respect to the implication connective \rightarrow) is called a BL-logic.

Observe that the axioms (BL1)–(BL8) also have influence on the choice of the truth-value lattice L and that each conjunction $\&$ in BL-logic corresponds to a conjunction operator in the sense of Definition 2. We will discuss these conjunction operators on special domains $L = \{0, \dots, n\}$ and $L = [0, 1]$ in the next sections. In BL-logics, the operator \rightarrow adjoint to the conjunction $\&$ corresponds to a special implication operator on L (see also Definition 6). This operator is also called a residual implication (to a given conjunction operator C), or Φ -operator [17], and we will turn back to these operations again after a brief discussion of disjunction operators.

Disjunction

Disjunction operators on L can be introduced axiomatically as commutative, associative, monotone binary operators on L with neutral element $\mathbf{0}$ (in analogy to Definition 2), or by duality from conjunction operators.

Proposition 4. Let $C: L^2 \rightarrow L$ be a conjunction operator on L and let $N: L \rightarrow L$ be a strong negation on L . Then the mapping $D: L^2 \rightarrow L$ given by

$$D(x, y) = N(C(N(x), N(y))) \tag{1}$$

is a disjunction operator on L .

However, note that the axiomatic approach to disjunction operators is more general, since not each truth-value lattice L admits a strong negation N (see Section 10.5).

Implication

Implication operators on L are hybrid monotone extensions of the Boolean implication on $\{\mathbf{0}, \mathbf{1}\}$ (i.e., they are non-increasing in the first coordinate and non-decreasing in the second coordinate). As such there exists a variety of different concepts. We will restrict here to the most important and common concepts – residual implications as already briefly mentioned before in the context of BL-logics (see Definition 3) and D -implications resp. S -implications referring to a negation and a disjunction operator. For the interested reader we mention that an exhaustive overview of fuzzy implications (i.e., implication operators on $L = [0, 1]$) will appear in [18].

Definition 5. Let $C: L^2 \rightarrow L$ be a conjunction operator. A mapping $R_C: L^2 \rightarrow L$ given by

$$R_C(x, y) = \bigvee \{z \in L \mid C(x, z) \leq y\}$$

for all $(x, y) \in L^2$ is called a residual implication related to C .

Observe that a conjunction operator $C: L^2 \rightarrow L$ and the corresponding residual implication $R_C: L^2 \rightarrow L$ form an adjoint pair whenever

$$C(x, y) \leq z \text{ if and only if } x \leq R_C(y, z).$$

Moreover, (C, R_C) is an adjoint pair if and only if C is \bigvee -preserving (compare with the concept of Galois connection). For a deeper discussion of the adjointness property we refer to [19, 20].

Conjunctions in BL-logics are special kinds of conjunction operators characterized by the *divisibility* property, involving both the conjunction and its corresponding residual implication.

Definition 6. A \bigvee -preserving conjunction operator $C: L^2 \rightarrow L$ is called divisible if, for all $(x, y) \in L^2$, we have $C(x, R_C(x, y)) = x \wedge y$.

Note that an axiomatic approach to residual implications on $L = [0, 1]$ was developed in [21], and it can be generalized in a straightforward way to a general L . Moreover, for a given residual implication R_C , the mapping $N_C: L \rightarrow L$ given by

$$N_C(x) = R_C(x, \mathbf{0})$$

is a negation on L , which satisfies $N_C \circ N_C \circ N_C = N_C$.

In the framework of BL-logics, the involutivity of N_C (i.e., N_C is a strong negation) and the divisibility of C lead to the class of Lukasiewicz's logics [12, 22].

Finally let us turn to a type of implication referring to a negation and a disjunction operator.

Definition 7. Let $D: L^2 \rightarrow L$ be a disjunction operator and $N: L \rightarrow L$ a negation. A mapping $I_{N,D}: L^2 \rightarrow L$ given by

$$I_{N,D}(x, y) = D(N(x), y)$$

for all $(x, y) \in L^2$ is called D -implication.

Observe that when $L = [0, 1]$, disjunction operators are usually denoted by the letter S and then the name S -implication is therefore used instead of D -implication.

Several further logical connectives on L like, e.g., biimplications can be derived by means of the four types of logical operations already introduced and we refer to, e.g., [12, 13, 22] for those connectives. Next, we will focus on particular cases of truth-value lattices and discuss negation, conjunction, disjunction, as well as implication operators on such lattices.

10.3 Logical Connectives on Finite Chains

A finite chain $L_{(n)} = \{0, 1, \dots, n\}$ with $n > 1$ is the most natural extension of the trivial lattice $L_{(1)} = \{0, 1\}$. Indeed, the origin of many-valued logics [23] was the three-valued logic on $L = \{0, \frac{1}{2}, 1\}$, which is isomorphic to $L_{(2)}$. Moreover, this case also covers all situations with a finite number of truth values equipped with a linear order. For example, truth-values can be granular, expressed linguistically as *false*, *more or less true*, *almost true*, and *true*, in which case they can be represented by the chain $L_{(3)}$.

For each $n \in \mathbb{N}$, and therefore for each $L_{(n)}$, there is a unique strong negation $N: L_{(n)} \rightarrow L_{(n)}$ given by $N(x) = n - x$.

Conjunction operators on $L_{(n)}$ are often called *discrete t-norms* [24] and are defined axiomatically in accordance with Definition 2. The number of conjunction operators on $L_{(n)}$ is growing extremely fast with n [25] (see Table 10.1). Divisible conjunction operators on $L_{(n)}$ were introduced in [26] (called there *smooth discrete t-norms*) and characterized in [27].

Theorem 8. A mapping $C: L_{(n)}^2 \rightarrow L_{(n)}$ is a divisible conjunction operator on $L_{(n)}$ if and only if there is a subset $K \subset L_{(n)}$, containing 0 and n , i.e., $K = \{i_1, \dots, i_m\}$ with $0 = i_1 < \dots < i_m = n$, such that

$$C(x, y) = \begin{cases} \max(x + y - i_{j+1}, i_j) & \text{if } (x, y) \in \{i_j, \dots, i_{j+1} - 1\}^2, \\ \min(x, y) & \text{otherwise.} \end{cases} \quad (2)$$

Each divisible conjunction operator on $L_{(n)}$ is therefore characterized by a subset $K \cap \{1, \dots, n - 1\}$. Hence there are exactly 2^{n-1} divisible conjunction operators on $L_{(n)}$. Divisible conjunction operators are further characterized by the 1-Lipschitz property [24].

Proposition 9. A conjunction operator $C: L_{(n)}^2 \rightarrow L_{(n)}$ is divisible if and only if it is 1-Lipschitz, i.e., for all $x, y, x', y' \in L_{(n)}$,

$$|C(x, y) - C(x', y')| \leq |x - x'| + |y - y'|. \quad (3)$$

The unique strong negation N on $L_{(n)}$ brings a one-to-one correspondence between disjunction and conjunction operators on $L_{(n)}$. Note that for a given divisible conjunction operator $C: L_{(n)}^2 \rightarrow L_{(n)}$, the corresponding disjunction operator $D: L_{(n)}^2 \rightarrow L_{(n)}$ given by (1) is also characterized by the 1-Lipschitz property (3). Another way of relating 1-Lipschitz discrete conjunction and disjunction operators on $L_{(n)}$ is characterized by the so-called *Frank functional equation*

$$C(x, y) + D(x, y) = x + y \quad (4)$$

for all $(x, y) \in L_{(n)}^2$, where C is an arbitrary divisible conjunction operator on $L_{(n)}$. If this C is given by (2), then $D: L_{(n)}^2 \rightarrow L_{(n)}$ is given by

$$D(x, y) = \begin{cases} \min(x + y - i_j, i_{j+1}) & \text{if } (x, y) \in \{i_j, \dots, i_{j+1} - 1\}^2, \\ \max(x, y) & \text{otherwise.} \end{cases} \quad (5)$$

Table 10.1 Number of conjunction operators on $L_{(n)}$

n	Conjunction operators	n	Conjunction operators	n	Conjunction operators
1	1	5	94	9	86.417
2	2	6	451	10	590.489
3	6	7	2.386	11	4,446.029
4	22	8	13.775	12	37,869.449

Note that the structure (2) of divisible conjunction operators on $L_{(n)}$ (in fact, a particular case of an *ordinal sum* of discrete t-norms [28, 29]) has an impact not only on the structure of the corresponding disjunctions, but also on the similar structure of the corresponding residual implications. Indeed, if $C: L_{(n)}^2 \rightarrow L_{(n)}$ is given by (2) then the related residual implication $R_C: L_{(n)}^2 \rightarrow L_{(n)}$ is given by

$$R_C(x, y) = \begin{cases} n & \text{if } x \leq y, \\ i_{j+1} - x + y & \text{if } x > y \text{ and } (x, y) \in \{i_j, \dots, i_{j+1} - 1\}^2, \\ y & \text{otherwise.} \end{cases} \quad (6)$$

The only divisible conjunction operator $C_L: L_{(n)}^2 \rightarrow L_{(n)}$ on $L_{(n)}$ such that the negation N_{C_L} is a strong negation (i.e., it is the only strong negation N on $L_{(n)}$) corresponds to the minimal set $K = \{0, n\}$ (see Theorem 8). It is given by

$$C_L(x, y) = \max(x + y - n, 0),$$

and it is called the *Lukasiewicz conjunction operator* (*Lukasiewicz discrete t-norm*) on $L_{(n)}$.

10.4 Logical Connectives on Closed Intervals

In the last decades, the greatest attention in the framework of many-valued logics was paid to the case when the range of truth values is $L = [0, 1]$, i.e., when L is the real unit interval. This range is also the background of fuzzy sets [30], rough sets [31], etc. Note that each lattice $L = [a, b] \subset [-\infty, \infty]$ is isomorphic to the lattice $[0, 1]$, and thus the logical connectives on $[a, b]$ can be derived in a straightforward way from the corresponding logical connectives on $[0, 1]$.

Negation

Let us first turn to negations which can be defined in accordance with Definition 1, i.e., any non-increasing function $N: [0, 1] \rightarrow [0, 1]$ with $N(0) = 1$ and $N(1) = 0$ constitutes a negation on $[0, 1]$. Strong negations on $[0, 1]$ were characterized in [32] and are related to increasing bijections.

Theorem 10. *A mapping $N: [0, 1] \rightarrow [0, 1]$ is a strong negation on $[0, 1]$ if and only if there is an increasing bijection $f: [0, 1] \rightarrow [0, 1]$, such that*

$$N(x) = f^{-1}(1 - f(x)). \quad (7)$$

Note that there are uncountably many bijections f leading to the same strong negation N . The *standard negation* $N_s: [0, 1] \rightarrow [0, 1]$ is generated by the identity $\text{id}_{[0,1]}$ and it is given by

$$N_s(x) = 1 - x.$$

Conjunction

Conjunction operators on $[0, 1]$ are called *triangular norms*, and they were originally introduced by Schweizer and Sklar [33, 34] in the framework of probabilistic metric spaces, generalizing earlier ideas of Menger [35]. More details about t-norms can be found in the recent monographs [29, 36].

Triangular norms can be rather peculiar, in general. For example, there are Borel non-measurable t-norms [37], or t-norms which are non-continuous in a single point [29, 36]. An example of the latter type is the t-norm $T: [0, 1]^2 \rightarrow [0, 1]$ given by

$$T(x, y) = \begin{cases} a & \text{if } x = y = \frac{1}{2}, \\ \frac{\max(x+y-1, 0)}{1-4(1-x)(1-y)} & \text{otherwise,} \end{cases} \quad (8)$$

where either $a = 0$ (in which case T is \vee -preserving) or $a = \frac{1}{2}$ (then T is \wedge -preserving).

The strongest t-norm $T^*: [0, 1]^2 \rightarrow [0, 1]$ is given simply by $T^*(x, y) = \min(x, y)$, and it is usually denoted as T_M . Similarly, the weakest t-norm $T_*: [0, 1]^2 \rightarrow [0, 1]$ is usually denoted as T_D (and it is called the *drastic product*). The t-norm T_M is an example of a continuous t-norm, while T_D is a non-continuous (but right-continuous) t-norm. Other typical examples of continuous t-norms are the product t-norm T_P and the Lukasiewicz t-norm T_L given by $T_P(x, y) = xy$ and $T_L(x, y) = \max(x + y - 1, 0)$, respectively.

Divisible triangular norms are just the continuous t-norms and their representation is given in [38]. Observe that the following representation theorem can also be derived from results of Mostert and Shields [39] in the context of I -semigroups and that some preliminary results can also be found in [40, 41].

Theorem 11. *A function $T: [0, 1]^2 \rightarrow [0, 1]$ is a continuous t-norm if and only if T is an ordinal sum of continuous Archimedean t-norms.*

Note that a t-norm $T: [0, 1]^2 \rightarrow [0, 1]$ is *Archimedean* if for each $x, y \in]0, 1[$, there is an $n \in \mathbb{N}$ such that $x_T^{(n)} < y$, where $x_T^{(1)} = x$ and for $n > 1$, $x_T^{(n)} = T(x_T^{(n-1)}, x)$. For example, the drastic product T_D is Archimedean. If T is a continuous t-norm, then its Archimedeanity is equivalent to the diagonal inequality $T(x, x) < x$ for all $x \in]0, 1[$. Observe that both T_P and T_L are continuous Archimedean t-norms.

Further, the *ordinal sum of t-norms* is a construction method coming from semigroup theory (introduced by Clifford [28] for abstract semigroups). For ordinal-sum-like conjunction operators on other truth-value lattices compare also equations (2), (11), and (13).

Definition 12. *Let $(T_\alpha)_{\alpha \in A}$ be a family of t-norms and $(]a_\alpha, e_\alpha[)_{\alpha \in A}$ be a family of non-empty, pairwise disjoint open subintervals of $[0, 1]$. The t-norm T defined by*

$$T(x, y) = \begin{cases} a_\alpha + (e_\alpha - a_\alpha) \cdot T_\alpha(\frac{x-a_\alpha}{e_\alpha-a_\alpha}, \frac{y-a_\alpha}{e_\alpha-a_\alpha}) & \text{if } (x, y) \in [a_\alpha, e_\alpha]^2, \\ \min(x, y) & \text{otherwise,} \end{cases}$$

is called the *ordinal sum of the summands $\langle a_\alpha, e_\alpha, T_\alpha \rangle$, $\alpha \in A$* , and we shall write

$$T = (\langle a_\alpha, e_\alpha, T_\alpha \rangle)_{\alpha \in A}.$$

Observe that the index set A is necessarily finite or countably infinite. It may also be empty, in which case the ordinal sum equals the strongest t-norm T_M .

Continuous Archimedean t-norms are strongly related to the standard addition on $[0, \infty]$.

Theorem 13. *For a function $T: [0, 1]^2 \rightarrow [0, 1]$, the following are equivalent:*

1. *T is a continuous Archimedean t-norm.*
2. *T has a continuous additive generator, i.e., there exists a continuous, strictly decreasing function $t: [0, 1] \rightarrow [0, \infty]$ with $t(1) = 0$, which is uniquely determined up to a positive multiplicative constant, such that for all $(x, y) \in [0, 1]^2$,*

$$T(x, y) = t^{-1}(\min(t(0), t(x) + t(y))). \quad (9)$$

Note that a continuous Archimedean t-norm $T: [0, 1]^2 \rightarrow [0, 1]$ is called *nilpotent* whenever there is an $x \in]0, 1[$ and $n \in \mathbb{N}$ such that $x_T^{(n)} = 0$, and it is called *strict* otherwise. Strict t-norms are also characterized by the *cancellation property*, i.e., $T(x, y) = T(x, z)$ only if either $x = 0$ or $y = z$.

Each strict t-norm T has an unbounded additive generator t , i.e., $t(0) = \infty$. Vice versa, each additive generator t of a nilpotent t-norm T is bounded, i.e., $t(0) < \infty$. Moreover, each strict t-norm T is isomorphic to T_P (i.e., there is an increasing bijection $\varphi: [0, 1] \rightarrow [0, 1]$ such that $T(x, y) = \varphi^{-1}(\varphi(x) \cdot \varphi(y))$) and each nilpotent t-norm T is isomorphic to T_L .

The combination of these results yields the following representation of continuous (i.e., divisible) t-norms:

Corollary 14. *For a function $T: [0, 1]^2 \rightarrow [0, 1]$, the following are equivalent:*

1. *T is a continuous t-norm.*
2. *T is isomorphic to an ordinal sum whose summands contain only the t-norms T_P and T_L .*
3. *There is a family $([a_\alpha, e_\alpha])_{\alpha \in A}$ of non-empty, pairwise disjoint open subintervals of $[0, 1]$ and a family $h_\alpha: [a_\alpha, e_\alpha] \rightarrow [0, \infty]$ of continuous, strictly decreasing functions with $h_\alpha(e_\alpha) = 0$ for each $\alpha \in A$ such that for all $(x, y) \in [0, 1]^2$,*

$$T(x, y) = \begin{cases} h_\alpha^{-1}(\min(h_\alpha(a_\alpha), h_\alpha(x) + h_\alpha(y))) & \text{if } (x, y) \in [a_\alpha, e_\alpha]^2, \\ \min(x, y) & \text{otherwise.} \end{cases} \quad (10)$$

Several examples of parameterized families of (continuous Archimedean) t-norms can be found in [29, 36]. We recall here only three such families.

Example 15.

1. *The family $(T_\lambda^{\text{SS}})_{\lambda \in [-\infty, \infty]}$ of Schweizer–Sklar t-norms is given by*

$$T_\lambda^{\text{SS}}(x, y) = \begin{cases} T_M(x, y) & \text{if } \lambda = -\infty, \\ T_P(x, y) & \text{if } \lambda = 0, \\ T_D(x, y) & \text{if } \lambda = \infty, \\ (\max((x^\lambda + y^\lambda - 1), 0))^{\frac{1}{\lambda}} & \text{if } \lambda \in]-\infty, 0[\cup]0, \infty[. \end{cases}$$

2. *Additive generators $t_\lambda^{\text{SS}}: [0, 1] \rightarrow [0, \infty]$ of the continuous Archimedean members $(T_\lambda^{\text{SS}})_{\lambda \in [-\infty, \infty]}$ of the family of Schweizer–Sklar t-norms are given by*

$$t_\lambda^{\text{SS}}(x) = \begin{cases} -\ln x & \text{if } \lambda = 0, \\ \frac{1-x^\lambda}{\lambda} & \text{if } \lambda \in]-\infty, 0[\cup]0, \infty[. \end{cases}$$

This family of t-norms is remarkable in the sense that it contains all four basic t-norms.

The investigations of the associativity of duals of copulas in the framework of distribution functions led to the following problem: characterize all continuous (or, equivalently, non-decreasing) associative functions $F: [0, 1]^2 \rightarrow [0, 1]$ which satisfy for each $x \in [0, 1]$, the boundary conditions $F(0, x) = F(x, 0) = 0$ and $F(x, 1) = F(1, x) = x$, such that the function $G: [0, 1]^2 \rightarrow [0, 1]$ given by

$$G(x, y) = x + y - F(x, y)$$

is also associative. In [42] it was shown that then F has to be an ordinal sum of members of the following family of t-norms.

Example 16.

1. *The family $(T_\lambda^F)_{\lambda \in [0, \infty]}$ of Frank t-norms (which were called fundamental t-norms in [43]) is given by*

$$T_\lambda^F(x, y) = \begin{cases} T_M(x, y) & \text{if } \lambda = 0, \\ T_P(x, y) & \text{if } \lambda = 1, \\ T_L(x, y) & \text{if } \lambda = \infty, \\ \log_\lambda(1 + \frac{(\lambda^x - 1)(\lambda^y - 1)}{\lambda - 1}) & \text{otherwise.} \end{cases}$$

2. Additive generators $t_\lambda^F: [0, 1] \rightarrow [0, \infty]$ of the continuous Archimedean members $(T_\lambda^F)_{\lambda \in]0, \infty]}$ of the family of Frank t-norms are given by

$$t_\lambda^F(x) = \begin{cases} -\ln x & \text{if } \lambda = 1, \\ 1 - x & \text{if } \lambda = \infty, \\ \ln(\frac{\lambda-1}{\lambda-1}) & \text{if } \lambda \in]0, 1[\cup]1, \infty[. \end{cases}$$

Another family used for modeling the intersection of fuzzy sets is the following family of t-norms (which was first introduced in [44] for the special case $\lambda \geq 1$ only). The idea was to use the parameter λ as a reciprocal measure for the strength of the logical AND. In this context, $\lambda = 1$ expresses the most demanding (i.e., smallest) AND, and $\lambda = \infty$ the least demanding (i.e., largest) AND.

Example 17.

1. The family $(T_\lambda^Y)_{\lambda \in [0, \infty]}$ of Yager t-norms is given by

$$T_\lambda^Y(x, y) = \begin{cases} T_D(x, y) & \text{if } \lambda = 0, \\ T_M(x, y) & \text{if } \lambda = \infty, \\ \max(1 - ((1-x)^\lambda + (1-y)^\lambda)^{\frac{1}{\lambda}}, 0) & \text{otherwise.} \end{cases}$$

2. Additive generators $t_\lambda^Y: [0, 1] \rightarrow [0, \infty]$ of the nilpotent members $(T_\lambda^Y)_{\lambda \in]0, \infty]}$ of the family of Yager t-norms are given by

$$t_\lambda^Y(x) = (1-x)^\lambda.$$

Another interesting class of t-norms are internal triangular norms, i.e., t-norms $T: [0, 1]^2 \rightarrow [0, 1]$ such that $T(x, y) \in \{0, x, y\}$ for all $(x, y) \in [0, 1]^2$ (see also [29]).

Theorem 18. A function $T: [0, 1]^2 \rightarrow [0, 1]$ is an internal t-norm if and only if there is a subset $A \subset]0, 1[^2$, such that $(x, y) \in A$ implies $(y, x) \in A$ (symmetry of A) and $(u, v) \in A$ for all $u \in]0, x]$, $v \in]0, y]$ (root property of A), and

$$T(x, y) = \begin{cases} 0 & \text{if } (x, y) \in A, \\ \min(x, y) & \text{otherwise.} \end{cases}$$

Note that T_M and T_D are internal t-norms (related to $A = \emptyset$ and $A =]0, 1[^2$, respectively). An important example of a \vee -preserving internal t-norm is the *nilpotent minimum* $T^{nM}: [0, 1]^2 \rightarrow [0, 1]$ given by

$$T^{nM}(x, y) = \begin{cases} 0 & \text{if } x + y \leq 1, \\ \min(x, y) & \text{otherwise.} \end{cases}$$

On the basis of these results, let us now turn to disjunction and implication operators on $[0, 1]$.

Disjunction

Disjunction operators on $[0, 1]$ are called *triangular conorms*, and they are usually denoted by letter S . All results concerning triangular conorms can be derived from the corresponding results for triangular norms by means of the duality. For a given t-norm $T: [0, 1]^2 \rightarrow [0, 1]$, its dual t-conorm $S: [0, 1]^2 \rightarrow [0, 1]$ is given by

$$S(x, y) = 1 - T(1 - x, 1 - y),$$

i.e., the standard negation N_s connects T and its dual S .

The four basic t-conorms (dual to the basic t-norms) are S_M , S_D , S_P , and S_L given by

$$S_M(x, y) = \max(x, y),$$

$$S_D(x, y) = \begin{cases} x & \text{if } y = 0, \\ y & \text{if } x = 0, \\ 1 & \text{otherwise,} \end{cases}$$

$$S_P(x, y) = 1 - (1 - x)(1 - y) = x + y - xy,$$

$$S_L(x, y) = \min(x + y, 1).$$

We only briefly note that in ordinal sums of t-conorms the operator max plays the same role as the operator min in the case of ordinal sums of t-norms. Concerning an additive generator $s: [0, 1] \rightarrow [0, \infty]$ of a continuous Archimedean t-conorm $S: [0, 1]^2 \rightarrow [0, 1]$, s is continuous, strictly increasing and $s(0) = 0$, and $S(x, y) = s^{-1}(\min(s(1), s(x) + s(y)))$. If $t: [0, 1] \rightarrow [0, \infty]$ is an additive generator of the corresponding dual t-norm T , then $s = t \circ N_s$.

Implication

Turning our attention to the implication operators on $[0, 1]$, observe that the residual implications forming an adjoint pair (T, R_T) are related to \vee -preserving t-norms (recall that, in the lattice $[0, 1]$, the fact that T is \vee -preserving is equivalent to the left continuity of T as a function from $[0, 1]^2$ to $[0, 1]$, so both notations are used in the literature). A deep survey on \vee -preserving t-norms is due to Jenei [45].

In the case of BL-logics, residual implications are related to divisible (i.e., continuous) t-norms. Observe that $R_T: [0, 1]^2 \rightarrow [0, 1]$ is continuous if and only if T is a nilpotent t-norm. Similarly, $N_T: [0, 1] \rightarrow [0, 1]$, $N_T(x) = R_T(x, 0)$, is a strong negation if and only if T is a nilpotent t-norm.

Note, however, that there are non-continuous \vee -preserving t-norms T such that N_T is a strong negation. As an example recall the nilpotent minimum T^{nM} , in which case $N_{T^{nM}} = N_s$. Similarly, $N_T = N_s$ for the t-norm T given in (8) for $a = 0$. Also another property of nilpotent t-norms is remarkable: for each nilpotent t-norm T , its adjoint residual implication R_T coincides with the S -implication $I_{N_T, S}$, where $S: [0, 1]^2 \rightarrow [0, 1]$ is a t-conorm N_T -dual to T , $S(x, y) = N_T(T(N_T(x), N_T(y)))$.

We recall the three basic residual implications:

$$R_{T_M}(x, y) = \begin{cases} 1 & \text{if } x \leq y, \\ y & \text{otherwise,} \end{cases} \quad (\text{G\"odel implication})$$

$$R_{T_P}(x, y) = \begin{cases} 1 & \text{if } x \leq y, \\ \frac{y}{x} & \text{otherwise,} \end{cases} \quad (\text{Goguen implication})$$

$$R_{T_L}(x, y) = \min(1, 1 - x + y). \quad (\text{Lukasiewicz implication})$$

Distinguished examples of S -implications based on the standard negation N_s are as follows: (Note that $I_{N_s, S_L} = R_{T_L}$ is the Lukasiewicz implication.)

$$I_{N_s, S_P}(x, y) = 1 - x + xy, \quad (\text{Reichenbach implication})$$

$$I_{N_s, S_M}(x, y) = \max(1 - x, y). \quad (\text{Kleene--Dienes implication})$$

Formally, all BL-logics based on a strict t-norm T are isomorphic to the *product logic*, i.e., to the BL-logic based on T_P . Adding a new connective to these logics, namely a strong negation N (observe that $N_T = N_{T_P} = N_*$ is the weakest negation for each strict t-norm T), we obtain at least two different types of logics. One of them is based (up to isomorphism) on T_P and N_s , while the another one on the Hamacher product T_H and N_s , where $T_H: [0, 1]^2 \rightarrow [0, 1]$ is defined by

$$T_H(x, y) = \frac{xy}{x + y - xy},$$

using the convention $\frac{0}{0} = 0$. For more details about strict BL-logics with a strong negation we refer to [46].

10.5 Logical Connectives on Infinite Discrete Chains

In Section 10.3, we discussed logical connectives on finite discrete lattices forming a chain $L_{(n)}$ with $n \in \mathbb{N}$, whereas Section 10.4 deals with logical connectives on the unit interval as such on a chain with infinitely many arguments. Some other discrete chain lattices (necessarily infinite) have been discussed in [24] and will be in the focus of this section, namely the truth-value lattices $L_{(\infty)} = \{0, 1, \dots, \infty\}$ resp. $L_{(-\infty)} = \{-\infty, \dots, -1, 0\}$ as well as $L_{(-\infty, \infty)} = \{-\infty, \dots, -1, 0, 1, \dots, \infty\}$.

As already mentioned in Section 10.2, there is no strong negation on the lattice $L_{(\infty)}$ (equipped with the standard order), and thus there is no duality between conjunction operators on $L_{(\infty)}$ and disjunction operators on $L_{(\infty)}$.

Further, there is no divisible Archimedean conjunction operator on $L_{(\infty)}$. (The Archimedean property is defined similarly as on $[0, 1]$, see also p. 211, and on $L_{(\infty)}$ it is equivalent to the non-existence of non-trivial idempotent elements.) The divisibility of conjunction and disjunction operators on $L_{(\infty)}$ is characterized by the 1-Lipschitz property similarly as in the case of finite chains $L_{(n)}$. However, there is a unique divisible Archimedean disjunction operator $D_+: L_{(\infty)}^2 \rightarrow L_{(\infty)}$ given by $D_+(x, y) = x + y$.

The following result from [24] characterizes all divisible conjunction operators on $L_{(\infty)}$.

Theorem 19. A mapping $C: L_{(\infty)}^2 \rightarrow L_{(\infty)}$ is a divisible conjunction operator on $L_{(\infty)}$ if and only if there is a strictly increasing sequence $(n_i)_{i=0}^\infty$ of elements of $L_{(\infty)}$ with $n_0 = 0$ such that

$$C(x, y) = \begin{cases} \max(n_i, x + y - n_{i+1}) & \text{if } (x, y) \in [n_i, n_{i+1}]^2, \\ \min(x, y) & \text{otherwise.} \end{cases} \quad (11)$$

For divisible disjunction operators on $L_{(\infty)}$ we have the following characterization.

Theorem 20. A mapping $D: L_{(\infty)}^2 \rightarrow L_{(\infty)}$ is a divisible disjunction operator on $L_{(\infty)}$ if and only if there is a strictly increasing sequence $(n_i)_{i=0}^\infty$ or $(n_i)_{i=0}^m$ of elements of $L_{(\infty)}$ with $n_0 = 0$, and whenever the sequence is finite then $n_m = \infty$, such that

$$D(x, y) = \begin{cases} \min(n_{i+1}, x + y - n_i) & \text{if } (x, y) \in [n_i, n_{i+1}]^2, \\ \max(x, y) & \text{otherwise.} \end{cases}$$

Observe that divisible disjunction operators on $L_{(\infty)}$ are in a one-to-one correspondence with the subsets on \mathbb{N} (non-trivial idempotent elements of D), while divisible conjunction operators on $L_{(\infty)}$ are related to infinite subsets of \mathbb{N} .

Note that to any divisible disjunction operator $D: L_{(\infty)}^2 \rightarrow L_{(\infty)}$ the mapping $C: L_{(\infty)}^2 \rightarrow L_{(\infty)}$ given by $C(x, y) = x + y - D(x, y)$, using the convention $x + y - D(x, y) = \min(x, y)$ if $\max(x, y) = \infty$, is a conjunction operator on $L_{(\infty)}$. It is divisible if and only if the set of idempotent elements of D is infinite. For example, for the only Archimedean divisible disjunction operator D_+ , the corresponding conjunction operator on $L_{(\infty)}$ is just the weakest one, i.e., C_* (which evidently is not divisible). The above relation means that the Frank functional equation $C(x, y) + D(x, y) = x + y$ on $L_{(\infty)}$ also has non-divisible solutions w.r.t. the conjunction operators C .

Concerning the implication operators on $L_{(\infty)}$, it is remarkable that there is no implication operator which is simultaneously a D -implication and a residual implication operator related to some divisible conjunction operator on $L_{(\infty)}$. This is no more true if we consider non-divisible conjunction operators on $L_{(\infty)}$. We give here some examples:

1. For $n \in \mathbb{N}$, let the negation $N_n: L_{(\infty)} \rightarrow L_{(\infty)}$ be given by

$$N_n(x) = \begin{cases} \infty & \text{if } x = 0, \\ n - x & \text{if } x \in [1, n[, \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$I_{N_n, D_+}(x, y) = \begin{cases} \infty & \text{if } x = 0, \\ n - x + y & \text{if } x \in [1, n[, \\ y & \text{otherwise.} \end{cases}$$

2. For the weakest conjunction operator C_* on $L_{(\infty)}$, we get

$$R_{C_*}(x, y) = \begin{cases} \infty & \text{if } x < \infty, \\ y & \text{if } x = \infty. \end{cases}$$

Observe that $R_{C_*} = I_{N^*, D^*}$, i.e., the residual implication related to the weakest conjunction operator C_* on $L_{(\infty)}$ coincides with the D -implication with respect to the strongest negation $N^*: L_{(\infty)} \rightarrow L_{(\infty)}$ and the strongest disjunction operator D^* on $L_{(\infty)}$.

3. Let $C: L_{(\infty)}^2 \rightarrow L_{(\infty)}$ be a divisible conjunction operator determined by the sequence $(n_i)_{i=0}^\infty$. Then the corresponding residual implication operator $R_C: L_{(\infty)}^2 \rightarrow L_{(\infty)}$ is given by

$$R_C(x, y) = \begin{cases} \infty & \text{if } x \leq y, \\ n_{i+1} - x + y & \text{if } x > y \text{ and } (x, y) \in [n_i, n_{i+1}]^2, \\ y & \text{otherwise.} \end{cases}$$

Logical connectives on the lattice $L_{(-\infty)} = \{-\infty, \dots, -1, 0\}$ can be derived from the logical connectives on $L_{(\infty)}$; only the role of conjunction operators and disjunction operators is reversed. So, e.g., the only divisible Archimedean conjunction operator $C_+: L_{(-\infty)}^2 \rightarrow L_{(-\infty)}$ is given by $C_+(x, y) = x + y$ (and there is no divisible Archimedean disjunction operator on $L_{(-\infty)}$).

Another interesting discrete chain is the lattice

$$L_{(-\infty, \infty)} = \{-\infty, \dots, -1, 0, 1, \dots, \infty\}.$$

Following [24], each strong negation on $L_{(-\infty, \infty)}$ is determined by its value in 0, and thus each strong negation on $L_{(-\infty, \infty)}$ belongs to the family $(N_n)_{n \in \mathbb{Z}}$, where \mathbb{Z} is the set of all integers, and $N_n: L_{(-\infty, \infty)} \rightarrow L_{(-\infty, \infty)}$ is given by $N_n(x) = n - x$.

The existence of strong negations ensures the duality between the classes of conjunction operators and disjunction operators on $L_{(-\infty, \infty)}$. Divisible conjunction operators on $L_{(-\infty, \infty)}$ are characterized by infinite sets of idempotent elements. Similarly as in the case of conjunction operators on $L_{(\infty)}$ (even the same formula can be applied), the only restriction is that there are always infinitely many idempotent elements from the set $\{0, 1, \dots\}$. Take, e.g., set $J = \{-\infty, 0, 1, \dots, \infty\}$. Then the corresponding conjunction operator $C_J: L_{(-\infty, \infty)}^2 \rightarrow L_{(-\infty, \infty)}$ is given by

$$C_J(x, y) = \begin{cases} x + y & \text{if } (x, y) \in \{-\infty, \dots, -1, 0\}^2, \\ \min(x, y) & \text{otherwise.} \end{cases}$$

Taking the strong negation $N_0: L_{(-\infty, \infty)} \rightarrow L_{(-\infty, \infty)}$ given by $N_0(x) = -x$, the dual disjunction operator $D_J: L_{(-\infty, \infty)}^2 \rightarrow L_{(-\infty, \infty)}$ is given by

$$D_J(x, y) = \begin{cases} x + y & \text{if } (x, y) \in \{0, 1, \dots, \infty\}^2, \\ \max(x, y) & \text{otherwise.} \end{cases}$$

Concerning the implication operators on $L_{(-\infty, \infty)}$, we introduce only two examples based on the logical connectives mentioned above. The residual implication $R_{C_J}: L_{(-\infty, \infty)}^2 \rightarrow L_{(-\infty, \infty)}$ is given by

$$R_{C_J}(x, y) = \begin{cases} \infty & \text{if } x \leq y, \\ y - x & \text{if } x > y \text{ and } (x, y) \in \{-\infty, \dots, -1, 0\}^2, \\ y & \text{otherwise.} \end{cases}$$

The D -implication $I_{N_0, D_J}: L_{(-\infty, \infty)}^2 \rightarrow L_{(-\infty, \infty)}$ is given by

$$I_{N_0, D_J}(x, y) = \begin{cases} y - x & \text{if } x \leq 0 \leq y, \\ y & \text{if } 0 < x \text{ and } -x \leq y, \\ -x & \text{otherwise.} \end{cases}$$

10.6 Logical Connectives on Interval-Valued Truth-Value Lattices

A genuine model of uncertainty of truth-values in many-valued logics is formed by intervals of some underlying lattice L . Denote by L' the set of all intervals $[x, y] = \{z \in L \mid x \leq z \leq y\}$ with $x, y \in L$ and $x \leq y$. Evidently, L can be embedded into L' by means of the trivial intervals $[x, x] = \{x\}$, $x \in L$. Moreover, L' is a lattice with bottom element $[\mathbf{0}, \mathbf{0}]$ and top element $[\mathbf{1}, \mathbf{1}]$, and with joint and meet inherited from the original lattice L , i.e.,

$$[x, y] \bigvee [u, v] = [x \vee u, y \vee v] \quad \text{and} \quad [x, y] \bigwedge [u, v] = [x \wedge u, y \wedge v].$$

Note that we cannot repeat the approach from interval arithmetic [47] when looking for the logical connectives on L' . For example, for any non-trivial lattice L , take any element $a \in L \setminus \{\mathbf{0}, \mathbf{1}\}$. For the weakest conjunctive $C_*: L^2 \rightarrow L$, putting $C'_*([a, \mathbf{1}], [a, \mathbf{1}]) = \{z \in L \mid z = C_*(x, y), (x, y) \in [a, \mathbf{1}]^2\}$ we see that the result of this operation is $[a, \mathbf{1}] \cup \{\mathbf{0}\}$, which is an element of L' only if $[a, \mathbf{1}] \cup \{\mathbf{0}\} = L$. Therefore we should elaborate the logical connectives on L' applying the approaches described in Section 10.2. In most cases only the interval lattice $[0, 1]^I$ is considered (see, e.g., [45, 48–50]), and thus in this section we will deal with this special case only. To simplify the notation, we denote $[0, 1]^I = \mathcal{L}$.

Observe that the lattice \mathcal{L} is isomorphic to the lattice $L^* = \{(a, b) \mid a, b \in [0, 1], a + b \leq 1\}$, which is the background of the *intuitionistic fuzzy logic* introduced by Atanassov [51] (for a critical comment about the mathematical misuse of the word ‘intuitionistic’ in this context see [52]), and that the logical connectives on L^* are extensively discussed in [53, 54].

Each negation $N: [0, 1] \rightarrow [0, 1]$ induces a negation $\mathcal{N}: \mathcal{L} \rightarrow \mathcal{L}$ given by $\mathcal{N}([x, y]) = [N(y), N(x)]$, but not vice versa. However, for the strong negations on \mathcal{L} we have the following result (compare also [53, 54]).

Theorem 21. A mapping $\mathcal{N}: \mathcal{L} \rightarrow \mathcal{L}$ is a strong negation on \mathcal{L} if and only if there is a strong negation $N: [0, 1] \rightarrow [0, 1]$ such that $\mathcal{N}(x, y) = [N(y), N(x)]$.

On the basis of standard negation N_s on $[0, 1]$, we introduce the standard negation \mathcal{N}_s on \mathcal{L} given by $\mathcal{N}_s(x, y) = [1 - y, 1 - x]$.

Conjunction operators on \mathcal{L} are discussed, e.g., in [56, 55] (compare also [53, 54]). We can distinguish four classes of conjunction operators on \mathcal{L} :

($\mathcal{L}1$) *t-representable* conjunction operators $\mathcal{C}_{T_1, T_2}: \mathcal{L}^2 \rightarrow \mathcal{L}$, given by

$$\mathcal{C}_{T_1, T_2}([x, y], [u, v]) = [T_1(x, u), T_2(y, v)],$$

where $T_1, T_2: [0, 1]^2 \rightarrow [0, 1]$ are t-norms such that $T_1 \leq T_2$;

($\mathcal{L}2$) *lower pseudo-t-representable* conjunction operators $\mathcal{C}_T: \mathcal{L}^2 \rightarrow \mathcal{L}$, given by

$$\mathcal{C}_T([x, y], [u, v]) = [T(x, u), \max(T(x, v), T(u, y))],$$

where $T: [0, 1]^2 \rightarrow [0, 1]$ is a t-norm;

($\mathcal{L}3$) *upper pseudo-t-representable* conjunction operators $\mathcal{C}^T: \mathcal{L}^2 \rightarrow \mathcal{L}$, given by

$$\mathcal{C}^T([x, y], [u, v]) = [\min(T(x, v), T(u, y)), T(y, v)],$$

where $T: [0, 1]^2 \rightarrow [0, 1]$ is a t-norm;

($\mathcal{L}4$) *non-representable* conjunction operators, i.e., conjunction operators on \mathcal{L} not belonging to ($\mathcal{L}1$), neither to ($\mathcal{L}2$) nor to ($\mathcal{L}3$).

Observe that for any t-norms $T_1, T_2: [0, 1]^2 \rightarrow [0, 1]$ with $T_1 \leq T_2$, we have $\mathcal{C}_{T_1} \leq \mathcal{C}_{T_1, T_2} \leq \mathcal{C}^{T_2}$.

The strongest conjunction operator $\mathcal{C}^*: \mathcal{L}^2 \rightarrow \mathcal{L}$ is t-representable because of $\mathcal{C}^* = \mathcal{C}_{T_M, T_M}$, while the weakest conjunction operator $\mathcal{C}_*: \mathcal{L}^2 \rightarrow \mathcal{L}$ is non-representable. Note that taking the weakest t-norm T_D , the corresponding t-representable conjunction operator $\mathcal{C}_{T_D, T_D}: \mathcal{L}^2 \rightarrow \mathcal{L}$ fulfills $\mathcal{C}_{T_D, T_D}([a, 1], [a, 1]) = [0, 1]$ for all $a \in [0, 1]$, while $\mathcal{C}_*([a, 1], [a, 1]) = [0, 0]$ whenever $a \in [0, 1[$.

An interesting parametric class of conjunction operators $\mathcal{C}_{a,T}: \mathcal{L}^2 \rightarrow \mathcal{L}$ with $a \in [0, 1]$, where $T: [0, 1]^2 \rightarrow [0, 1]$ is a t-norm, is given by (see [56])

$$\mathcal{C}_{a,T}([x, y], [u, v]) = [T(x, u), \max(T(a, T(y, v)), T(x, v), T(u, y))].$$

Then $\mathcal{C}_{1,T} = \mathcal{C}_{T,T}$ is t-representable, $\mathcal{C}_{0,T} = \mathcal{C}_T$ is lower pseudo-t-representable, and for $a \in]0, 1[$, $\mathcal{C}_{a,T}$ is a non-representable conjunction operator on \mathcal{L} .

Observe that there are no divisible conjunction operators on \mathcal{L} , and thus \mathcal{L} cannot serve as a carrier for a BL-logic. Moreover, continuous conjunction operators on \mathcal{L} are not necessarily \vee -preserving. For example, the conjunction operator $\mathcal{C}: \mathcal{L}^2 \rightarrow \mathcal{L}$ given by

$$\mathcal{C}([x, y], [u, v]) = [\max(0, x + u - (1 - y)(1 - v) - 1), \max(0, y + v - 1)]$$

is continuous and non-representable. However, it is not \vee -preserving (see [54]).

Disjunction operators $\mathcal{D}: \mathcal{L}^2 \rightarrow \mathcal{L}$ can be derived from conjunction operators $\mathcal{C}: \mathcal{L}^2 \rightarrow \mathcal{L}$ by duality, e.g., applying the standard negation $\mathcal{N}_s: \mathcal{L} \rightarrow \mathcal{L}$, putting

$$\mathcal{D}([x, y], [u, v]) = \mathcal{N}_s(\mathcal{C}(\mathcal{N}_s([x, y]), \mathcal{N}_s([u, v]))).$$

Therefore we distinguish again four classes of disjunction operators on \mathcal{L} , namely,

($\mathcal{L}1'$) *t-representable* disjunction operators $\mathcal{D}_{S_1, S_2}: \mathcal{L}^2 \rightarrow \mathcal{L}$, given by

$$\mathcal{D}_{S_1, S_2}([x, y], [u, v]) = [S_1(x, u), S_2(y, v)],$$

where $S_1, S_2: [0, 1]^2 \rightarrow [0, 1]$ are t-conorms such that $S_1 \leq S_2$;

($\mathcal{L}2'$) *lower pseudo-t-representable* disjunction operators $\mathcal{D}_S: \mathcal{L}^2 \rightarrow \mathcal{L}$, given by

$$\mathcal{D}_S([x, y], [u, v]) = [S(x, u), \max(S(x, v), S(u, y))],$$

where $S: [0, 1]^2 \rightarrow [0, 1]$ is a t-conorm;

($\mathcal{L}3'$) *upper pseudo-t-representable* disjunction operators $\mathcal{D}^S: \mathcal{L}^2 \rightarrow \mathcal{L}$, given by

$$\mathcal{D}^S([x, y], [u, v]) = [\min(S(x, v), S(u, y)), S(y, v)],$$

where $S: [0, 1]^2 \rightarrow [0, 1]$ is a t-conorm;

($\mathcal{L}4'$) *non-representable* disjunction operators, i.e., disjunction operators on \mathcal{L} not belonging to ($\mathcal{L}1'$), neither to ($\mathcal{L}2'$), nor to ($\mathcal{L}3'$).

Note that the class ($\mathcal{L}1'$) is dual to ($\mathcal{L}1$), ($\mathcal{L}2'$) is dual to ($\mathcal{L}3$), ($\mathcal{L}3'$) is dual to ($\mathcal{L}2$), and ($\mathcal{L}4'$) is dual to ($\mathcal{L}4$).

Recall that the weakest disjunction operator $\mathcal{D}_*: \mathcal{L}^2 \rightarrow \mathcal{L}$ is t-representable, $\mathcal{D}_* = \mathcal{D}_{S_M, S_M}$, while the strongest disjunction operator $\mathcal{D}^*: \mathcal{L}^2 \rightarrow \mathcal{L}$ is non-representable.

We introduce here the parametric class $\mathcal{D}_{a,S}$ with $a \in [0, 1]$ of disjunction operators on \mathcal{L} generated by a t-conorm $S: [0, 1]^2 \rightarrow [0, 1]$ and given by

$$\mathcal{D}_{a,S}([x, y], [u, v]) = [\min(S(a, S(x, u)), S(u, y), S(x, v)), S(y, v)].$$

Observe that $\mathcal{D}_{a,S}$ is \mathcal{N}_s -dual to $\mathcal{C}_{1-a,T}$ whenever T is a t-norm which is N_s -dual to S . Moreover, $\mathcal{D}_{0,S} = \mathcal{D}_{S,S}$ is a t-representable, $\mathcal{D}_{1,S} = \mathcal{D}^S$ is an upper pseudo-t-representable, and for $a \in]0, 1[$, $\mathcal{D}_{a,S}$ is a non-representable disjunction operator.

Among several types of implication operators on \mathcal{L} , we recall the two of them discussed in Section 10.2.

For a conjunction operator $\mathcal{C}: \mathcal{L}^2 \rightarrow \mathcal{L}$, the corresponding residual implication $\mathcal{R}_\mathcal{C}: \mathcal{L}^2 \rightarrow \mathcal{L}$ is given by

$$\mathcal{R}_\mathcal{C}([x, y], [u, v]) = \bigvee \{[\alpha, \beta] \in \mathcal{L} \mid \mathcal{C}([x, y], [\alpha, \beta]) \leq [u, v]\}.$$

Some examples of residual implications $\mathcal{R}_\mathcal{C}$ are given as follows:

$$\mathcal{R}_{\mathcal{C}_{T_1, T_2}}([x, y], [u, v]) = [\min(R_{T_1}(x, u), R_{T_2}(y, v)), R_{T_2}(y, v)],$$

$$\mathcal{R}_{\mathcal{C}_T}([x, y], [u, v]) = [\min(R_T(x, u), R_T(y, v)), R_T(x, v)],$$

$$\mathcal{R}_{\mathcal{C}^T}([x, y], [u, v]) = [\min(R_T(x, u), R_T(y, v)), R_T(y, v)].$$

Recall that the mapping $\mathcal{N}_\mathcal{C}: \mathcal{L} \rightarrow \mathcal{L}$ given by $\mathcal{N}_\mathcal{C}([x, y]) = \mathcal{R}_\mathcal{C}([x, y], [0, 0])$ is a negation on \mathcal{L} for any conjunction operator $\mathcal{C}: \mathcal{L}^2 \rightarrow \mathcal{L}$.

The \mathcal{D} -implication $\mathcal{I}_{\mathcal{N}, \mathcal{D}}: \mathcal{L}^2 \rightarrow \mathcal{L}$ is given by

$$\mathcal{I}_{\mathcal{N}, \mathcal{D}}([x, y], [u, v]) = \mathcal{D}(\mathcal{N}([x, y]), [u, v]),$$

where \mathcal{D} is a disjunction operator on \mathcal{L} and \mathcal{N} is a negation on \mathcal{L} . Some examples of \mathcal{D} -implications on \mathcal{L} are as follows:

$$\mathcal{I}_{\mathcal{N}_s, \mathcal{D}_*}([x, y], [u, v]) = [\max(1 - y, u), \max(1 - x, v)],$$

$$\mathcal{I}_{\mathcal{N}_s, \mathcal{D}_{S_1, S_2}}([x, y], [u, v]) = [I_{N_s, S_1}(y, u), I_{N_s, S_2}(x, v)],$$

$$\mathcal{I}_{\mathcal{N}, \mathcal{D}_S}([x, y], [u, v]) = [I_{N_s}(y, u), \max(I_{N_s}(y, v), I_{N_s}(x, u))],$$

$$\mathcal{I}_{\mathcal{N}_s, \mathcal{D}_{S_L}}([x, y], [u, v]) = [\min(I_{N_s, S_L}(x, u), I_{N_s, S_L}(y, v)), I_{N_s, S_L}(x, v)],$$

where $\mathcal{N}([x, y]) = [N(y), N(x)]$ (compare Theorem 21).

Observe that $\mathcal{N}_{\mathcal{C}_{T_L}} = \mathcal{N}_s$ and that $\mathcal{I}_{\mathcal{N}_s, \mathcal{D}^{S_L}} = \mathcal{R}_{\mathcal{C}_{T_L}}$, where the upper pseudo-t-representable disjunction operator \mathcal{D}^{S_L} is \mathcal{N}_s -dual to the lower pseudo-t-representable conjunction operator \mathcal{C}_{T_L} . Moreover, all these operators are continuous, thus copying the properties of Lukasiewicz operators in $[0, 1]$ -valued logics.

10.7 Logical Connectives on Other Lattices of Truth-Values

We have already seen in Section 10.6 that logical connectives on more complex lattices might, but need not, be related to logical connectives on basic resp. underlying lattices. Therefore, we now particularly turn to such lattice structures L which are built from a family of given lattices L_k with $k \in \mathcal{K}$ and discuss the corresponding logical connectives. We will see that although we can always look at that new lattice L independently of the originally given lattices L_k and of the applied construction method (compare, e.g., the approach of Goguen [57] to L -fuzzy sets), several logical connectives on L can be derived from the corresponding logical connectives on the lattices L_k . In particular, we will focus on logical connectives on Cartesian products of lattices, horizontal as well as ordinal sums of lattices.

Cartesian Products

The most common construction method is the *Cartesian product*. Therefore, consider a system $(L_k)_{k \in \mathcal{K}}$ of lattices and put

$$L = \prod_{k \in \mathcal{K}} L_k = \{(x_k)_{k \in \mathcal{K}} \mid x_k \in L_k\}.$$

The lattice operations on L are defined coordinatewise, and $\mathbf{1} = (\mathbf{1}_k)_{k \in \mathcal{K}}$ and $\mathbf{0} = (\mathbf{0}_k)_{k \in \mathcal{K}}$ are its top and bottom elements, respectively, such that L is again a bounded lattice.

Evidently, for any kind of logical connectives on L , we can derive them from the corresponding logical connectives on L_k 's. However, not each logical connective on L can be constructed coordinatewise.

For example, let $L = L_1 \times L_2$ and $N_i: L_i \rightarrow L_i$ be a (strong) negation on L_i with $i \in \{1, 2\}$. Then also $N: L \rightarrow L$ given by $N(x_1, x_2) = (N_1(x_1), N_2(x_2))$ is a (strong) negation. However, if $L_1 = L_2$ and $N_1 = N_2$, then also $N': L \rightarrow L$ given by $N'(x_1, x_2) = (N_1(x_2), N_1(x_1))$ is a (strong) negation on L . Also the strongest (weakest) negation $N^*: L \rightarrow L$ ($N_*: L \rightarrow L$) is not built coordinatewise.

Conjunction operators on product lattices were discussed in [25] (see also [58, 59]). The strongest conjunction operator $C^*: L^2 \rightarrow L$ is derived coordinatewise from the strongest conjunction operators $C_k^*: L_k^2 \rightarrow L_k$, $k \in \mathcal{K}$, contrary to the the weakest conjunction operator $C_*: L^2 \rightarrow L$. Sufficient conditions ensuring that a conjunction operator $C: L^2 \rightarrow L$ is defined coordinatewise, i.e.,

$$C((x_k)_{k \in \mathcal{K}}, (y_k)_{k \in \mathcal{K}}) = (C_k(x_k, y_k))_{k \in \mathcal{K}},$$

are the \vee -preserving property or the \wedge -preserving property of C (see [25]). The situation with the disjunction operators and the implication operators is the same.

Horizontal and Ordinal Sums

Similar is the situation with *horizontal* and *ordinal sums* of lattices.

Definition 22. Let $(L_k)_{k \in \mathcal{K}}$ be a system of lattices such that $(L_k \setminus \{\mathbf{0}_k, \mathbf{1}_k\})_{k \in \mathcal{K}}$ is a pairwise disjoint system. Then

1. If $\mathbf{1}_k = \mathbf{1}$ and $\mathbf{0}_k = \mathbf{0}$ for all $k \in \mathcal{K}$, then the horizontal sum of lattices $(L_k)_{k \in \mathcal{K}}$ is the lattice $L = \bigcup_{k \in \mathcal{K}} L_k$ with top element $\mathbf{1}$ and bottom element $\mathbf{0}$. Moreover, $x \leq y$ if and only if $x, y \in L_k$ for some $k \in \mathcal{K}$ and $x \leq_k y$ (i.e., non-extremal elements from different L_k 's are incomparable).
2. If \mathcal{K} is a linearly ordered set with top element k^* and bottom element k_* and if, for $k_1 < k_2$, $x \in L_{k_1} \cap L_{k_2}$ implies $x = \mathbf{1}_{k_1} = \mathbf{0}_{k_2}$, then the ordinal sum of lattices $(L_k)_{k \in \mathcal{K}}$ is the lattice $L = \bigcup_{k \in \mathcal{K}} L_k$ with top element $\mathbf{1} = \mathbf{1}_{k^*}$, and bottom element $\mathbf{0} = \mathbf{0}_{k_*}$ as well as $x \leq y$ if either $x, y \in L_k$ and $x \leq_k y$ for some $k \in \mathcal{K}$, or $x \in L_{k_1}$, $y \in L_{k_2}$ and $k_1 < k_2$.

Observe that the only non-trivial product lattice coinciding with a horizontal sum is the diamond lattice $L = L_{(1)} \times L_{(1)}$, which is also a horizontal sum of two $L_{(2)}$ lattices with common top and bottom elements.

Any system of (strong) negations $N_k: L_k \rightarrow L_k$ on lattices L_k induces a (strong) negation $N: L \rightarrow L$ on a horizontal sum lattice $L = \bigcup_{k \in \mathcal{K}} L_k$, but not vice versa. However, each strong negation $N: L \rightarrow L$ on such a lattice is characterized by an involutive permutation $\sigma: \mathcal{K} \rightarrow \mathcal{K}$, i.e., $\sigma \circ \sigma = \text{id}$, such that the lattices L_k and $L_{\sigma(k)}$ are isomorphic by means of isomorphisms $\varphi_k: L_k \rightarrow L_{\sigma(k)}$ for all $k \in \mathcal{K}$ and a system $N_k: L_k \rightarrow L_k$ of strong negations on L'_k 's, $k \in \mathcal{K}$, such that for all $k \in \mathcal{K}$, $x \in L_k$, we have

$$N(x) = \varphi_k(N_k(x)) = N_{\sigma(k)}(\varphi_k(x)).$$

Conjunction operators on horizontal as well as ordinal sums have been discussed, e.g., in [59, 60]. Each conjunction operator $C: L^2 \rightarrow L$ on a horizontal sum lattice $L = \bigcup_{k \in \mathcal{K}} L_k$ is characterized by a corresponding system $C_k: L_k^2 \rightarrow L_k$ of conjunction operators on L_k , $k \in \mathcal{K}$,

$$C(x, y) = \begin{cases} C_k(x, y) & \text{if } (x, y) \in L_k^2, \\ \mathbf{0} & \text{otherwise.} \end{cases} \quad (12)$$

The structure of disjunction operators $D: L^2 \rightarrow L$ on a horizontal sum lattice $L = \bigcup_{k \in \mathcal{K}} L_k$ is similar to (12), i.e., they are characterized by disjunction operators $D_k: L_k^2 \rightarrow L_k$, $k \in \mathcal{K}$, so that

$$D(x, y) = \begin{cases} D_k(x, y) & \text{if } (x, y) \in L_k^2, \\ \mathbf{1} & \text{otherwise.} \end{cases}$$

Given a conjunction operator C on a horizontal sum lattice L , the corresponding residual implication $R_C: L^2 \rightarrow L$ is given by

$$R_C(x, y) = \begin{cases} y & \text{if } x = \mathbf{1}, \\ \mathbf{1} & \text{otherwise,} \end{cases}$$

whenever L is a non-trivial horizontal sum (i.e., $\text{card } \mathcal{K} > 1$). Hence there are no divisible conjunction operators on a non-trivial horizontal sum lattice L .

Then, for a negation $N: L \rightarrow L$ given by $N(x) = N_k(x)$ whenever $x \in L_k$, where $N_k: L_k \rightarrow L_k$ is a negation on L_k , $k \in \mathcal{K}$, the corresponding D -implication $I_{N,D}: L^2 \rightarrow L$ is given by

$$I_{N,D}(x, y) = \begin{cases} I_{N_k, D_k}(x, y) & \text{if } (x, y) \in L_k^2, \\ y & \text{if } N(x) = \mathbf{0}, \\ \mathbf{1} & \text{otherwise,} \end{cases}$$

and if $N: L \rightarrow L$ is moreover a strong negation on L , then

$$I_{N,D}(x, y) = \begin{cases} I_{N_k, D_k}(x, y) & \text{if } (x, y) \in L_k^2, \\ \mathbf{1} & \text{otherwise.} \end{cases}$$

In the case of ordinal sums, negation operators on L are not related (up to some special cases) to negation operators on single summands L_k . Moreover, there are cases where single summands do not admit any strong negation; however, the ordinal sum lattice L possesses strong negations. This is, e.g., the case for $L_{(-\infty, \infty)}$, which is an ordinal sum of $L_1 = L_{(-\infty)}$ and $L_2 = L_{(\infty)}$.

However, turning to conjunction and disjunction operators on an ordinal sum lattice $L = \bigcup_{k \in \mathcal{K}} L_k$, the following holds: for any system $(C_k)_{k \in \mathcal{K}}$ of conjunction operators resp. any system $(D_k)_{k \in \mathcal{K}}$ of disjunction operators on L_k , $k \in \mathcal{K}$, the mapping $C: L^2 \rightarrow L$ given by

$$C(x, y) = \begin{cases} C_k(x, y) & \text{if } (x, y) \in L_k^2, \\ \min(x, y) & \text{otherwise,} \end{cases} \quad (13)$$

resp., the mapping $D: L \rightarrow L$ given by

$$D(x, y) = \begin{cases} D_k(x, y) & \text{if } (x, y) \in L_k^2, \\ \max(x, y) & \text{otherwise,} \end{cases}$$

is a conjunction resp. a disjunction operator on L (compare also (2), (5)). Observe that they are usually called *ordinal sums* of the corresponding operators. Note also that while C^* is an ordinal sum of $((C^*)_k)_{k \in \mathcal{K}}$, this is no more true in the case of the weakest conjunction operator C_* on L . Similarly, D_* is an ordinal sum of $((D_*)_k)_{k \in \mathcal{K}}$, but D^* is not an ordinal sum.

Finally recall that the residual implication $R_C: L^2 \rightarrow L$ related to an ordinal sum conjunction operator $C: L^2 \rightarrow L$ given by (13) is given by (compare with (6))

$$R_C(x, y) = \begin{cases} \mathbf{1} & \text{if } x \leq y, \\ R_{C_k}(x, y) & \text{if } x > y \text{ and } (x, y) \in L_k^2, \\ y & \text{otherwise.} \end{cases}$$

10.8 Conclusion

We have discussed logical connectives for different types of truth-value lattices relevant for dealing with information granules. Particular emphasis has been set on discrete chains, intervals, as well as interval-valued lattices. We have illustrated that depending on the underlying structure particular types of connectives, like strong negations or relationships between different connectives cannot be provided. Further, it has been demonstrated how connectives on constructed lattices can be related to connectives on underlying given lattices.

The diversity of models for granular computing presented here opens new possibilities for fitting a mathematical model to real data. A suitable tool for such a fitting is, e.g., the software package developed by Gleb Beliakov. For a free download, see <http://www.it.deakin.edu.au/~gleb>.

Acknowledgments

Radko Mesiar was supported by the grants VEGA 1/3006/06 and MSM VZ 6198898701, and Andrea Mesiarová-Zemánková by grant VEGA 2/7142/27. Susanne SamingerPlatz has been on a sabbatical year at the Dipartimento di Matematica ‘Ennio De Giorgi,’ Università del Salento (Italy), when writing large parts of this chapter. She therefore gratefully acknowledges the support by the Austrian Science Fund (FWF) in the framework of the Erwin Schrödinger Fellowship J 2636 ‘The Property of Dominance – From Products of Probabilistic Metric Space to (Quasi-)Copulas and Applications.’

References

- [1] L.A. Zadeh. Fuzzy sets and information granularity. In: M.M. Gupta, R.K. Ragade, and R.R. Yager (eds), *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, 1979, pp. 3–18.
- [2] L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Trans. Fuzzy Syst.* 4 (1996) 103–111.
- [3] L.A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.
- [4] L.A. Zadeh. From computing with numbers to computing with words – from manipulation of measurements to manipulation of perceptions. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* 46 (1999) 105–119.
- [5] L.A. Zadeh. Toward a logic of perceptions based on fuzzy logic. In: V. Novák and I. Perfilieva (eds), *Discovering the World With Fuzzy Logic*. Physica-Verlag, Heidelberg, 2000, pp. 4–28.
- [6] A. Bargiela and W. Pedrycz. *Granular Computing*. Kluwer Academic Publishers, Boston, 2003.
- [7] V. Novák. Granularity via properties: the logical approach. In: *Proceedings of the European Society for fuzzy Logic and Technology 2001*, Leicester, 2001, pp. 372–376.
- [8] W. Pedrycz (ed.). *Granular Computing*. Physica-Verlag, Heidelberg, 2001.
- [9] W. Pedrycz. From granular computing to computational intelligence and human-centric systems. Personal communication, 2007.
- [10] S.K. Pal, L. Polkowski, and A. Skowron (eds.). *Rough-Neural Computing. Techniques for Computing with Words*. Springer, Berlin, 2004.
- [11] L.A. Zadeh. From computing with numbers to computing with words – from manipulation of measurements to manipulation of perceptions. *Int. J. Appl. Math. Comput. Sci.* 12 (2002) 307–324.
- [12] P. Hájek. *Metamathematics of Fuzzy Logic*. Kluwer Academic Publishers, Dordrecht, 1998.
- [13] V. Novák, I. Perfilieva, and J. Močkoř. *Mathematical Principles of Fuzzy Logic*. Kluwer Academic Publishers, Norwell, 1999.
- [14] N.N. Karnik and J.M. Mendel. Operations on type-2 fuzzy sets. *Fuzzy Sets Syst.* 122 (2001) 327–348.
- [15] Q. Liang and J.M. Mendel. Interval type-2 fuzzy logic systems: theory and design. *IEEE Trans. Fuzzy Syst.* 8 (2000) 535–550.
- [16] E.P. Klement and R. Mesiar (eds.). *Logical, Algebraic, Analytic, and Probabilistic Aspects of Triangular Norms*. Elsevier, Amsterdam, 2005.
- [17] W. Pedrycz. *Fuzzy Control and Fuzzy Systems*. Technical Report 82 14. Delft University of Technology, Department of Mathematics, Delft, 1982.
- [18] M. Baczyński and J. Balasubramaniam. Fuzzy implications. *Studies in Fuzziness and Soft Computing*. Springer, Heidelberg, to appear.

- [19] U. Höhle. Commutative, residuated ℓ -monoids. In: U. Höhle and E.P. Klement (eds), *Non-Classical Logics and Their Applications to Fuzzy Subsets. A Handbook of the Mathematical Foundations of Fuzzy Set Theory*. Kluwer Academic Publishers, Dordrecht, 1995, pp. 53–106.
- [20] N.N. Morsi and E.M. Roshdy. Issues on adjointness in multiple-valued logics. *Inf. Sci.* 176 (2006) 2886–2909.
- [21] M. Miyakoshi and M. Shimbo. Solutions of composite fuzzy relational equations with triangular norms. *Fuzzy Sets Syst.* 16 (1985) 53–63.
- [22] S. Gottwald. *A Treatise on Many-Valued Logic. Studies in Logic and Computation*. Research Studies Press, Baldock, 2001.
- [23] J. Łukasiewicz. O logice trówartosciowej. *Ruch Filozoficzny* 5 (1920) 170–171. (English translation contained in L. Borkowski (ed.), *J. Łukasiewicz: Selected Works. Studies in Logic and Foundations of Mathematics*. North-Holland, Amsterdam, 1970.)
- [24] G. Mayor and J. Torrens. Triangular norms on discrete settings. In: E.P. Klement and R. Mesiar (eds), *Logical, Algebraic, Analytic, and Probabilistic Aspects of Triangular Norms*. Elsevier, Amsterdam, 2005, chapter 7, pp. 189–230.
- [25] B. De Baets and R. Mesiar. Triangular norms on product lattices. *Fuzzy Sets Syst.* 104 (1999) 61–75.
- [26] L. Godo and C. Sierra. A new approach to connective generation in the framework of expert systems using fuzzy logic. In: *Proceedings of 18th International Symposium on Multiple-Valued Logic, Palma de Mallorca*, IEEE Computer Society Press, Washington, DC, 1988, pp. 157–162.
- [27] G. Mayor and J. Torrens. On a class of operators for expert systems. *Int. J. Intell. Syst.* 8 (1993) 771–778.
- [28] A.H. Clifford. Naturally totally ordered commutative semigroups. *Am. J. Math.* 76 (1954) 631–646.
- [29] E.P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*. Kluwer Academic Publishers, Dordrecht, 2000.
- [30] L.A. Zadeh. Fuzzy sets. *Inf. Control* 8 (1965) 338–353.
- [31] Z. Pawlak. Rough sets. *Int. J. Comput. Inf. Sci.* 11 (1982) 341–356.
- [32] E. Trillas. Sobre funciones de negación en la teoría de conjuntos difusos. *Stochastica* 3 (1979) 47–60.
- [33] B. Schweizer and A. Sklar. Statistical metric spaces. *Pac. J. Math.* 10 (1960) 313–334.
- [34] B. Schweizer and A. Sklar. *Probabilistic Metric Spaces*. North-Holland, New York, 1983.
- [35] K. Menger. Statistical metrics. *Proc. Natl. Acad. Sci. USA* 8 (1942) 535–537.
- [36] C. Alsina, M.J. Frank, and B. Schweizer. *Associative Functions: Triangular Norms and Copulas*. World Scientific, Singapore, 2006.
- [37] E.P. Klement. Construction of fuzzy σ -algebras using triangular norms. *J. Math. Anal. Appl.* 85 (1982) 543–565.
- [38] C.M. Ling. Representation of associative functions. *Publ. Math. Debrecen* 12 (1965) 189–212.
- [39] P.S. Mostert and A.L. Shields. On the structure of semi-groups on a compact manifold with boundary. *Ann. Math. II Ser.* 65 (1957) 117–143.
- [40] J. Aczél. Sur les opérations définies pour des nombres réels. *Bull. Soc. Math. Fr.* 76 (1949) 59–64.
- [41] B. Schweizer and A. Sklar. Associative functions and abstract semigroups. *Publ. Math. Debrecen* 10 (1963) 69–81.
- [42] M.J. Frank. On the simultaneous associativity of $F(x, y)$ and $x + y - F(x, y)$. *Aeq. Math.* 19 (1979) 194–226.
- [43] D. Butnariu and E.P. Klement. *Triangular Norm-Based Measures and Games with Fuzzy Coalitions*. Kluwer Academic Publishers, Dordrecht, 1993.
- [44] R.R. Yager. On a general class of fuzzy connectives. *Fuzzy Sets Syst.* 4 (1980) 235–242.
- [45] S. Jenei. A survey on left-continuous t-norms and pseudo t-norms. In: E.P. Klement and R. Mesiar (eds), *Logical, Algebraic, Analytic, and Probabilistic Aspects of Triangular Norms*. Elsevier, Amsterdam, 2005, chapter 5, pp. 113–142.
- [46] P. Cintula, E.P. Klement, R. Mesiar, and M. Navara. Residuated logics based on strict triangular norms with an involutive negation. *Math. Log. Quart.* 52 (2006) 269–282.
- [47] R.E. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1966.
- [48] M.B. Gorzalczany. A method of inference in approximate reasoning based on interval-valued fuzzy sets. *Fuzzy Sets Syst.* 21 (1987) 1–17.
- [49] H.T. Nguyen and E. Walker. *A First Course in Fuzzy Logic*. CRC Press, Boca Raton, FL, 1997.
- [50] R. Sambuc. *Fonctions Φ -floues. Application à l'aide au diagnostic en pathologie thyroïdienne*. Ph.D. Thesis. Université de Marseille II, France, 1975.
- [51] K.T. Atanassov. *Intuitionistic Fuzzy Sets*. Physica-Verlag, Heidelberg, 1999.
- [52] D. Dubois, S. Gottwald, P. Hájek, J. Kacprzyk, and H. Prade. Terminological difficulties in fuzzy set theory – the case of ‘intuitionistic fuzzy sets.’ *Fuzzy Sets Syst.* 156 (2005) 485–491.
- [53] G. Deschrijver, C. Cornelis, and E.E. Kerre. On the representation of intuitionistic fuzzy t-norms and t-conorms. *IEEE Trans. Fuzzy Syst.* 12 (2004) 45–61.

-
- [54] G. Deschrijver and E.E. Kerre. Triangular norms and related operators in L^* -fuzzy set theory. In: E.P. Klement and R. Mesiar (eds), *Logical, Algebraic, Analytic, and Probabilistic Aspects of Triangular Norms*. Elsevier, Amsterdam, 2005, chapter 8, pp. 231–259.
 - [55] G. Deschrijver. The Archimedean property for t-norms in interval-valued fuzzy set theory. *Fuzzy Sets Syst.* 157 (2006) 2311–2327.
 - [56] G. Deschrijver. Archimedean t-norms in interval-valued fuzzy set theory. In: *Proceedings of Eleventh International Conference IPMU 2006, Information Processing and Management of Uncertainty in Knowledge-Based Systems, Paris*, Vol. 1, Éditions EDK, Paris, 2006, pp. 580–586.
 - [57] J.A. Goguen. The logic of inexact concepts. *Synthese* 19 (1968/69) 325–373.
 - [58] S. Jenei and B. De Baets. On the direct decomposability of t-norms over direct product lattices. *Fuzzy Sets Syst.* 139 (2003) 699–707.
 - [59] S. Saminger. On ordinal sums of triangular norms on bounded lattices. *Fuzzy Sets Syst.* 157 (2006) 1403–1416.
 - [60] S. Saminger, E.P. Klement, and R. Mesiar. On extensions of triangular norms on bounded lattices. Submitted for publication.

11

Calculi of Information Granules: Fuzzy Relational Equations

Siegfried Gottwald

11.1 Introduction

The earliest and most paradigmatic examples of granular computing are the fuzzy control approaches which are based on finite lists of linguistic control rules, each of which has a finite number of fuzzy input values and a fuzzy output value – each of them a typical information granule.

In engineering science, fuzzy control methods have become a standard tool, which allows to apply computerized control approaches to a wider class of problems as those which can be reasonably and effectively treated with the more traditional mathematical methods like the Proportional-Derivative (PD) or Proportional-Integral-Derivative (PID) control strategies.

For an industrial engineer, usually success in control applications is the main criterion. Then he or she even accepts methods which have, to a larger extent, only a heuristic basis. And this has been the situation with fuzzy control approaches for a considerable amount of time. Particularly with respect to the linguistic control rules which are constitutive for a lot of fuzzy control approaches.

Of course, success in applications then calls for mathematical reflections about and mathematical foundations for the methods under consideration. For linguistic control rules their transformation into fuzzy relation equations has been the core idea in a lot of such theoretical reflections.

Here we discuss this type of a mathematical treatment of rule-based fuzzy control, which has the problem of the solvability of systems of fuzzy relation equations as its core, with particular emphasis on some more recent viewpoints which tend toward a more general view at this mathematization.

One of these rather new points studied here, first in Section 11.7, is the idea of a certain iteration of different methods to determine pseudosolutions of such systems, methods which aim at finding approximate solutions. But also the same method may be iterated and one may ask for a kind of ‘stability’ in this case, as is done in the more general context of Section 11.16.

Another new point of view is to look at this control problem as an interpolation problem.

And finally, a third new idea is the treatment of the whole standard approach toward fuzzy control from a more general point of view, which understands the usual compositional rule of inference as a particular inference mechanism which is combined with further aggregation operations.

Some of the results which are collected here have been described in previous papers of the author and some of his coauthors, particularly in [1–4].

11.2 Preliminaries

We use in this chapter a set-theoretic notation for fuzzy sets, which refers to a logic with truth degree set $[0, 1]$ based on a left-continuous t -norm t , or – more general – based on a class of (complete) prelinear residuated lattices with semigroup operation $*$. This means that we consider the logic MTL of left-continuous t -norms as the formal background (cf. [5]).

This logic has as standard connectives two conjunctions and a disjunction:

$$\& = *, \quad \wedge = \min, \quad \vee = \max.$$

In the lattice case we mean, by a slight abuse of language, by \min , \max the lattice meet, and the lattice join, respectively.

It also has an *implication* \rightarrow characterized by the adjointness condition

$$u * v \leq w \quad \text{iff} \quad u \leq (v \rightarrow w),$$

as well as a *negation* – given by $-H = H \rightarrow 0$.

The *quantifiers* \forall and \exists mean the *infimum* and *supremum*, respectively, of the truth degrees of all instances.

And the truth degree 1 is the only designated one. Therefore *logical validity* $\models H$ means that H has always truth degree 1.

The shorthand notation $\llbracket H \rrbracket$ denotes the truth degree of formula H , assuming that the corresponding evaluation of the (free) variables, as well as the model under consideration (in the first-order case), is clear from the context.

The class term notation $\{x \mid H(x)\}$ denotes the fuzzy set A with

$$\mu_A(a) = A(a) = \llbracket H(a) \rrbracket \quad \text{for each } a \in \mathbf{X}.$$

Occasionally we also use graded identity relations \equiv and \equiv^* for fuzzy sets, based on the graded inclusion relation

$$A \subseteq B = \forall x(A(x) \rightarrow B(x)),$$

and defined as

$$\begin{aligned} A \equiv B &= A \subseteq B \quad \& \quad B \subseteq A, \\ A \equiv^* B &= A \subseteq B \quad \wedge \quad B \subseteq A. \end{aligned}$$

Obviously one has the relationships

$$\begin{aligned} \models B_i \equiv^* B_j &\leftrightarrow \forall y(B_i(y) \leftrightarrow B_j(y)), \\ \models A \equiv B &\rightarrow A \equiv^* B. \end{aligned}$$

11.3 Fuzzy Control and Relation Equations

The standard paradigm of rule-based fuzzy control is that one supposes to have given, in a granular way, an incomplete and fuzzy description of a control function Φ from an input space \mathbf{X} to an output space \mathbf{Y} , realized by a finite family

$$\mathcal{D} = ((A_i, B_i))_{1 \leq i \leq n} \tag{1}$$

of (fuzzy) input–output data pairs. These granular data are supposed to characterize this function Φ sufficiently well.

In the usual approaches such a family of input–output data pairs is provided by a finite list

$$\text{IF } \alpha \text{ is } A_i, \text{ THEN } \beta \text{ is } B_i, \quad i = 1, \dots, n, \quad (2)$$

of linguistic control rules, also called fuzzy IF–THEN rules, describing some control procedure with input variable α and output variable β .

Mainly in engineering papers one often also consider the case of different input variables $\alpha_1, \dots, \alpha_n$; in this case the linguistic control rules become of the form

$$\text{IF } \alpha_1 \text{ is } A_1^1, \text{ AND } \dots \text{ AND } \alpha_n \text{ is } A_n^n, \text{ THEN } \beta \text{ is } B_i, \quad i = 1, \dots, n.$$

But from a mathematical point of view such rules are equivalent to the former ones: one simply has to allow as the input universe for α the cartesian product of the input universes of $\alpha_1, \dots, \alpha_n$.

Let us assume for simplicity that all the input data A_i are *normal*; i.e., there is a point x_0 in the universe of discourse with $A_i(x_0) = 1$. Sometimes even weak normality would suffice; i.e., the supremum over all the membership degrees of the A_i equals 1; but we do not indent to discuss this in detail.

The main mathematical problem of fuzzy control, besides the engineering problem to get a suitable list of linguistic control rules for the actual control problem, is therefore the interpolation problem to find a function $\Phi^* : \text{IF}(\mathcal{X}) \rightarrow \text{IF}(\mathcal{Y})$, which interpolates these data, i.e., which satisfies

$$\Phi^*(A_i) = B_i \quad \text{for each } i = 1, \dots, n, \quad (3)$$

and which, in this way, gives a fuzzy representation for the control function Φ .

Actually the standard approach is to look for *one* single function, more precisely: for some uniformly defined function, which should interpolate all these data, and which should be globally defined over $\text{IF}(\mathcal{X})$, or at least over a suitably chosen sufficiently large subclass of $\text{IF}(\mathcal{X})$.

11.3.1 The Compositional Rule of Inference

Following the basic ideas of Zadeh [6], such a fuzzy controller is formally realized by a fuzzy relation R which connects fuzzy input information A with fuzzy output information B via the *compositional rule of inference* (CRI):

$$B = A \circ R = R''A = \{y \mid \exists x(A(x) \& R(x, y))\}. \quad (4)$$

Therefore, applying this idea to the linguistic control rules themselves, these rules in a natural way become transformed into fuzzy relation equations

$$A_i \circ R = B_i, \quad \text{for } i = 1, \dots, n, \quad (5)$$

i.e., form a system of such relation equations.

The problem, to determine a fuzzy relation R which realizes via (4) such a list (2) of linguistic control rules, therefore becomes the problem to determine a solution of the corresponding system (5) of relation equations.

This problem proves to be a rather difficult one: it often happens that a given system (5) of relation equations is unsolvable. This is already the case in the more specific situation that the membership degrees belong to a Boolean algebra, as discussed (as a problem for Boolean matrices), e.g., in [7].

Nice solvability criteria are still largely unknown. Thus the investigation of the structure of the solution space for (5) was one of the problems discussed rather early. One essentially has that this space is an upper semilattice under the simple set union determined by the maximum of the membership degrees (cf., e.g., [8]).

And this semilattice has, if it is nonempty, a universal upper bound. To state the main result, one has to consider the particular fuzzy relation

$$\widehat{R} = \bigcap_{i=1}^n \{(x, y) \mid A_i(x) \rightarrow B_i(y)\}. \quad (6)$$

Theorem 1. *The system (5) of relation equations is solvable iff the fuzzy relation \widehat{R} is a solution of it. And in the case of solvability, \widehat{R} is always the largest solution of the system (5) of relation equations.*

This result was first stated by Sanchez [9] for the particular case of the min-based Gödel implication \rightarrow in (6), and generalized to the case of the residuated implications based on arbitrary left-continuous t-norms – and hence to the present situation – by the author in [10] (cf. also his [11]).

11.3.2 Modeling Strategies

Besides the reference to the CRI in this type of approach toward fuzzy control, the crucial point is to determine a fuzzy relation out of a list of linguistic control rules.

The fuzzy relation \widehat{R} can be seen as a formalization of the idea that the list (2) of control rules has to be read as

IF input is A_1 THEN output is B_1
AND
...
AND
IF input is A_n THEN output is B_n .

Having in mind such a formalization of the list (2) of control rules, there is immediately also another way of how to read this list:

input is A_1 AND output is B_1
OR
...
OR
input is A_n AND output is B_n .

It is this understanding of the list of linguistic control rules as a (rough) description of a *fuzzy function* which characterizes the approach of Mamdani and Assilian [12]. Therefore they consider instead of \widehat{R} the fuzzy relation

$$R_{MA} = \bigcup_{i=1}^n (A_i \times B_i),$$

again combined with the compositional rule of inference.

11.4 Toward a Solvability Criterion for R_{MA}

Having in mind Theorem 1, one is immediately confronted with the following:

Problem.. Under which conditions is the fuzzy relation R_{MA} a solution of the corresponding system of relation equations.

This problem is discussed in [13]. And one of the main results is the next theorem.

Theorem 2. Let all the input sets A_i be normal. Then the fuzzy relation R_{MA} is a solution of the corresponding system of fuzzy relation equations iff for all $i, j = 1, \dots, n$, one has

$$\models \exists x(A_i(x) \& A_j(x)) \rightarrow B_i \equiv^* B_j. \quad (7)$$

This MA-solvability criterion (7) is a kind of functionality of the list of linguistic control rules, at least in the case of the presence of an involutive negation: because in such a case one has

$$\models \exists x(A_i(x) \& A_j(x)) \leftrightarrow A_i \cap_t A_j \not\equiv^* \emptyset,$$

and thus condition (7) becomes

$$\models A_i \cap_t A_j \not\equiv^* \emptyset \rightarrow B_i \equiv^* B_j. \quad (8)$$

And this can be understood as a fuzzification of the following idea:

'if A_i and A_j coincide to some degree, then also B_i and B_j should coincide to a certain degree.'

Of course, this fuzzification is neither obvious nor completely natural, because it translates 'degree of coincidence' in two different ways.

Corollary 3. If condition (8) is satisfied, then the system of relation equations has \widehat{R} as a solution.

This leads back to the well-known result, explained, e.g., in [11], that the system of relation equations is solvable in the case that all the input fuzzy sets A_i are pairwise t -disjoint:

$$A_i \cap_t A_j = \emptyset \quad \text{for all } i \neq j.$$

It is furthermore known, cf. again [11], that functionality holds true for the relational composition at least in the form

$$\models A \equiv B \rightarrow A \circ R \equiv B \circ R,$$

because one has the (generalized) monotonicity

$$\models A \subseteq B \rightarrow A \circ R \subseteq B \circ R.$$

This, by the way, gives the following corollary.

Corollary 4. A necessary condition for the solvability of a system of relation equations is that one always has

$$\models A_i \equiv A_j \rightarrow B_i \equiv B_j.$$

This condition is symmetric in i, j . Therefore one gets as a slight generalization also the following corollary.

Corollary 5. Let all the input sets A_i be normal. Then the fuzzy relation R_{MA} is a solution of our system of fuzzy relation equations iff for all $i, j = 1, \dots, n$, one has

$$\models A_i \cap_t A_j \not\equiv^* \emptyset \rightarrow B_i \subseteq B_j.$$

11.5 Relating R_{MA} with the Largest Solution

However, it may happen that the system of relation equations is solvable, i.e., has \widehat{R} as a solution, without having the fuzzy relation R_{MA} as a solution.

An example is given in [3].

Therefore Klawonn's condition (7) is a sufficient one only for the solvability of the system (5) of relation equations.

Hence one has as a new problem to give additional assumptions, besides the solvability of the system (5) of relation equations, which are sufficient to guarantee that R_{MA} is a solution of (5).

As in [11] and already in [14], we subdivide the problem whether a fuzzy relation R is a solution of the system of relation equations into two cases.

Definition 1. A fuzzy relation R has the subset property w.r.t. a system (5) of relation equations iff one has

$$A_i \circ R \subseteq B_i, \quad \text{for } i = 1, \dots, n, \quad (9)$$

and it has the superset property w.r.t. (5) iff one has

$$A_i \circ R \supseteq B_i, \quad \text{for } i = 1, \dots, n. \quad (10)$$

Particularly for R_{MA} quite natural sufficient conditions for the superset property have been given, but only rather strong ones for the subset property.

Proposition 6. If all input sets A_i are normal then R_{MA} has the superset property.

So we know with the fuzzy relation R_{MA} , assuming that all input sets A_i are normal, at least one upper approximation of the (possible) solution for the system of relation equations.

Proposition 7. If all input sets are pairwise disjoint (under \cap_t), then R_{MA} has the subset property.

It is also of interest to ask for conditions under which the relation \widehat{R} satisfies these properties. Fortunately, for the subset property there is a nice answer.

Proposition 8. \widehat{R} has the subset property.

Together with Proposition 6 this immediately gives the following.

Corollary 9. If all input sets A_i are normal then one has for all indices i the inclusions

$$A_i \circ \widehat{R} \subseteq B_i \subseteq A_i \circ R_{\text{MA}}. \quad (11)$$

Thus we know with the fuzzy relation \widehat{R} at least one lower approximation of the (possible) solution for the system of relation equations. However, the single inclusion relations (11) can already be proved from slightly weaker assumptions.

Proposition 10. If the input set A_k is normal then $A_k \circ \widehat{R} \subseteq B_k \subseteq A_k \circ R_{\text{MA}}$.

So we know that with normal input sets the fuzzy outputs $A_i \circ \widehat{R}$ are always subsets of $A_i \circ R_{\text{MA}}$. Furthermore, we immediately have the following global result.

Proposition 11. If all the input sets A_i of the system of relation equations are normal and if one also has $R_{\text{MA}} \subseteq \widehat{R}$, then the system of relation equations is solvable, and R_{MA} is a solution.

Now we ask for conditions under which the relation R_{MA} maps the input fuzzy sets A_i to subsets of $A_i \circ \widehat{R}$. And that means to again ask for some conditions which give the subset property of R_{MA} and thus the solvability of the system of relation equations.

Proposition 12. Assume the normality of all the input sets A_i . Then to have for some index $1 \leq k \leq n$,

$$A_k \circ R_{\text{MA}} \subseteq A_k \circ \widehat{R}$$

is equivalent to the equality

$$A_k \circ R_{\text{MA}} = A_k \circ \widehat{R} = B_k.$$

Corollary 13. Assume the normality of all the input sets A_i . Then the condition to have for all indices $1 \leq i \leq n$

$$A_i \circ R_{\text{MA}} \subseteq A_i \circ \widehat{R}$$

is equivalent to the fact that R_{MA} is a solution of the system of relation equations, and hence equivalent to the second criterion of Klawonn.

11.6 Toward the Superset Property of \widehat{R}

The solvability of the system of relation equations is equivalent to the fact that the relation \widehat{R} is a solution. Therefore the solvability of our system of relation equations is also equivalent to the fact that \widehat{R} has the subset as well as the superset properties.

Now, as seen in Proposition 8, the subset property is generally satisfied for the fuzzy relation \widehat{R} . This means we immediately have the following.

Corollary 14. A system of relation equations is solvable iff its relation \widehat{R} has the superset property.

Hence, to get sufficient solvability conditions for the system of relation equations means to look for sufficient conditions for this superset property of \widehat{R} .

And this seems to be an astonishingly hard problem.

What one immediately has in general are the equivalences:

$$\models B_k \subseteq A_k \circ \widehat{R}$$

iff for all y

$$\models B_k(y) \rightarrow \exists x(A_k(x) \& \bigwedge_i (A_i(x) \rightarrow B_i(y)))$$

iff for all y and all i

$$\models B_k(y) \rightarrow \exists x(A_k(x) \& (A_i(x) \rightarrow B_i(y))). \quad (12)$$

And just this last condition offers the main open **problem**: to find suitable conditions which are equivalent to (12).

Particularly for $i = k$ and continuous t -norms this is equivalent to

$$\models B_k(y) \rightarrow \exists x(A_k(x) \wedge B_k(y)).$$

Corollary 15. For continuous t -norms t , a necessary condition for the superset property of \widehat{R} is that

$$\text{hgt}(B_k) \leq \text{hgt}(A_k)$$

holds for all input-output pairs (A_k, B_k) .

Part of the present problem is to look for *sufficient* conditions which imply (12).

Here a nice candidate seems to have for given i, k , and y the existence of some x with

$$\models B_k(y) \rightarrow A_k(x) \text{ & } (A_i(x) \rightarrow B_i(y)).$$

Routine calculations show that this means that it is sufficient for (12) to have, for a given y , either the existence of some x with

$$B_k(y) \leq A_k(x) \quad \text{and} \quad A_i(x) \leq B_i(y)$$

or the existence of some x with

$$A_k(x) = 1 \quad \text{and} \quad A_i(x) \leq \llbracket B_k(y) \rightarrow B_i(y) \rrbracket.$$

However, both these sufficient conditions do not look very promising.

11.7 Getting New Pseudosolutions

Suppose again that all the input sets A_i are normal.

The standard strategy to ‘solve’ such a system of relation equations is to refer to its Mamdani–Assilian relation R_{MA} and to apply, for a given fuzzy input A , the CRI, i.e., to treat the fuzzy set $A \circ R_{MA}$ as the corresponding, ‘right’ output.

Similarly one can ‘solve’ the system of relation equations with reference to its possible largest solution \widehat{R} and to the CRI, which means to treat for any fuzzy input A the fuzzy set $A \circ \widehat{R}$ as its ‘right’ output.

But both these ‘solution’ strategies have the (at least theoretical) disadvantage that they may give insufficient results, at least for the predetermined input sets.

Thus R_{MA} and \widehat{R} may be considered as *pseudosolutions*. Call \widehat{R} the *maximal* and R_{MA} the *MA-pseudosolution*.

As was mentioned previously, these pseudosolutions R_{MA} and \widehat{R} are upper and lower approximations for the realizations of the linguistic control rules.

Now one may equally well look for new pseudosolutions, e.g., by some *iteration* of these pseudosolutions in the way that for the *next iteration step* in such an iteration process the system of relation equations is changed such that its (new) output sets become the *real output* of the *former* iteration step. This has been done in [3].

To formulate the dependence of the pseudosolutions R_{MA} and \widehat{R} from the input and output data, we denote the ‘original’ pseudosolutions with the input–output data (A_i, B_i) in another way and write

$$R_{MA}[B_k] \text{ for } R_{MA}, \quad \widehat{R}[B_k] \text{ for } \widehat{R}.$$

Using the fact that for a given solvable system of relation equations its maximal pseudosolution \widehat{R} is really a solution one immediately gets.

Proposition 16. *For any fuzzy relation S one has for all i*

$$A_i \circ \widehat{R}[A_k \circ S] = A_i \circ S.$$

Hence it does not give a new pseudosolution if one iterates the solution strategy of the maximal, i.e., Sanchez pseudosolution after some (other) pseudosolution.

The situation changes if one uses the Mamdani–Assilian solution strategy after another pseudosolution strategy. Because R_{MA} has the superset property, one should use it for an iteration step which follows a pseudosolution step w.r.t. a fuzzy relation which has the subset property, e.g., after the strategy using \widehat{R} . This gives, cf. again [3], the following result.

Theorem 17. One always has

$$A_i \circ \widehat{R}[B_k] \subseteq A_i \circ R_{\text{MA}}[A_k \circ \widehat{R}[B_k]] \subseteq A_i \circ R_{\text{MA}}[B_k].$$

Thus the iterated relation $R_{\text{MA}}[A_k \circ \widehat{R}]$ is a **better** pseudosolution as each one of R_{MA} and \widehat{R} .

11.8 Approximation and Interpolation

The standard mathematical understanding of *approximation* is that by an approximation process some mathematical object A , e.g., some function, is approximated, i.e., determined within some (usually previously unspecified) error bounds.

Additionally one assumes that the approximating object B for A is of some predetermined, usually ‘simpler,’ kind, e.g., a polynomial function.

So one may approximate some transcendental function, e.g., the trajectory of some non-linear process by a piecewise linear function or by a polynomial function of some bounded degree. Similarly one approximates, e.g., in the Runge–Kutta methods the solution of a differential equation by a piecewise linear function, or one uses splines to approximate a difficult surface in 3-space by planar pieces.

The standard mathematical understanding of *interpolation* is that a function f is only partially given by its values *at some points* of the domain of the function, the interpolation nodes.

The problem then is to determine ‘the’ values of f for all the other points of the domain (usually *between* the interpolation nodes – sometimes also outside these interpolation nodes (extrapolation).

And this is usually done in such a way that one considers groups of neighboring interpolation nodes which uniquely determine an interpolating function of some predetermined type within their convex hull (or something like): a function which has the interpolation nodes of the actual group as argument–value pairs – and which in this sense *locally* approximates the function f .

In the *standard fuzzy control approach* the input–output data pairs of the linguistic control rules just provide interpolation nodes.

However, what is lacking – at least up to now – is the idea of a *local approximation* of the intended crisp control function by some fuzzy function. Instead, in the standard contexts one always asks for something like a *global* interpolation; i.e., one is interested in interpolating *all* nodes by *only one* interpolation function.

To get a local approximation of the intended crisp control function Φ , one needs some notion of ‘nearness’ or of ‘neighboring’ for fuzzy data granules. Such a notion is lacking in general.

For the particular case of a linearly ordered input universe \mathbf{X} , and the additional assumption that the fuzzy input data are unimodal, one gets in a natural way from this crisp background a notion of neighboring interpolation nodes: fuzzy nodes are neighboring if their kernel points are.

In general, however, it seems most appropriate to suppose that one may be able to infer from the control problem a – perhaps itself fuzzy – partitioning of the whole input space (or similarly of the output space). Then one will be in a position to split in a natural way the data set (1) or, correspondingly, the list (2) of control rules into different groups – and to consider the localized interpolation problems separately for these groups.

This obviously offers better chances for finding interpolating functions, particularly for getting solvable systems of fuzzy relation equations. However, one has to be aware that one should additionally take care that the different local interpolation functions fit together somehow smoothly – again an open problem that needs a separate discussion, and a problem that is more complicated for fuzzy interpolation than for the crisp counterpart because the fuzzy interpolating functions may realize the fuzzy interpolation nodes only approximately.

However, one may start from ideas like these to speculate about fuzzy versions of the standard *spline interpolation* methodology.

11.9 CRI as Approximation and Interpolation

In the context of fuzzy control, the object which has to be determined, some control function Φ , is described only roughly, i.e., given only by its behavior in some (fuzzy) points of the state space.

The standard way to roughly describe the control function is to give a list (2) of linguistic control rules connecting fuzzy subsets A_i of the input space \mathcal{X} with fuzzy subsets B_i of the output space \mathcal{Y} , indicating that one likes to have

$$\Phi^*(A_i) = B_i, \quad i = 1, \dots, n, \quad (13)$$

for a suitable ‘fuzzified’ version $\Phi^* : \text{IF}(\mathcal{X}) \rightarrow \text{IF}(\mathcal{Y})$ of the control function $\Phi : \mathcal{X} \rightarrow \mathcal{Y}$.

The additional approximation idea of the CRI is to approximate Φ^* by a fuzzy function $\Psi^* : \text{IF}(\mathcal{X}) \rightarrow \text{IF}(\mathcal{Y})$ determined for all $A \in \text{IF}(\mathcal{X})$ by

$$\Psi^*(A) = A \circ R, \quad (14)$$

which refers to some suitable fuzzy relation $R \in \text{IF}(\mathcal{X} \times \mathcal{Y})$ and understands \circ as sup-t composition.

Formally, thus, the equation (13) becomes transformed into some well-known system (5) of relation equations

$$A_i \circ R = B_i, \quad i = 1, \dots, n,$$

to be solved for the unknown fuzzy relation R .

This approximation idea fits well with the fact that one often is satisfied with pseudosolutions of (5), and particularly with the MA-pseudo-solution R_{MA} of Mamdani and Assilian, or the S-pseudo-solution \tilde{R} of Sanchez. Both of them determine approximations Ψ^* to the (fuzzified) control function Φ^* .

11.10 Approximate Solutions of Fuzzy Relation Equations

The author used in previous papers the notion of approximate solution only naively in the sense of a fuzzy relation which roughly describes the intended control behavior given via some list of linguistic control rules.

11.10.1 A Formal Definition

A precise definition of a notion of approximate solution was given by Wu [15]. In that approach an approximate solution \tilde{R} of a system (5) of fuzzy relation equations (FREs) is defined as a fuzzy relation satisfying

1. There are fuzzy sets A_i' and B_i' such that for all $i = 1, \dots, n$, one has

$$A_i \subseteq A_i' \text{ and } B_i \subseteq B_i' \text{ as well as } A_i' \circ \tilde{R} = B_i'.$$

2. If there exist fuzzy sets A_i^* and B_i^* for $i = 1, \dots, n$ and a fuzzy relation R^* such that for all $i = 1, \dots, n$,

$$A_i^* \circ R^* = B_i^* \text{ and } A_i \subseteq A_i^* \subseteq A_i', \quad B_i \subseteq B_i^* \subseteq B_i',$$

then one has $A_i^* = A_i'$ and $B_i^* = B_i'$ for all $i = 1, \dots, n$.

11.10.2 Generalizations of Wu's Approach

It is obvious that the two conditions (1) and (2) of Wu are independent.

What is, however, not obvious at all – and even rather arbitrary – is that condition (1) also says that the approximating input–output data (A'_i, B'_i) should approximate the original input data *from above* and the original output data *from below*.

Before we give a generalized definition we coin the name of an *approximating system* for (5) and understand by it any system

$$C_i \circ R = D_i, \quad i = 1, \dots, n \quad (15)$$

of relation equations with the same number of equations.

Definition 2. A *ul-approximate solution of a system (5) of relation equations* is a solution of a *ul-approximating system for (5)* which satisfies

$$A_i \subseteq C_i \quad \text{and} \quad B_i \supseteq D_i, \quad \text{for } i = 1, \dots, n. \quad (16)$$

An *lu-approximate solution of a system (5) of relation equations* is a solution of an *lu-approximating system for (5)* which satisfies

$$A_i \supseteq C_i \quad \text{and} \quad B_i \subseteq D_i, \quad \text{for } i = 1, \dots, n. \quad (17)$$

An *l*-approximate solution of a system (5) of relation equations* is a solution of an *l*-approximating system for (5)* which satisfies

$$A_i \supseteq C_i \quad \text{and} \quad B_i = D_i, \quad \text{for } i = 1, \dots, n. \quad (18)$$

In a similar way one defines the notions of *ll-approximate solution*, *uu-approximate solution*, *u*-approximate solution*, **l-approximate solution*, and **u-approximate solution*.

Corollary 18. (i) Each **l-approximate solution of (5)* is an *ul-approximate solution* and an *ll-approximate solution* of (5).

(ii) Each *u*-approximate solution of (5)* is also an *ul-approximate solution* and an *uu-approximate solution* of (5).

Proposition 19. For each system (5) of relation equations its *S-pseudo-solution* \widehat{R} is an **l-approximate solution*.

This generalizes a result of Klir and Yuan [16].

Proposition 20. For each system (5) of relation equations with normal input data, its *MA-pseudo-solution* R_{MA} is an **u-approximate solution*.

Together with Corollary 18 these two propositions say that each system of relation equations has approximate solutions of any one of the types introduced in this section.

However, it should be mentioned that these types of approximate solutions belong to a rather restricted class: caused by the fact that we considered, following Wu, only lower and upper approximations w.r.t. the inclusion relation; i.e., they are *inclusion based*. Other and more general approximations of the given input–output data systems are obviously possible. But we will not discuss further versions here.

11.10.3 Optimality of Approximate Solutions

All the previous results do not give any information about some kind of ‘quality’ of the approximate solutions or the approximating systems. This is to some extent related to the fact that up to now we disregarded in our modified terminology Wu’s condition (2), which is a kind of optimality condition.

Definition 3. An inclusion-based approximate solution \tilde{R} of a system (5) is called optimal iff there does not exist a solvable system $R''C_i' = D_i'$ of relation equations whose input–output data (C_i', D_i') approximate the original input–output data of (5) strongly better than the input–output data (C_i, D_i) of the system which determines the fuzzy relation \tilde{R} .

Proposition 21. If an inclusion-based *l-approximate solution \tilde{R} is optimal, then it is also optimal as a ul-approximate solution and as an ll-approximate solution.

Similar results hold true also for l*-, u*-, and *u-approximate solutions.

In those considerations we look for solutions of ‘approximating systems’ of FREs: of course, these solutions form some space of functions – and within this space one is interested to find ‘optimal members’ for the solution problem under consideration.

An obvious modification is to fix in some other way such a space \mathcal{R} of functions, i.e., independent of the idea of approximating systems of FREs.

In that moment one also has to specify some ranking for the members of that space \mathcal{R} of functions.

In the following we go on to discuss optimality results from both these points of view.

11.11 Some Optimality Results for Approximate Solutions

The problem now is whether the pseudosolutions \hat{R} and R_{MA} are optimal approximate solutions.

11.11.1 Optimality of the S-Pseudo-Solution

For the S-pseudo-solution \hat{R} as a ul-approximate solution this optimality was shown by Klir and Yuan [16].

Proposition 22. The fuzzy relation \hat{R} is always an \subseteq -optimal *l-approximate solution of (5).

From the second point of view we have, slightly reformulating and extending results presented in [4], the following result, given also in [17].

Theorem 23. Consider an unsolvable system of FREs. Then the S-pseudo-solution \hat{R} is the best approximate solution in the space \mathcal{R}_l :

$$\mathcal{R}_l = \{R \in \mathcal{R} \mid A_i \circ R \subseteq B_i \text{ for all } 1 \leq i \leq n\}.$$

under the ranking \leq_l :

$$R' \leq_l R'' \quad \text{iff} \quad A_i \circ R'' \subseteq A_i \circ R' \text{ for all } 1 \leq i \leq n.$$

Remark. Similarly one can prove that \hat{R} is the best approximate solution in the space \mathcal{R}_l under the ranking \leq_δ :

$$R' \leq_\delta R'' \quad \text{iff} \quad \delta(R'') \leq \delta(R')$$

for

$$\delta^*(R) = \bigwedge_{i=1}^n (B_i \equiv^* A_i \circ R) = \bigwedge_{i=1}^n \bigwedge_{y \in Y} (B_i(y) \leftrightarrow (A_i \circ R)(y)). \quad (19)$$

This index $\delta^*(R)$ is quite similar to the solvability degree $\delta(R)$ to be introduced later on in (23).

11.11.2 Optimality of the MA-Pseudo-Solution

For the MA-pseudo-solution the situation is different, as was indicated in [3].

Proposition 24. *There exist systems (5) of relation equations for which their MA-pseudo-solution R_{MA} is an $*u$ -approximate solution which is not optimal, i.e., an approximate solution in the approximation space \mathcal{R}_u :*

$$\mathcal{R}_u = \{R \in \mathcal{R} \mid A_i \circ R \supseteq B_i \text{ for all } 1 \leq i \leq n\},$$

but is not optimal in this set under the preorder \leq_u :

$$R' \leq_u R'' \quad \text{iff} \quad A_i \circ R' \leq A_i \circ R'' \text{ for all } 1 \leq i \leq n.$$

The crucial difference of the optimality result for \widehat{R} to the situation for R_{MA} is that in the former case the solvable approximating system has its own (largest) solution \widehat{S} . But a solvable approximating system may fail to have its MA-pseudo-solution R_{MA} as a solution.

The last remark leads us to a partial optimality result w.r.t. the MA-pseudo-solution. The proofs of the results which shall be mentioned now can be found in [4], or easily derived from the results given there.

Definition 4. *Let us call a system (5) of relation equations MA-solvable iff its MA-pseudo-solution R_{MA} is a solution of this system.*

Proposition 25. *If a system of FREs has an MA-solvable $*u$ -approximating system*

$$A_i \circ R = B_i^*, \quad i = 1, \dots, n, \quad (20)$$

such that for the MA-pseudo-solution R_{MA} of the original system of FREs one has

$$B_i \subseteq B_i^* \subseteq A_i \circ R_{MA}, \quad i = 1, \dots, n,$$

then one has

$$B_i^* = A_i \circ R_{MA} \quad \text{for all } i = 1, \dots, n.$$

Corollary 26. *If all input sets of (5) are normal then the system*

$$A_i \circ R = A_i \circ R_{MA}, \quad i = 1, \dots, n, \quad (21)$$

is the smallest MA-solvable $*u$ -supersystem for (5).

This leads back to the iterated pseudosolution strategies.

Corollary 27. *Let \widehat{R} be the S-pseudo-solution of (5), let $\widehat{B}_i = A_i \circ \widehat{R}$ for $i = 1, \dots, n$, and suppose that the modified system*

$$A_i \circ R = \widehat{B}_i, \quad i = 1, \dots, n, \quad (22)$$

is MA-solvable. Then this iterated pseudosolution $R_{MA}[A_k \circ \widehat{R}]$ is an optimal $*l$ -approximate solution of (5).

Furthermore it is a best approximate solution of the original system in the space \mathcal{R}_l under the ranking \leq_l .

Let us also mention the following result (cf. [17]).

Theorem 28. Consider an unsolvable system of FREs such that all input fuzzy sets A_i , $1 \leq i \leq n$, are normal and form a semipartition of \mathbf{X} . Then

$$R_{MA}(x, y) = \bigvee_{i=1}^n (A_i(x) * B_i(y))$$

is a best possible approximate solution in the space

$$\mathcal{R}_u = \{R \in \mathcal{R} \mid A_i \circ R \supseteq B_i \quad \text{for all } 1 \leq i \leq n\},$$

under the preorder \leq_u :

$$R' \leq_u R'' \text{ iff } A_i \circ R' \leq A_i \circ R'' \quad \text{for all } 1 \leq i \leq n.$$

These considerations can be further generalized.

Consider some *pseudosolution strategy* \mathbb{S} , i.e., some mapping from the class of families $(A_i, B_i)_{1 \leq i \leq n}$ of input–output data pairs into the class of fuzzy relations, which yields for any given system (5) of relation equations an \mathbb{S} -*pseudosolution* $R_{\mathbb{S}}$.

Then the system (5) will be called \mathbb{S} -solvable iff $R_{\mathbb{S}}$ is a solution of this system.

Definition 5. We shall say that the \mathbb{S} -pseudo-solution $R_{\mathbb{S}}$ depends isotonically (w.r.t. inclusion) on the output data of the system (5) of relation equations iff the condition

$$\text{if } B_i \subseteq B'_i \text{ for all } i = 1, \dots, n \text{ then } R_{\mathbb{S}} \subseteq R'_{\mathbb{S}}$$

holds true for the \mathbb{S} -pseudo-solutions $R_{\mathbb{S}}$ of the system (5) and $R'_{\mathbb{S}}$ of an ‘output-modified’ system $A_i \circ R = B'_i$, $i = 1, \dots, n$.

Definition 6. We understand by an \mathbb{S} -optimal $*u$ -approximate solution of the system (5) the \mathbb{S} -pseudo-solution of an \mathbb{S} -solvable $*u$ -approximating system of (5) which has the additional property that no strongly better $*u$ -approximating system of (5) is \mathbb{S} -solvable.

Proposition 29. Suppose that the \mathbb{S} -pseudo-solution depends isotonically on the output data of the systems of relation equations. Assume furthermore that for the \mathbb{S} -pseudo-solution $R_{\mathbb{S}}$ of (5) one always has $B_i \subseteq A_i \circ R_{\mathbb{S}}$ (or always has $A_i \circ R_{\mathbb{S}} \subseteq B_i$) for $i = 1, \dots, n$. Then the \mathbb{S} -pseudo-solution $R_{\mathbb{S}}$ of (5) is an \mathbb{S} -optimal $*u$ -approximate (or: $*l$ -approximate) solution of system (5).

It is clear that Corollary 26 is the particular case of the MA-pseudo-solution strategy. But also Proposition 22 is a particular case of this Proposition 29: the case of the S-pseudo-solution strategy (having in mind that S-solvability and solvability are equivalent notions).

11.12 Introducing the Solvability Degree

Following [1, 11] one may consider for a system of relation equations the (global) *solvability degree*

$$\xi = \left[\left[\exists X \prod_{i=1}^n (A_i \circ X \equiv B_i) \right] \right], \quad (23)$$

and for any fuzzy relation R their *solution degree*

$$\delta(R) = \left[\left[\prod_{i=1}^n (A_i \circ R \equiv B_i) \right] \right]. \quad (24)$$

Here \prod means the finite iteration of the strong conjunction connective $\&$, and is defined in the standard way.

The following result was first proved in [10], and has been further discussed in [1, 11].

Theorem 30. $\xi^n \leq \delta(\widehat{R}) \leq \xi$.

Of course, the n th power here is again the iteration of the strong conjunction operation $*$, i.e., the semantical counterpart of the syntactic operation \prod .

Obviously this result can be rewritten in a slightly modified form which makes it more transparent that Theorem 30 really gives an estimation for the solvability degree ξ in terms of a particular solution degree.

Corollary 31. $\delta(\widehat{R})^n \leq \xi^n \leq \delta(\widehat{R})$.

One has for continuous t -norms that they are ordinal sums of isomorphic copies of two basic t -norms, of the Lukasiewicz t -norm t_L given by $t_L(u, v) = \max\{u + v - 1, 0\}$ and of the arithmetic product t_P . (Sometimes Gödel's t -norm min is also allowed for these summands. However, this is unimportant because of the definition of an ordinal sum of t -norms.)

Corollary 32. In the case that $*$ is a continuous t -norm t , the values $\delta(\widehat{R})$ and ξ always belong to the same ordinal t -summand.

A further property is of interest for the case of t -norm-based structures \mathcal{L} .

Proposition 33. For each continuous t -norm t and each $1 \leq n \in \mathbb{N}$ there exists n th roots.

Having this in mind, one can immediately rewrite Theorem 30 for this particular case in an even nicer form as we did it in Corollary 31.

Proposition 34. For t -norms which have n th roots one has the inequalities

$$\delta(\widehat{R}) \leq \xi \leq \sqrt[n]{\delta(\widehat{R})}.$$

Using as in [18] the notation $z(u)$ for the largest t -idempotent below u , this last result allows for the following slight modification.

Corollary 35. For t -norms which have n th roots one has the inequalities

$$z(\delta(\widehat{R})) \leq \xi \leq \sqrt[n]{\delta(\widehat{R})}.$$

Besides these core results which involve the solution degree $\delta(\widehat{R})$ of the S-pseudo-solution of the system (5), the problem appears to determine the solution degree of the relation R_{MA} .

Proposition 36. If all input sets A_i are normal then

$$\delta^*(R_{MA}) = \left[\left[\prod_i \bigwedge_j (A_i \cap_t A_j \neq \emptyset \rightarrow B_i \subseteq B_j) \right] \right].$$

This is a generalization of the former Klawonn criterion.

We also find, as explained in [3], a second result which indicates that $R_{\text{MA}}[A_k \circ \widehat{R}[B_k]]$ is at least sometimes as good a pseudosolution as R_{MA} .

Proposition 37. *If all input sets A_i are normal and if one has*

$$\models B_i \subseteq B_j \rightarrow A_i \circ \widehat{R} \subseteq A_j \circ \widehat{R},$$

then

$$\delta^*(R_{\text{MA}}) \leq \delta^*(R_{\text{MA}}[A_k \circ \widehat{R}[B_k]]).$$

11.13 Interpolation Strategies and Aggregation Operators

There is the well-known distinction between FATI and FITA strategies to evaluate systems of linguistic control rules w.r.t. arbitrary fuzzy inputs from $\mathcal{F}(X)$.

The core idea of a FITA strategy is that it is a strategy which first infers (by reference to the single rules) and then aggregates starting from the actual input information A . Contrary to that, a FATI strategy is a strategy which first aggregates (the information in all the rules into one fuzzy relation) and then infers starting from the actual input information A .

Both these strategies use the set-theoretic union as their aggregation operator. Furthermore, both of them refer to the CRI as their core tool of inference.

In general, however, the interpolation operators we intend to consider depend more generally on some inference operator(s) as well as on some aggregation operator.

By an *inference operator* we mean here simply a mapping from the fuzzy subsets of the input space to the fuzzy subsets of the output space.¹

And an *aggregation operator* \mathbf{A} , as explained e.g. in [19, 20], is a family $(f^n)_{n \in \mathbb{N}}$ of ('aggregation') operations, each f^n an n -ary one, over some partially ordered set \mathbf{M} , with ordering \leq , with a bottom element $\mathbf{0}$ and a top element $\mathbf{1}$, such that each operation f^n is non-decreasing, maps the bottom to the bottom: $f^n(\mathbf{0}, \dots, \mathbf{0}) = \mathbf{0}$, and the top to the top: $f^n(\mathbf{1}, \dots, \mathbf{1}) = \mathbf{1}$.

Such an aggregation operator $\mathbf{A} = (f^n)_{n \in \mathbb{N}}$ is a *commutative* one iff each operation f^n is commutative. And \mathbf{A} is an *associative* aggregation operator iff, e.g., for $n = k + l$ one always has $f^n(a_1, \dots, a_n) = f^2(f^k(a_1, \dots, a_k), f^l(a_{k+1}, \dots, a_n))$ and in general

$$f^n(a_1, \dots, a_n) = f^r(f^{k_1}(a_1, \dots, a_{k_1}), \dots, f^{k_r}(a_{m+1}, \dots, a_n))$$

for $n = \sum_{i=1}^r k_i$ and $m = \sum_{i=1}^{r-1} k_i$.

Our aggregation operators further on are supposed to be commutative as well as associative ones.²

Observe that an associative aggregation operator $\mathbf{A} = (f^n)_{n \in \mathbb{N}}$ is essentially determined by its binary aggregation function f^2 , more precisely, by its subfamily $(f^n)_{n \leq 2}$.

Additionally we call an aggregation operator $\mathbf{A} = (f^n)_{n \in \mathbb{N}}$

<i>additive</i>	iff always $b \leq f^2(b, c)$,
<i>multiplicative</i>	iff always $f^2(b, c) \leq b$,
<i>idempotent</i>	iff always $b = f^2(b, b)$.

¹ This terminology has its historical roots in the fuzzy control community. There is no relationship at all with the logical notion of inference intended and supposed here; but—of course—also not ruled out.

² It seems that this is a rather restrictive choice from a theoretical point of view. However, in all the usual cases these restrictions are satisfied.

Corollary 38. Let $\mathbf{A} = (f^n)_{n \in \mathbb{N}}$ be an aggregation operator.

- (i) If \mathbf{A} is idempotent, then one has always $f^2(\mathbf{0}, b) \leq f^2(b, b) = b$;
- (ii) If \mathbf{A} is additive, then one has always $b \leq f^2(\mathbf{0}, b)$;
- (iii) If \mathbf{A} is multiplicative, then one has always $f^2(\mathbf{0}, b) = \mathbf{0}$.

As in [21], we now consider interpolation operators Φ of FITA type and interpolation operators Ψ of FATI type, which have the abstract forms

$$\Psi_D(A) = \mathbf{A}(\theta_1(A), \dots, \theta_n(A)), \quad (25)$$

$$\Xi_D(A) = \widehat{\mathbf{A}}(\theta_1, \dots, \theta_n)(A). \quad (26)$$

Here we assume that each one of the ‘local’ inference operators θ_i is determined by the single input–output pair $\langle A_i, B_i \rangle$. Therefore we shall prefer to write $\theta_{\langle A_i, B_i \rangle}$ instead of θ_i only because this extended notation makes the reference to (or even dependence from) the input–output data more transparent. And we have to assume that the aggregation operator \mathbf{A} operates on fuzzy sets and that the aggregation operator $\widehat{\mathbf{A}}$ operates on inference operators.

With this extended notation the formulas (25) and (26) become

$$\Psi_D(A) = \mathbf{A}(\theta_{\langle A_1, B_1 \rangle}(A), \dots, \theta_{\langle A_n, B_n \rangle}(A)), \quad (27)$$

$$\Xi_D(A) = \widehat{\mathbf{A}}(\theta_{\langle A_1, B_1 \rangle}, \dots, \theta_{\langle A_n, B_n \rangle})(A). \quad (28)$$

11.14 Some Particular Examples

Some particular cases of these interpolation procedures have been discussed in [22]. These authors consider four different cases. First they look at the FITA-type interpolation

$$\Psi_D^1(A) = \bigcap_i (A \circ (A_i \triangleright B_i)), \quad (29)$$

using as in [11] the notation $A_i \triangleright B_i$ to denote the fuzzy relation with membership function

$$(A_i \triangleright B_i)(x, y) = A_i(x) \rightarrow B_i(y).$$

Obviously this is just (a slight modification of) the fuzzy control Strategy of Holmblad/Ostergaard [23].

Their second example discusses a FATI-type approach given by

$$\Xi_D^2(A) = A \circ \bigcap_i ((A_i \triangleright B_i)), \quad (30)$$

and is thus just the common CRI-based strategy of the S-pseudo-solution, used in this general form already in [10] (cf. also [11]).

Their third example is again of FITA type and determined by

$$\Psi_D^3(A) = \bigcap_i \{y \mid \delta(A, A_i) \rightarrow B_i(y)\}, \quad (31)$$

using besides the previously mentioned class term notation for fuzzy sets the activation degree

$$\delta(A, A_i) = \bigwedge_{x \in X} (A(x) \rightarrow A_i(x)), \quad (32)$$

which is a degree of subsethood of the actual input fuzzy set A w.r.t. the i th rule input A_i .

And the fourth one is a modification of the third one, determined by

$$\Psi_D^4(A) = \bigcap_{\emptyset \neq J \subseteq N} \{y \mid \delta(A, \bigcup_{j \in J} A_j) \rightarrow \bigcup_{j \in J} B_i(y)\}, \quad (33)$$

using $N = \{1, 2, \dots, n\}$.

In these examples the main aggregation operators are the set-theoretic union and the set-theoretic intersection. Both are obviously associative, commutative, and idempotent. Additionally the union is an additive and the intersection a multiplicative aggregation operator.

11.15 Stability Conditions for the Given Data

If Θ_D is a fuzzy inference operator of one of the types (27) and (28), then the interpolation property one likes to have realized is that one has

$$\Theta_D(A_i) = B_i \quad (34)$$

for all the data pairs (A_i, B_i) . In the particular case that the operator Θ_D is given by (4), this is just the problem to solve the system (34) of fuzzy relation equations.

Definition 7. *In the present generalized context let us call the property (34) the D -stability of the fuzzy inference operator Θ_D .*

To find D -stability conditions on this abstract level seems to be rather difficult in general. However, the restriction to fuzzy inference operators of FITA type makes things easier.

It is necessary to have a closer look at the aggregation operator $\mathbf{A} = (f^n)_{n \in \mathbb{N}}$ involved in (25) which operates on $\mathcal{F}(\mathbf{Y})$, of course with inclusion as partial ordering.

Definition 8. *Having $B, C \in \mathcal{F}(\mathbf{Y})$ we say that C is \mathbf{A} -negligible w.r.t. B iff $f^2(B, C) = f^1(B)$ holds true.*

The core idea here is that in any aggregation by \mathbf{A} the presence of the fuzzy set B among the aggregated fuzzy sets makes any presence of C superfluous.

Example..

1. C is \bigcup -negligible w.r.t. B iff $C \subseteq B$; and this holds similarly true for all idempotent and additive aggregation operators.
2. C is \bigcap -negligible w.r.t. B iff $C \supseteq B$; and this holds similarly true for all idempotent and multiplicative aggregation operators.
3. The bottom element $C = \mathbf{0}$ in the domain of an additive and idempotent aggregation operator \mathbf{A} is \mathbf{A} -negligible w.r.t. any other element of that domain.

Proposition 39. *Consider a fuzzy inference operator of FITA type*

$$\Psi_D = \mathbf{A}(\theta_{(A_1, B_1)}, \dots, \theta_{(A_n, B_n)}).$$

It is sufficient for the D -stability of Ψ_D , i.e., to have

$$\Psi_D(A_k) = B_k \quad \text{for all } k = 1, \dots, n$$

that one always has

$$\theta_{(A_k, B_k)}(A_k) = B_k$$

and additionally that for each $i \neq k$, the fuzzy set

$$\theta_{(A_k, B_k)}(A_i) \text{ is } \mathbf{A}\text{-negligible w.r.t. } \theta_{(A_k, B_k)}(A_k).$$

The proof follows immediately from the corresponding definitions. And this result has two quite interesting specializations which themselves generalize well-known results about fuzzy relation equations.

Corollary 40. *It is sufficient for the \mathcal{D} -stability of a fuzzy inference operator Ψ_D of FITA type that one has*

$$\Psi_D(A_i) = B_i \quad \text{for all } 1 \leq i \leq n$$

and that always $\theta_{(A_i, B_i)}(A_j)$ is \mathbf{A} -negligible w.r.t. $\theta_{(A_i, B_i)}(A_i)$.

Corollary 41. *It is sufficient for the \mathcal{D} -stability of a fuzzy inference operator Ψ_D of FITA type, which is based on an additive and idempotent aggregation operator, that one has*

$$\Psi_D(A_i) = B_i \quad \text{for all } 1 \leq i \leq n$$

and that always $\theta_{(A_i, B_i)}(A_j)$ is the bottom element in the domain of the aggregation operator \mathbf{A} .

Obviously this is a direct generalization of the fact that systems of fuzzy relation equations are solvable if their input data form a pairwise disjoint family (w.r.t. the corresponding t -norm-based intersection) because in this case one usually has

$$\begin{aligned} \theta_{(A_i, B_i)}(A_j) &= A_j \circ (A_i \times B_i) = \{y \mid \exists x (x \in A_j \& (x, y) \in A_i \times B_i)\} \\ &= \{y \mid \exists x (x \in A_j \cap_+ A_i \& y \in B_i)\}. \end{aligned}$$

To extend these considerations from inference operators (25) of the FITA type to those ones of the FATI type (26) let us consider the following notion.

Definition 9. *Suppose that $\widehat{\mathbf{A}}$ is an aggregation operator for inference operators and that \mathbf{A} is an aggregation operator for fuzzy sets. Then $(\widehat{\mathbf{A}}, \mathbf{A})$ is an application distributive pair of aggregation operators iff*

$$\widehat{\mathbf{A}}(\theta_1, \dots, \theta_n)(X) = \mathbf{A}(\theta_1(X), \dots, \theta_n(X)) \tag{35}$$

holds true for arbitrary inference operators $\theta_1, \dots, \theta_n$ and fuzzy sets X .

Using this notion it is easy to see that one has on the left-hand side of (35) a FATI-type inference operator and on the right-hand side an associated FITA-type inference operator. So one is able to give a reduction of the FATI case to the FITA case.

Proposition 42. *Suppose that $(\widehat{\mathbf{A}}, \mathbf{A})$ is an application distributive pair of aggregation operators. Then a fuzzy inference operator Ξ_D of FATI type is \mathcal{D} -stable iff its associated fuzzy inference operator Ψ_D of FITA type is \mathcal{D} -stable.*

11.16 Stability Conditions for Modified Data

The combined approximation and interpolation problem, as previously explained, sheds new light on the standard approaches toward fuzzy control via CRI-representable functions originating from the works of Mamdani and Assilian [12] and Sanchez [9] particularly for the case that neither the Mamdani–Assilian

relation R_{MA} , determined by the membership degrees,

$$R_{MA}(x, y) = \bigvee_{i=1}^n A_i(x) * B_i(y), \quad (36)$$

nor the Sanchez relation \widehat{R} , determined by the membership degrees,

$$\widehat{R}(x, y) = \bigwedge_{i=1}^n (A_i(x) \rightarrow B_i(y)), \quad (37)$$

offers a solution for the system of fuzzy relation equations. In any case both these fuzzy relations determine CRI-representable fuzzy functions which provide approximate solutions for the interpolation problem.

In other words, the consideration of CRI-representable functions determined by (36) as well as by (37) provides two methods for an approximate solution of the main interpolation problem. As is well known and explained, e.g., in [11], the approximating interpolation function CRI-represented by \widehat{R} always gives a lower approximation and that one CRI-represented by R_{MA} gives an upper approximation for normal input data.

Extending these results, in [3] the iterative combination of these methods has been discussed to get better approximation results. For the iterations there, always the next iteration step consisted in an application of a predetermined one of the two approximation methods to the data family with the original input data and the real, approximating output data which resulted from the application of the former approximation method.

A similar iteration idea was also discussed in [22], however, restricted always to the iteration of only one of the approximation methods explained in (29), (30), (31), and (33).

Therefore let us now, in the general context of this chapter, discuss the problem of \mathcal{D} -stability for a modified operator $\Theta_{\mathcal{D}}^*$, which is determined by the kind of iteration of $\Theta_{\mathcal{D}}$ just explained.

Let us consider the $\Theta_{\mathcal{D}}$ -modified data set \mathcal{D}^* given as

$$\mathcal{D}^* = (\langle A_i, \Theta_{\mathcal{D}}(A_i) \rangle)_{1 \leq i \leq n}, \quad (38)$$

and define from it the modified fuzzy inference operator $\Theta_{\mathcal{D}}^*$ as

$$\Theta_{\mathcal{D}}^* = \Theta_{\mathcal{D}^*}. \quad (39)$$

For these modifications, the problem of stability reappears. Of course, the new situation here is only a particular case of the former. And it becomes a simpler one in the sense that the stability criteria now refer only to the input data A_i of the data set $\mathcal{D} = (\langle A_i, B_i \rangle)_{1 \leq i \leq n}$.

Proposition 43. *It is sufficient for the \mathcal{D}^* -stability of a fuzzy inference operator $\Psi_{\mathcal{D}}^*$ of FITA type that one has*

$$\Psi_{\mathcal{D}}^*(A_i) = \Psi_{\mathcal{D}^*}(A_i) = \Psi_{\mathcal{D}}(A_i) \quad \text{for all } 1 \leq i \leq n \quad (40)$$

and that always $\theta_{\langle A_i, \Psi_{\mathcal{D}}(A_i) \rangle}(A_j)$ is **A-negligible** w.r.t. $\theta_{\langle A_i, \Psi_{\mathcal{D}}(A_i) \rangle}(A_i)$.

Let us look separately at the condition (40) and at the negligibility conditions.

Corollary 44. *The condition (40) is always satisfied if the inference operator $\Psi_{\mathcal{D}}^*$ is determined by the standard output-modified system of relation equations $A_i \circ R[A_k \circ R] = B_i$ in the notation of [3].*

Corollary 45. *In the case that the aggregation operator is the set-theoretic union, i.e., $A = \bigcup$, the condition (40) together with the inclusion relationships*

$$\theta_{\langle A_i, \Psi_{\mathcal{D}}(A_i) \rangle}(A_j) \subseteq \theta_{\langle A_i, \Psi_{\mathcal{D}}(A_i) \rangle}(A_i)$$

is sufficient for the \mathcal{D}^* -stability of a fuzzy inference operator $\Psi_{\mathcal{D}}^*$.

As in Section 11.15 one is able to transfer this result to FATI-type fuzzy inference operators.

Corollary 46. Suppose that $(\widehat{\mathbf{A}}, \mathbf{A})$ is an application distributive pair of aggregation operators. Then a fuzzy inference operator Φ_D^* of FATI type is \mathcal{D}^* -stable iff its associated fuzzy inference operator Ψ_D^* of FITA type is \mathcal{D}^* -stable.

11.17 Application Distributivity

Based on the notion of application distributive pair of aggregation operators, the property of \mathcal{D} -stability can be transferred back and forth between two inference operators of FATI type and FITA type if they are based on a pair of application distributive aggregation operators.

What has not been discussed previously was the existence and the uniqueness of such pairs. Here are some results concerning these problems.

The uniqueness problem has a simple solution.

Proposition 47. If $(\widehat{\mathbf{A}}, \mathbf{A})$ is an application distributive pair of aggregation operators then $\widehat{\mathbf{A}}$ is uniquely determined by \mathbf{A} , and conversely also \mathbf{A} is uniquely determined by $\widehat{\mathbf{A}}$.

Proof. Let \mathbf{A} be given. Then condition (35), being valid for all fuzzy sets X , determines for all fuzzy inference operators $\theta_1, \dots, \theta_n$ uniquely the functions $\widehat{\mathbf{A}}(\theta_1, \dots, \theta_n)$. And therefore (35) also determines the aggregation operator $\widehat{\mathbf{A}}$ uniquely. ■

The converse statement follows in a similar way. ■

And for the existence problem we have a nice reduction to the two-argument case.

Proposition 48. Suppose that \mathbf{A} is a commutative and associative aggregation operator and G some operation for fuzzy inference operators satisfying

$$\mathbf{A}(\theta_1(X), \theta_2(X)) = G(\theta_1, \theta_2)(X) \quad (41)$$

for all fuzzy sets X . Then G can be extended to an aggregation operator $\widehat{\mathbf{G}}$ which is commutative and associative and forms with \mathbf{A} an application distributive pair $(\widehat{\mathbf{G}}, \mathbf{A})$ of aggregation operators.

Proof. The commutativity of \mathbf{A} yields

$$G(\theta_1, \theta_2)(X) = G(\theta_2, \theta_1)(X)$$

for all fuzzy sets X and hence $G(\theta_1, \theta_2) = G(\theta_2, \theta_1)$, i.e., the commutativity of G as an operation for fuzzy inference operators.

In a similar way the associativity of \mathbf{A} implies the associativity of G .

Hence it is a routine matter to expand the binary operator G to an n -ary one G^n for each $n \geq 2$. Thus one has a commutative and associative aggregation operator $\widehat{\mathbf{G}} = (G^n)_{n \in \mathbb{N}}$ for fuzzy inference operators if one additionally puts $G^1 = \text{id}$.

Finally the application distributivity the pair $(\widehat{\mathbf{G}}, \mathbf{A})$ is again easily derived from (41) and the definition of $\widehat{\mathbf{G}}$. ■

It is easy to recognize that this result can be reversed.

Corollary 49. Suppose that \mathbf{A} is a commutative and associative aggregation operator and $(\widehat{\mathbf{A}}, \mathbf{A})$ is an application distributive pair of aggregation operators, and let $\widehat{\mathbf{A}} = (\widehat{f}^n)_{n \in \mathbb{N}}$. Then $G = \widehat{f}^2$ satisfies condition (41) for all fuzzy sets X .

Both results together give us the following reduction of the full-application distributivity condition.

Theorem 50. Suppose that \mathbf{A} is a commutative and associative aggregation operator. For the case that there exists an aggregation operator $\widehat{\mathbf{A}}$ such that $(\widehat{\mathbf{A}}, \mathbf{A})$ forms an application distributive pair of aggregation operators it is necessary and sufficient that there exists some operation G for fuzzy inference operators satisfying

$$\mathbf{A}(\theta_1(X), \theta_2(X)) = G(\theta_1, \theta_2)(X) \quad (42)$$

for all fuzzy inference operators θ_1 and θ_2 and all fuzzy sets X .

The proof is obvious from these last two results.

For the particular, and very popular, cases that one has $\mathbf{A} = \bigcup$ or $\mathbf{A} = \bigcap$, and that the application of a fuzzy inference operator θ to a fuzzy set X means the CRI application of a fuzzy relation to a fuzzy set, one immediately sees that one may choose $\widehat{\mathbf{G}} = \bigcup$ or $\widehat{\mathbf{G}} = \bigcap$, respectively.

11.18 Invoking a Defuzzification Strategy

In a lot of practical applications of the fuzzy control strategies which form the starting point for the previous general considerations, the fuzzy model – e.g., determined by a list of linguistic IF–THEN rules – is realized in the context of a further defuzzification strategy, which is nothing but a mapping $F : \mathcal{F}(\mathbb{Y}) \rightarrow \mathbb{Y}$ for fuzzy subsets of the output space \mathbb{Y} .

Having this in mind, it seems reasonable to consider the following modification of the \mathcal{D} -stability condition, which is a formalization of the idea to have ‘stability modulo defuzzification.’

Definition 10. A fuzzy inference operator $\Theta_{\mathcal{D}}$ is (F, \mathcal{D}) -stable w.r.t. a fuzzification method $F : \mathcal{F}(\mathbb{Y}) \rightarrow \mathbb{Y}$ iff one has

$$F(\Theta_{\mathcal{D}}(A_i)) = F(B_i) \quad (43)$$

for all the data pairs $\langle A_i, B_i \rangle$ from \mathcal{D} .

For the fuzzy modeling process which is manifested in the data set \mathcal{D} this condition (43) is supposed to fit well with the control behavior one is interested to implement. If for some application this condition (43) seems to be unreasonable, this indicates that either the data set \mathcal{D} or the chosen defuzzification method F is unsuitable.

As a first, and rather restricted stability result for this modified situation, the following proposition shall be mentioned.

Proposition 51. Suppose that $\Theta_{\mathcal{D}}$ is a fuzzy inference operator of FITA type, i.e., of the form (25), that the aggregation is union $\mathbf{A} = \bigcup$ as, e.g., in the fuzzy inference operator for the Mamdani–Assilian case, and that the defuzzification strategy F is the ‘mean of max’ method. Then it is sufficient for the (F, \mathcal{D}) -stability of $\Theta_{\mathcal{D}}$ to have satisfied

$$\text{hgt}\left(\bigcup_{j=1, j \neq k}^n \theta_k(A_j)\right) < \text{hgt}(\theta_k(A_k)) \quad (44)$$

for all $k = 1, \dots, n$.

The proof follows from the corresponding definitions by straightforward routine calculations, and hgt means the ‘height’ of a fuzzy set, i.e., the supremum of its membership degrees.

11.19 Conclusion

Essentially the first appearance of the idea of information granulation has been the idea of linguistic values of some variables, their use for the rough description of functional dependencies using ‘linguistic’ rules, and the application of this idea to fuzzy control.

The most suitable mathematical context for fuzzy control problems determined by systems of linguistic control rules is to understand them as interpolation problems: a function from fuzzy subsets of an input space \mathbf{X} to fuzzy subsets of an output space \mathbf{Y} has to be determined from a (usually finite) list of information granules, i.e., in this functional setting of argument–value pairs.

With suitably restricted classes of interpolating functions, however, this global interpolation problem may become unsolvable. Then one is interested in approximate solutions of acceptable quality. We discuss a series of optimal approximation results for classes of approximating functions which naturally arise out of the natural transformation of the interpolation problem into the problem of solving systems of fuzzy relational equations.

But one may also consider some modifications of the original input–output data. For one such approach we also discuss sufficient conditions for the solvability of the modified interpolation problem.

Additionally the whole approaches may be put into a more general context. What has been considered here is a context which focuses on different combinations of aggregation and inference operators. Interestingly mainly the properties of the aggregation operators proved to be of importance for these considerations. So it actually remains an open problem whether the inference operations are really of minor importance, or whether our discussion simply missed some aspects for which the properties of the inference operations become crucial.

For completeness it shall be mentioned that only other generalizations are possible and may become important too. One such more algebraically oriented generalization was quite recently offered in [24].

References

- [1] S. Gottwald. Generalised solvability behaviour for systems of fuzzy equations. In: V. Novák and I. Perfilieva (eds), *Discovering the World with Fuzzy Logic, Advances in Soft Computing*. Physica-Verlag, Heidelberg, 2000, pp. 401–430.
- [2] S. Gottwald. Mathematical fuzzy control. A survey of some recent results. *Log. J. IGPL* 13 (5) (2005) 525–541.
- [3] S. Gottwald, V. Novák, and I. Perfilieva. Fuzzy control and t -norm-based fuzzy logic. Some recent results. In: *Proceedings of the 9th International Conference of IPMU'2002, ESIA – Université de Savoie*, Annecy, 2002, pp. 1087–1094.
- [4] I. Perfilieva and S. Gottwald. Fuzzy function as a solution to a system of fuzzy relation equations. *Int. J. Gen. Syst.* 32 (2003) 361–372.
- [5] S. Gottwald. *A Treatise on Many-Valued Logics. Studies in Logic and Computation*, Vol. 9. Research Studies Press, Baldock, 2001.
- [6] L.A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Syst. Man Cybern. SMC-3* (1973) 28–44.
- [7] R.D. Luce. A note on Boolean matrix theory. *Proc. Am. Math. Soc.* 3 (1952) 382–388.
- [8] A. DiNola, S. Sessa, W. Pedrycz, and E. Sanchez. *Fuzzy Relation Equations and Their Applications to Knowledge Engineering. Theory and Decision Library, Series D*. Kluwer, Dordrecht, 1989.
- [9] E. Sanchez. Resolution of composite fuzzy relation equations. *Inf. Control* 30 (1976) 38–48.
- [10] S. Gottwald. Characterizations of the solvability of fuzzy equations. *Elektron. Inf. Kybern.* 22 (1986) 67–91.
- [11] S. Gottwald. *Fuzzy Sets and Fuzzy Logic. The Foundations of Application – From a Mathematical Point of View*. Vieweg: Braunschweig/Wiesbaden and Teknea, Toulouse, 1993.
- [12] A. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Mach. Stud.* 7 (1975) 1–13.
- [13] F. Klawonn. Fuzzy points, fuzzy relations and fuzzy functions. In: V. Novák and I. Perfilieva (eds), *Discovering the World with Fuzzy Logic, Advances in Soft Computing*. Physica-Verlag, Heidelberg, 2000, pp. 431–453.
- [14] S. Gottwald. Criteria for non-interactivity of fuzzy logic controller rules. In: A. Straszak(ed), *Large Scale Systems: Theory and Applications, Proceedings of the 3rd IFAC/IFORS Symposium Warsaw 1983*. Pergamon Press, Oxford, 1984, pp. 229–233.
- [15] W. Wu. Fuzzy reasoning and fuzzy relation equations. *Fuzzy Sets Syst.* 20 (1986) 67–78.

- [16] G. Klir and B. Yuan. Approximate solutions of systems of fuzzy relation equations. In: *FUZZ-IEEE '94. Proceedings of the 3rd International Conference on Fuzzy Systems, Orlando, FL*, 1994, pp. 1452–1457.
- [17] I. Perfilieva. Fuzzy function as an approximate solution to a system of fuzzy relation equations. *Fuzzy Sets Syst.* 147 (2004) 363–383.
- [18] I. Perfilieva and A. Tonis. Compatibility of systems of fuzzy relation equations. *Int. J. Gen. Syst.* 29 (2000) 511–528.
- [19] T. Calvo, G. Mayor, and R. Mesiar (eds). *Aggregation Operators: New Trends and Applications*. Physica-Verlag, Heidelberg, 2002.
- [20] D. Dubois and H. Prade. On the use of aggregation operations in information fusion processes. *Fuzzy Sets Syst.* 142 (2004) 143–161.
- [21] S. Gottwald. On a generalization of fuzzy relation equations. In: *Proceedings of the 11th International Conference of IPMU 2006*, Edition EDK, Paris, 2006, Vol. 2, pp. 2572–2577.
- [22] N.N. Morsi and A.A. Fahmy. On generalized modus ponens with multiple rules and a residuated implication. *Fuzzy Sets Syst.* 129 (2002) 267–274.
- [23] L.P. Holmblad and J.J. Ostergaard. Control of a cement kiln by fuzzy logic. In: M.M. Gupta and E. Sanchez (eds), *Fuzzy Information and Decision Processes*. North-Holland, Amsterdam, 1982, pp. 389–399.
- [24] A. DiNola, A. Lettieri, I. Perfilieva, and V. Novák. Algebraic analysis of fuzzy systems. *Fuzzy Sets Syst.* 158 (2007) 1–22.

12

Fuzzy Numbers and Fuzzy Arithmetic

Luciano Stefanini, Laerte Sorini, and Maria Letizia Guerra

12.1 Introduction

The scientific literature on fuzzy numbers and arithmetic calculations is rich in several approaches to define fuzzy operations having many desired properties that are not always present in the implementations of classical extension principle or its approximations (shape preservation, reduction of the overestimation effect, requisite constraints, distributivity of multiplication and division, etc.).

What is well known to all practitioners is that appropriate use of fuzzy numbers in applications requires at least two features to be satisfied:

1. An easy way to represent and model fuzzy information with a sufficient or possibly high flexibility of shapes, without being constrained to strong simplifications, e.g., allowing asymmetries or nonlinearities;
2. A relative simplicity and computational efficiency to perform exact fuzzy calculations or to obtain good or error-controlled approximations of the results.

The two requirements above, if not solved, are often a bottleneck in the utilization of fuzzy information and a lot of work and scientific literature has been spent in those directions. On the other hand, as we will see, the fuzzy calculations are not immediate to be performed and in many cases they require to solve mathematically or computationally hard subproblems (e.g., global optimization, set-valued analysis, interval-based arithmetic, functional inverse calculation, and integration) for which a closed form is not available.

Fuzzy sets (and numbers) are complementary to probability and statistics in modeling uncertainty, imprecision, and vagueness of data and information (see the presentation in [1]); together with the methodologies of interval analysis and rough sets, they are the basic elements of granular computing (GrC) and serve *as a basis for the methodology of computing with words* ([2]). In particular, fuzziness is essentially related to imprecision (or partial truth) and uncertainty in the boundaries of sets and numbers, while granularity (and the granulation techniques) defines the scale or the detail level at which the domain of the interested variable or object values are described and coded. Fuzzy granulation of information and data (see [3–5]) produces fuzzy sets and numbers for the represented granules; fuzzy logic and calculus are the basic mathematical concepts and tools to formalize fuzzy variable functions and relations.

The arithmetical and topological structures of fuzzy numbers have been developed in the 1980s and this enabled to design the elements of fuzzy calculus (see [6, 7]); Dubois and Prade stated the exact analytical fuzzy mathematics and introduced the well-known LR model and the corresponding formulas for the fuzzy operations. For the basic concepts see, e.g., [8–12]. More recently, the literature on fuzzy numbers has grown in terms of contributions to fuzzy arithmetic operations and to the use of simple formulas to approximate them; an extensive recent survey and bibliography on fuzzy intervals is in [13].

Zadeh's extension principle (with some generalizations) plays a very important role in fuzzy set theory as it is a *quite natural and reasonable principle to extend the operators and the mapping from classical set theory, as well as its structures and properties, into the operators and the mappings in fuzzy set theory* ([14, 15]).

In general, the arithmetic operations on fuzzy numbers can be approached either by the direct use of the membership function (by Zadeh's extension principle) or by the equivalent use of the α -cuts representation.

The arithmetic operations and more general fuzzy calculations are natural when dealing with fuzzy reasoning and systems, where variables and information are described by fuzzy numbers and sets; in particular, procedures and algorithms have to take into account the existing dependencies (and constraints) relating all the operands involved and their meaning. The essential uncertainties are generally modeled in the preliminary definitions of the variables, but it is very important to pay great attention to how they propagate during the calculations. A solid result in fuzzy theory and practice is that calculations cannot be performed by using the same rules as in arithmetic with real numbers and in fact fuzzy calculus will not always satisfy the same properties (e.g., distributivity, invertibility, and others). If not performed by taking into account existing dependencies between the data, fuzzy calculations will produce excessive propagation of initial uncertainties (see [16–19]).

As we will see, the application of Zadeh's extension principle to the calculation of fuzzy expressions requires to solve simultaneously global (constrained) minimization and maximization problems and they have typically a combinatorial structure; the task is not easy, except for particular cases. For this reason, general algorithms have been proposed (the vertex method and its variants) but also specific methods based on the exploitation of the problem at hand to produce exact solutions or generate approximated subproblems to be solved more efficiently than the original ones.

By the α -cuts approach, it is possible to define a parametric representation of fuzzy numbers that allow a large variety of possible shapes and is very simple to implement, with the advantage of obtaining a much wider family of fuzzy numbers than for standard LR model (see [20–22]). This representation has the relevant advantage of being applied to the same $[0, 1]$ interval for all the fuzzy numbers involved in the computations.

In many fields of different sciences (physics, engineering, economics, social, and political sciences) and disciplines, where fuzzy sets and fuzzy logic are applied (e.g., approximate reasoning, image processing, fuzzy systems modeling and control, fuzzy decision making, statistics, operations research and optimization, computational engineering, artificial intelligence, and fuzzy finance and business) fuzzy numbers and arithmetic play a central role and are frequently and increasingly the main instruments (see [1, 9, 11, 12, 17, 19, 23, 24]).

A significant research activity has been devoted to the approximation of fuzzy numbers and fuzzy arithmetic operations, by following essentially two approaches: the first is based on approximating the non-linearities introduced by the operations, e.g., multiplication and division (see [20, 21] and references therein); the other consists in producing trapezoidal (linear) approximations based on the minimization of appropriate distance measures to obtain preservation of desired elements like expected intervals, values, ambiguities, correlation, and properties such as ordering, invariancy to translation, and scale transformation (see [25–29]). An advantage of the second approach is that, in general, the shape representations are simplified, but possibly uncontrolled errors are introduced by forcing linearization; on the other hand, the first approach has the advantage of better approximating the shape of the fuzzy numbers and this allows in most cases to control and reduce the errors but with a computational cost associated with the handling of non-linearities.

A difficulty in the adoption of fuzzy modeling is related to the fact that, from a mathematical and a practical view, fuzzy numbers do not have the same algebraic properties common to the algebra of

real numbers (e.g., a group algebraic structure) as, for example, the lack of inverses in fuzzy arithmetic (see [30]).

It follows that modeling fuzzy numbers and performing fuzzy calculations has many facets and possible solutions have to balance simple representations and approximated calculations with a sufficient control in error propagation.

The organization of the chapter is the following: Section 12.2 contains an introduction to the fuzzy numbers in the unidimensional and multidimensional cases; Section 12.3 introduces some simple and flexible representations of the fuzzy numbers, based on shape-function modeling; in Section 12.4 the fundamental elements of the fuzzy operations and calculus are given; in Sections 12.5 and 12.6 we describe the procedures and detail some algorithms for the fuzzy arithmetic operations; and in Section 12.7 we illustrate some extensions to fuzzy mathematics (integration and differentiation of fuzzy-valued functions, fuzzy differential equations). The final Section 12.8 contains a brief account of recent applications and some concluding remarks.

12.2 Fuzzy Quantities and Numbers

We will consider fuzzy quantities, i.e., fuzzy sets defined over the field \mathbb{R} of real numbers and the n -dimensional space \mathbb{R}^n . In particular we will focus on particular fuzzy quantities, called fuzzy numbers, having a particular form of the membership function.

Definition 1. A general fuzzy set over a given set (or space) \mathbb{X} of elements (the universe) is usually defined by its membership function $\mu : \mathbb{X} \rightarrow \mathbb{T} \subseteq [0, 1]$ and a fuzzy (sub)set u of \mathbb{X} is uniquely characterized by the pairs $(x, \mu_u(x))$ for each $x \in \mathbb{X}$; the value $\mu_u(x) \in [0, 1]$ is the membership grade of x to the fuzzy set u .

If $\mathbb{T} = \{0, 1\}$ (i.e., μ_u assumes only the two values 0 or 1), we obtain a subset of \mathbb{X} in the classical set-theoretic sense (what is called a *crisp* set in the fuzzy context) and μ_u is simply the characteristic function of u .

Denote by $\mathcal{F}(\mathbb{X})$ the collection of all the fuzzy sets over \mathbb{X} . Elements of $\mathcal{F}(\mathbb{X})$ will be denoted by letters u, v, w and the corresponding membership functions by μ_u, μ_v, μ_w . Of our interest are fuzzy sets when the space \mathbb{X} is \mathbb{R} (unidimensional real fuzzy sets) or \mathbb{R}^n (multidimensional real fuzzy sets).

Fundamental concepts in fuzzy theory are the *support*, the *level sets* (or *level cuts*), and the *core* of a fuzzy set (or of its membership function).

Definition 2. Let μ_u be the membership function of a fuzzy set u over \mathbb{X} . The support of u is the (crisp) subset of points of \mathbb{X} at which the membership grade $\mu_u(x)$ is positive:

$$\text{supp}(u) = \{x \mid x \in \mathbb{X}, \mu_u(x) > 0\}; \quad (1)$$

we always assume that $\text{supp}(u) \neq \emptyset$. For $\alpha \in]0, 1]$, the α -level cut of u (or simply the α -cut) is defined by

$$[u]_\alpha = \{x \mid x \in \mathbb{X}, \mu_u(x) \geq \alpha\} \quad (2)$$

and for $\alpha = 0$ (or $\alpha \rightarrow +0$) by the closure of the support

$$[u]_0 = \text{cl}\{x \mid x \in \mathbb{X}, \mu_u(x) > 0\}.$$

The core of u is the set of elements of \mathbb{X} having membership grade 1

$$\text{core}(u) = \{x \mid x \in \mathbb{X}, \mu_u(x) = 1\} \quad (3)$$

and we say that u is normal if $\text{core}(u) \neq \emptyset$.

Well-known properties of the *level-cuts* are

$$[u]_\alpha \subseteq [u]_\beta \quad \text{for } \alpha > \beta, \quad (4)$$

$$[u]_\alpha = \bigcap_{\beta < \alpha} [u]_\beta \quad \text{for } \alpha \in [0, 1] \quad (5)$$

and (if $x \in \text{supp}(u)$, otherwise $\mu_u(x) = 0$)

$$\mu_u(x) = \sup \{\alpha \mid \alpha \in [0, 1] \text{ for which } x \in [u]_\alpha\}. \quad (6)$$

A particular class of fuzzy sets $u \in \mathcal{F}(\mathbb{R}^n)$ is when the support is a convex set (C is said *convex* if $(1-t)x' + tx'' \in C$ for every $x', x'' \in C$ and all $t \in [0, 1]$) and the membership function is quasi concave:

Definition 3. Consider $u \in \mathcal{F}(\mathbb{R}^n)$, $n \geq 1$, and assume that $\text{supp}(u)$ is a convex set; we say that the membership function μ_u is *quasi concave* if

$$\mu_u((1-t)x' + tx'') \geq \min\{\mu_u(x'), \mu_u(x'')\}$$

for every $x', x'' \in \text{supp}(u)$ and $t \in [0, 1]$. Equivalently, μ_u is *quasi concave* if the level sets $[u]_\alpha$ are convex sets for all $\alpha \in [0, 1]$.

A third property of the fuzzy quantities is related to the semicontinuity of the membership function and to the closedness of the level cuts:

Definition 4. Consider $u \in \mathcal{F}(\mathbb{R}^n)$, $n \geq 1$; its membership function is said to be *upper semicontinuous* if at every $\hat{x} \in \text{supp}(u)$

$$\lim_{x \rightarrow \hat{x}} \sup \mu_u(x) = \mu_u(\hat{x})$$

or, equivalently, if the level cuts $[u]_\alpha$ are closed sets for all $\alpha \in [0, 1]$.

With the definitions above we can define the *fuzzy quantities*.

Definition 5. A *fuzzy quantity* is a fuzzy set $u \in \mathcal{F}(\mathbb{R}^n)$ with the following properties:

- (i) μ_u has non-empty bounded support and is normal.
 - (ii) μ_u is quasi concave.
 - (iii) μ_u is upper semicontinuous.
- (7)

or, equivalently,

- (a) $[u]_\alpha$ are non-empty convex sets for all $\alpha \in [0, 1]$.
 - (b) $[u]_\alpha$ are closed and bounded (compact) sets for all $\alpha \in [0, 1]$.
 - (c) $[u]_\alpha$ satisfy conditions (4) and (5).
- (8)

We will denote by \mathcal{F}^n the set of n -dimensional fuzzy quantities. A fundamental theorem in fuzzy theory characterizes (uniquely) fuzzy quantities $u \in \mathcal{F}^n$ either in terms of the membership function or in terms of the associated level cuts: if the membership function μ_u satisfies (i)–(iii) then its α -cuts satisfy (a)–(c) and, vice versa, if a family of sets $\{A_\alpha \subset \mathbb{R}^n \mid \alpha \in [0, 1]\}$ satisfies (a)–(c) then the membership function defined by

$$\mu(x) = \begin{cases} \sup\{\alpha \mid \alpha \in [0, 1] \text{ for which } x \in A_\alpha\}, & \text{if } x \in A_0 \\ 0, & \text{if } x \notin A_0 \end{cases}$$

has properties (i)–(iii) and defines a fuzzy quantity $u \in \mathcal{F}^n$ such that $[u]_\alpha = A_\alpha, \forall \alpha \in [0, 1]$.

By using this fact, we can structure \mathcal{F}^n by an addition and a scalar multiplication, defined either by the level sets or, equivalently, by Zadeh's extension principle. Let $u, v \in \mathcal{F}^n$ have membership functions μ_u, μ_v and α -cuts $[u]_\alpha, [v]_\alpha, \alpha \in [0, 1]$, respectively. The addition $u + v \in \mathcal{F}^n$ and the scalar multiplication $ku \in \mathcal{F}^n$ for $k \in \mathbb{R}, k \neq 0$ have membership functions (extension principle)

$$\mu_{u+v}(z) = \sup\{\min\{\mu_u(x), \mu_v(y)\} | z = x + y\} \quad (9)$$

$$\mu_{ku}(x) = \mu_u\left(\frac{x}{k}\right) \quad (10)$$

and level cuts

$$[u + v]_\alpha = [u]_\alpha + [v]_\alpha = \{x + y | x \in [u]_\alpha, y \in [v]_\alpha\} \quad (11)$$

$$[ku]_\alpha = k[u]_\alpha = \{kx | x \in [u]_\alpha\}. \quad (12)$$

\mathcal{F}^n is closed with respect to addition and scalar multiplication and can be structured as a metric space by introducing various types of metrics (see [8, 31] for a complete presentation and results). To do so, we use the Hausdorff distance between the level sets:

$$d_H([u]_\alpha, [v]_\alpha) = \max\{d^*([u]_\alpha, [v]_\alpha), d^*([v]_\alpha, [u]_\alpha)\} \quad (13)$$

$$\text{where } d^*(A, B) = \sup_{x \in A} \inf_{y \in B} \|x - y\| \text{ for } A, B \subset \mathbb{R}^n.$$

The *supremum metric* d_∞ on \mathcal{F}^n is defined by

$$d_\infty(u, v) = \sup\{d_H([u]_\alpha, [v]_\alpha) | \alpha \in [0, 1]\} \quad (14)$$

and the L_p metric d_p is

$$d_p(u, v) = \left(\int_0^1 (d_H([u]_\alpha, [v]_\alpha))^p d\alpha \right)^{1/p} \quad (15)$$

for all finite $p \geq 1$.

The spaces $(\mathcal{F}^n, d_\infty)$ and (\mathcal{F}^n, d_p) are complete but only (\mathcal{F}^n, d_p) is separable (it has a countable dense subset) for all finite $p \geq 1$. The following properties are valid for $d = d_\infty$ and $d = d_p$ ($p \geq 1$):

$$d(tu, tv) = |t| d(u, v), \forall t \neq 0, \forall u, v \in \mathcal{F}^n \quad (16)$$

$$d(u + w, v + w) = d(u, v), \forall u, v, w \in \mathcal{F}^n \quad (17)$$

$$d(u + w, v + z) \leq d(u, v) + d(w, z), \forall u, v, w, z \in \mathcal{F}^n. \quad (18)$$

12.2.1 Unidimensional Fuzzy Numbers

Definition 6. In the unidimensional case, a fuzzy quantity u is called a *fuzzy number* if $\exists \widehat{u} \in \mathbb{R}$ such that $\text{core}(u) = \{\widehat{u}\}$, and is called a *fuzzy interval* if $\exists \widehat{u}^-, \widehat{u}^+ \in \mathbb{R}, \widehat{u}^- < \widehat{u}^+$ such that $\text{core}(u) = [\widehat{u}^-, \widehat{u}^+]$. In particular, the α -cuts of a fuzzy number or interval are non-empty, compact intervals of the form

$$[u]_\alpha = [u_\alpha^-, u_\alpha^+] \subset \mathbb{R}. \quad (19)$$

We denote by $\mathbb{F}_{\mathbb{I}}$ the set of fuzzy intervals and by $\mathbb{F} \subset \mathbb{F}_{\mathbb{I}}$ the set of fuzzy numbers.

If $u_\alpha^- = \widehat{u}^-$ and $u_\alpha^+ = \widehat{u}^+, \forall \alpha \in [0, 1]$, we have a crisp interval or a crisp number; we denote by $\widehat{\mathbb{F}}_{\mathbb{I}}$ and $\widehat{\mathbb{F}}$ the corresponding sets. If $\widehat{u}^- = \widehat{u}^+ = 0$, we obtain a 0-fuzzy number and denote the corresponding

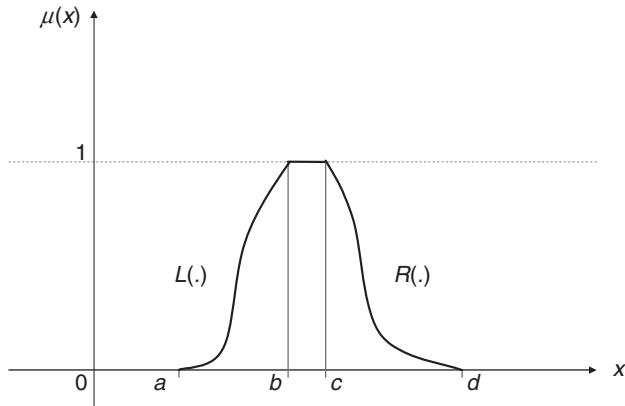


Figure 12.1 Membership function of an LR fuzzy number. $[a, d]$ is the support and $[b, c]$ is the core

set by \mathbb{F}_0 . If $u_\alpha^- + u_\alpha^+ = \hat{u}^+ + \hat{u}^-$, $\forall \alpha \in [0, 1]$, then the fuzzy interval is called symmetric; we denote by $\mathbb{S}_{\mathbb{I}}$ and by \mathbb{S} the sets of the symmetric fuzzy intervals and numbers; $\mathbb{S}_0 = \mathbb{S} \cap \mathbb{F}_0$ will be the set of the symmetric 0-fuzzy numbers. We say that u is positive if $u_\alpha^- > 0$, $\forall \alpha \in [0, 1]$ and that u is negative if $u_\alpha^+ < 0$, $\forall \alpha \in [0, 1]$; the sets of the positive and negative fuzzy numbers are denoted by \mathbb{F}_+ and \mathbb{F}_- respectively and their symmetric subsets by \mathbb{S}_+ and \mathbb{S}_- .

A well-known theorem in (generalized) convexity states that a function of a single variable over an interval I , $\mu : I \rightarrow [0, 1]$ is quasi concave if and only if I can be partitioned into two subintervals I_1 and I_2 such that μ is non-decreasing over I_1 and non-increasing over I_2 ; it follows that a quasi-concave membership function is formed of two monotonic branches, one on the left subinterval I_1 and one on the right subinterval I_2 ; further, if it reaches the maximum value in more than one point, there exists a third central subinterval where it is constant (and maximal). This is the basis for the so-called *LR fuzzy numbers*, as in Figure 12.1.

Definition 7. An LR fuzzy quantity (number or interval) u has membership function of the form

$$\mu_u(x) = \begin{cases} L\left(\frac{b-x}{b-a}\right) & \text{if } a \leq x \leq b \\ 1 & \text{if } b \leq x \leq c \\ R\left(\frac{x-c}{d-c}\right) & \text{if } c \leq x \leq d \\ 0 & \text{otherwise,} \end{cases} \quad (20)$$

where $L, R : [0, 1] \rightarrow [0, 1]$ are two non-increasing shape functions such that $R(0) = L(0) = 1$ and $R(1) = L(1) = 0$. If $b = c$, we obtain a fuzzy number.

If L and R are invertible functions, then the α -cuts are obtained by

$$[u]_\alpha = [b - (b-a)L^{-1}(\alpha), c + (d-c)R^{-1}(\alpha)]. \quad (21)$$

The usual notation for an LR fuzzy quantity is $u = \langle a, b, c, d \rangle_{L,R}$ for an interval and $u = \langle a, b, c \rangle_{L,R}$ for a number. We refer to functions $L(\cdot)$ and $R(\cdot)$ as the left and right branches (shape functions) of u , respectively.

On the other hand, the *level cuts* of a fuzzy number are ‘nested’ closed intervals and this property is the basis for the LU representation.

Definition 8. An LU fuzzy quantity (number or interval) u is completely determined by any pair $u = (u^-, u^+)$ of functions u^- , $u^+ : [0, 1] \rightarrow \mathbb{R}$, defining the endpoints of the α -cuts, satisfying the three conditions: (i) $u^- : \alpha \rightarrow u_\alpha^- \in \mathbb{R}$ is a bounded monotonic non-decreasing left-continuous function

$\forall \alpha \in]0, 1]$ and right-continuous for $\alpha = 0$; (ii) $u^+ : \alpha \rightarrow u_\alpha^+ \in \mathbb{R}$ is a bounded monotonic non-increasing left-continuous function $\forall \alpha \in]0, 1]$ and right-continuous for $\alpha = 0$; (iii) $u_\alpha^- \leq u_\alpha^+, \forall \alpha \in [0, 1]$.

The support of u is the interval $[u_0^-, u_0^+]$ and the core is $[u_1^-, u_1^+]$. If $u_1^- < u_1^+$, we have a fuzzy interval and if $u_1^- = u_1^+$ we have a fuzzy number. We refer to the functions $u_{(.)}^-$ and $u_{(.)}^+$ as the lower and upper branches on u , respectively.

The obvious relation between u^-, u^+ , and the membership function μ_u is

$$\mu_u(x) = \sup\{\alpha | x \in [u_\alpha^-, u_\alpha^+]\}. \quad (22)$$

In particular, if the two branches $u_{(.)}^-$ and $u_{(.)}^+$ are continuous invertible functions then $\mu_u(.)$ is formed by two continuous branches, the left being the increasing inverse of $u_{(.)}^-$ on $[u_0^-, u_1^-]$ and the right the decreasing inverse of $u_{(.)}^+$ on $[u_1^+, u_0^+]$.

There are many choices for functions $L(.)$, $R(.)$ (and correspondingly for $u_{(.)}^-$ and $u_{(.)}^+$); note that the same model function is valid both for L and for R (or $u_{(.)}^-$ and $u_{(.)}^+$). Simple examples are ($p = 1$ for linear shapes)

$$L(t) = (1 - t)^p \text{ with } p > 0, t \in [0, 1] \text{ and} \quad (23)$$

$$R(t) = 1 - t^p \text{ with } p > 0, t \in [0, 1].$$

Some more general forms can be obtained by orthogonal polynomials: for $i = 1, 2, \dots, p$, $p \geq 1$,

$$L(t) = \varphi_{i,p}(t) = \sum_{j=0}^i B_{j,p}(t), t \in [0, 1], \quad (24)$$

where $B_{j,p}(t)$ is the j th Bernstein polynomial of degree p given by $B_{j,p}(t) = \frac{p!}{j!(p-j)!} t^j (1-t)^{p-j}$, $j = 0, 1, \dots, p$.

Analogous forms can be used for the LU fuzzy quantities (as in Figure 12.2). If we start with an increasing shape function $p(.)$ such that $p(0) = 0$, $p(1) = 1$, and a decreasing shape function $q(.)$ such that $q(0) = 1$, $q(1) = 0$, and with four numbers $u_0^- \leq u_1^- \leq u_1^+ \leq u_0^+$ defining the support and the core of $u = (u^-, u^+)$, then we can model $u_{(.)}^-$ and $u_{(.)}^+$ by

$$\begin{aligned} u_\alpha^- &= u_0^- + (u_1^- - u_0^-)p(\alpha) \quad \text{and} \\ u_\alpha^+ &= u_1^+ + (u_0^+ - u_1^+)q(\alpha) \quad \text{for all } \alpha \in [0, 1]. \end{aligned} \quad (25)$$

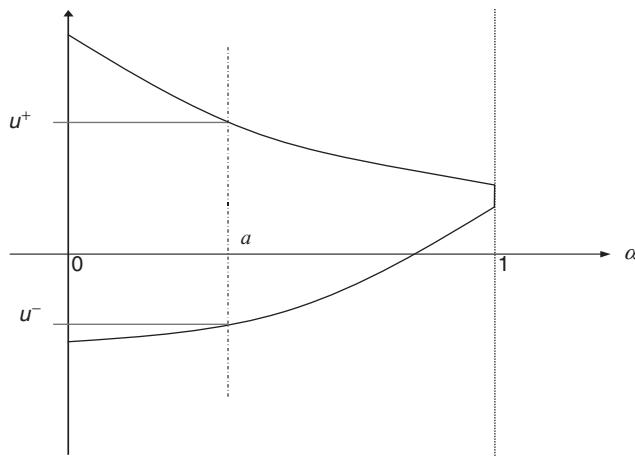


Figure 12.2 Upper and lower branches of an LU fuzzy number. For each $\alpha \in [0, 1]$ the functions u^- and u^+ form the α -cuts $[u_\alpha^-, u_\alpha^+]$

The simplest fuzzy quantities have linear branches (in LR and LU representations):

Definition 9. A trapezoidal fuzzy interval, denoted by $u = \langle a, b, c, d \rangle$, where $a \leq b \leq c \leq d$, has α -cuts

$$[u]_\alpha = [a + \alpha(b - a), d - \alpha(d - c)], \alpha \in [0, 1],$$

and membership function

$$\mu_{\text{Tra}}(x) = \begin{cases} \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ 1 & \text{if } b \leq x \leq c \\ \frac{d-x}{d-c} & \text{if } c \leq x \leq d \\ 0 & \text{otherwise.} \end{cases}$$

Some authors use the equivalent notation $u = \langle b, p, c, q \rangle$, with $p = b - a \geq 0$ and $q = d - c \geq 0$ so that the support of u is $[b - p, c + q]$ and the core is $[b, c]$.

Definition 10. A triangular fuzzy number, denoted by $u = \langle a, b, c \rangle$, where $a \leq b \leq c$, has α -cuts

$$[u]_\alpha = [a + \alpha(b - a), c - \alpha(c - b)], \alpha \in [0, 1],$$

and membership function

$$\mu_{\text{Tri}}(x) = \begin{cases} \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{if } b \leq x \leq c \\ 0 & \text{otherwise.} \end{cases}$$

Some authors use the equivalent notation $u = \langle b, p, q \rangle$, with $p = b - a \geq 0$ and $q = c - b \geq 0$ so that the support of u is $[b - p, b + q]$ and the core is $\{b\}$.

Other forms of fuzzy numbers have been proposed in the literature, e.g., the quasi-Gaussian membership function ($m \in \mathbb{R}$, $k, \sigma \in \mathbb{R}^+$, and if $k \rightarrow +\infty$, the support is unbounded)

$$\mu_{\text{qGauss}}(x) = \begin{cases} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) & \text{if } m - k\sigma \leq x \leq m + k\sigma \\ 0 & \text{otherwise,} \end{cases}$$

and the hyperbolic tangent membership function

$$\mu_{\text{hTangent}}(x) = \begin{cases} 1 + \tanh\left(-\frac{(x-m)^2}{\sigma^2}\right) & \text{if } m - k\sigma \leq x \leq m + k\sigma \\ 0 & \text{otherwise.} \end{cases}$$

To have continuity and $\mu = 0$ at the extreme values of the support $[m - k\sigma, m + k\sigma]$, we modify the fuzzy membership functions above to the following:

$$\mu(x) = \begin{cases} \frac{\exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) - \exp\left(-\frac{k^2}{2}\right)}{1 - \exp\left(-\frac{k^2}{2}\right)} & \text{if } m - k\sigma \leq x \leq m + k\sigma \\ 0 & \text{otherwise,} \end{cases} \quad (26)$$

and

$$\mu(x) = \begin{cases} \frac{\tanh(-k^2) - \tanh\left(-\frac{(x-m)^2}{\sigma^2}\right)}{\tanh(-k^2)} & \text{if } m - k\sigma \leq x \leq m + k\sigma. \\ 0 & \text{otherwise.} \end{cases} \quad (27)$$

12.2.2 Multidimensional Fuzzy Quantities

Any quasi-concave upper-semicontinuous membership function $\mu_u : \mathbb{R}^n \rightarrow [0, 1]$, with compact support and non-empty core, defines a fuzzy quantity $u \in \mathcal{F}^n$ and it can be considered as a general possibility distribution (see [32–34]).

A membership function $\mu_j : \mathbb{R} \rightarrow [0, 1]$ is called the j -th marginal of $\mu_u : \mathbb{R}^n \rightarrow [0, 1]$ if, for all $x \in \mathbb{R}$,

$$\mu_j(x) = \max\{\mu_u(x_1, \dots, x_{j-1}, x, x_{j+1}, \dots, x_n) \mid x_i \in \mathbb{R}, i \neq j\} \quad (28)$$

and the corresponding fuzzy set (i.e., having μ_j as membership function) is called the j th projection of $u \in \mathcal{F}^n$. It is obvious that the availability of all the projections is not sufficient, in general, to reconstruct the original membership function μ_u and we say that the projections are *interacting* each other. (For a discussion of interacting fuzzy numbers see [11, 35, 36].)

Particular n -dimensional membership functions can be obtained by the Cartesian product of n unidimensional fuzzy numbers or intervals.

Let $u_j \in \mathbb{F}_{\mathbb{I}}$ have membership functions $\mu_{u_j}(x_j)$ for $j = 1, 2, \dots, n$; the membership function of the vector $u = (u_1, \dots, u_n)$ of *non-interacting* fuzzy quantities $u_j \in \mathbb{F}_{\mathbb{I}}$ is defined by (or satisfies)

$$\mu_u(x_1, \dots, x_n) = \min\{\mu_{u_j}(x_j), j = 1, 2, \dots, n\}.$$

In this case, if the α -cuts of u_j are $[u_j]_\alpha = [u_{j,\alpha}^-, u_{j,\alpha}^+]$, $\alpha \in [0, 1]$, $j = 1, 2, \dots, n$, then the α -cuts of u are the cartesian products

$$[u]_\alpha = [u_1]_\alpha \times \dots \times [u_n]_\alpha = [u_{1,\alpha}^-, u_{1,\alpha}^+] \times \dots \times [u_{n,\alpha}^-, u_{n,\alpha}^+]. \quad (29)$$

For non-interacting fuzzy quantities, the availability of the projections is sufficient to define the vector; we denote by $\mathbb{F}_{\mathbb{I}}^n$ (or by \mathbb{F}^n if all $u_j \in \mathbb{F}$) the corresponding set.

Fuzzy calculations with interacting numbers are in general quite difficult, with few exceptions; in the following we will consider fuzzy arithmetic based on unidimensional and multidimensional non-interacting fuzzy quantities.

12.3 Representation of Fuzzy Numbers

As we have seen in the previous section, the LR and the LU representations of fuzzy numbers require to use appropriate (monotonic) shape functions to model either the left and right branches of the membership function or the lower and upper branches of the α -cuts.

In this section we present the basic elements of a parametric representation of the shape functions proposed in [20] and [21] based on monotonic Hermite-type interpolation. The parametric representations can be used both to define the shape functions and to calculate the arithmetic operations by error-controlled approximations.

We first introduce some models for ‘standardized’ differentiable monotonic shape functions $p : [0, 1] \rightarrow [0, 1]$ such that

$$p(0) = 0 \quad \text{and} \quad p(1) = 1 \quad \text{with } p(t) \text{ increasing on } [0, 1];$$

if interested in decreasing functions, we can start with an increasing function $p(\cdot)$ and simply define corresponding decreasing functions $q : [0, 1] \rightarrow [0, 1]$ by

$$q(t) = 1 - p(t) \quad \text{or} \quad q(t) = p(\varphi(t)),$$

where $\varphi : [0, 1] \rightarrow [0, 1]$ is any decreasing bijection (e.g., $\varphi(t) = 1 - t$).

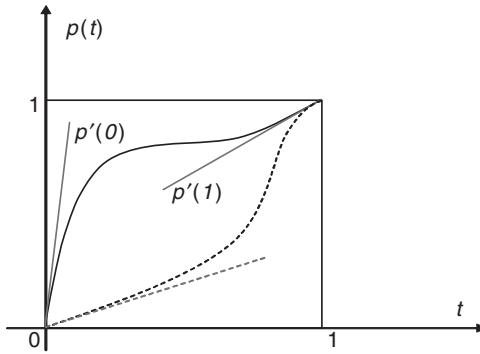


Figure 12.3 Standard form of the monotonic Hermite-type interpolation function: $p(0) = 0$, $p(1) = 1$ and $p'(0) = \beta_0$, $p'(1) = \beta_1$

As illustrated in [21], increasing functions $p : [0, 1] \rightarrow [0, 1]$ satisfying the four Hermite-type interpolation conditions

$$p(0) = 0, \quad p(1) = 1 \quad \text{and} \quad p'(0) = \beta_0, \quad p'(1) = \beta_1$$

for any value of the two non-negative parameters $\beta_i \geq 0$, $i = 0, 1$, can be used as valid shape function: we obtain infinite many functions simply by fixing the two parameters β_i that give the slopes (first derivatives) of the function at $t = 0$ and $t = 1$ (see Figure 12.3). To explicit the slope parameters we denote the interpolating function by

$$t \rightarrow p(t; \beta_0, \beta_1) \quad \text{for } t \in [0, 1].$$

We recall here two of the basic forms illustrated in [21]:

- (2,2)-Rational spline:

$$p(t; \beta_0, \beta_1) = \frac{t^2 + \beta_0 t(1-t)}{1 + (\beta_0 + \beta_1 - 2)t(1-t)}. \quad (30)$$

- Mixed exponential spline:

$$p(t; \beta_0, \beta_1) = \frac{1}{a}[t^2(3-2t) + \beta_0 - \beta_0(1-t)^a + \beta_1 t^a], \quad (31)$$

where $a = 1 + \beta_0 + \beta_1$.

Note that in (30) and (31) we obtain a linear $p(t) = t$, $\forall t \in [0, 1]$, if $\beta_0 = \beta_1 = 1$ and a quadratic $p(t) = t^2 + \beta_0 t(1-t)$ if $\beta_0 + \beta_1 = 2$.

In order to produce different shapes we can either fix the slopes β_0 and β_1 (if we have information on the first derivatives at $t = 0, t = 1$) or we can estimate them by knowing the values of $p(t)$ in additional points.

For example, if $0 < p_1 < \dots < p_k < 1$ are given $k \geq 2$ increasing values of $p(t_i)$, $i = 1, \dots, k$, at internal points $0 < t_1 < \dots < t_k < 1$, we can estimate the slopes by solving the following two-variable constrained minimization problem:

$$\begin{aligned} \min F(\beta_0, \beta_1) &= \sum_{j=1}^k [p(t_j; \beta_0, \beta_1) - p_j]^2 \\ \text{s.t. } \beta_0, \beta_1 &\geq 0. \end{aligned} \quad (32)$$

If the data $1 > q_1 > \dots > q_k > 0$ are decreasing (as for the right or upper branches), the minimization (32) will have the objective function

$$G(\beta_0, \beta_1) = \sum_{j=1}^k [1 - p(t_j; \beta_0, \beta_1) - q_j]^2.$$

The model functions above can be adopted not only to define globally the shapes, but also to represent the functions ‘piecewise’, on a decomposition of the interval $[0, 1]$ into N subintervals

$$0 = \alpha_0 < \alpha_1 < \dots < \alpha_{i-1} < \alpha_i < \dots < \alpha_N = 1.$$

It is convenient to use the same subdivision for both the lower u_α^- and upper u_α^+ branches (we can always reduce to this situation by the union of two different subdivisions). We have a preference in using a uniform subdivision of the interval $[0, 1]$ and in refining the decomposition by successively bisecting each subinterval, producing $N = 2^K, K \geq 0$. In each subinterval $I_i = [\alpha_{i-1}, \alpha_i]$, the values and the slopes of the two functions are

$$\begin{aligned} u_{(\alpha_{i-1})}^- &= u_{0,i}^-, u_{(\alpha_{i-1})}^+ = u_{0,i}^+, u_{(\alpha_i)}^- = u_{1,i}^-, u_{(\alpha_i)}^+ = u_{1,i}^+, \\ u_{(\alpha_{i-1})}'^- &= d_{0,i}^-, u_{(\alpha_{i-1})}'^+ = d_{0,i}^+, u_{(\alpha_i)}'^- = d_{1,i}^-, u_{(\alpha_i)}'^+ = d_{1,i}^+, \end{aligned} \quad (33)$$

and by the transformation $t_\alpha = \frac{\alpha - \alpha_{i-1}}{\alpha_i - \alpha_{i-1}}$, $\alpha \in I_i$, each subinterval I_i is mapped into the standard $[0, 1]$ interval to determine each piece independently and obtain general left-continuous LU fuzzy numbers. Globally continuous or more regular $C^{(1)}$ fuzzy numbers can be obtained directly from the data (e.g., $u_{1,i}^- = u_{0,i+1}^-, u_{1,i}^+ = u_{0,i+1}^+$ for continuity and $d_{1,i}^- = d_{0,i+1}^-, d_{1,i}^+ = d_{0,i+1}^+$ for differentiability at $\alpha = \alpha_i$).

Let $p_i^\pm(t)$ denote the model function on I_i ; we easily obtain

$$p_i^-(t) = p(t; \beta_{0,i}^-, \beta_{1,i}^-), p_i^+(t) = 1 - p(t; \beta_{0,i}^+, \beta_{1,i}^+), \quad (34)$$

$$\text{with } \beta_{j,i}^- = \frac{\alpha_i - \alpha_{i-1}}{u_{1,i}^- - u_{0,i}^-} d_{j,i}^- \text{ and } \beta_{j,i}^+ = -\frac{\alpha_i - \alpha_{i-1}}{u_{1,i}^+ - u_{0,i}^+} d_{j,i}^+ \text{ for } j = 0, 1,$$

so that, for $\alpha \in [\alpha_{i-1}, \alpha_i]$ and $i = 1, 2, \dots, N$,

$$u_\alpha^- = u_{0,i}^- + (u_{1,i}^- - u_{0,i}^-) p_i^-(t_\alpha), t_\alpha = \frac{\alpha - \alpha_{i-1}}{\alpha_i - \alpha_{i-1}}; \quad (35)$$

$$u_\alpha^+ = u_{0,i}^+ + (u_{1,i}^+ - u_{0,i}^+) p_i^+(t_\alpha), t_\alpha = \frac{\alpha - \alpha_{i-1}}{\alpha_i - \alpha_{i-1}}. \quad (36)$$

12.3.1 Parametric LU Fuzzy Numbers

The monotonic models illustrated in the previous section suggest a first parametrization of fuzzy numbers obtained by representing the lower and upper branches u_α^- and u_α^+ of u on the trivial decomposition of interval $[0, 1]$, with $N = 1$ (without internal points) and $\alpha_0 = 0, \alpha_1 = 1$. In this simple case, u can be represented by a vector of eight components: (The slopes corresponding to u_i^- are denoted by δu_i^- , etc.)

$$u = (u_0^-, \delta u_0^-, u_0^+, \delta u_0^+; u_1^-, \delta u_1^-, u_1^+, \delta u_1^+), \quad (37)$$

where $u_0^-, \delta u_0^-, u_1^-, \delta u_1^-$ are used for the lower branch u_α^- , and $u_0^+, \delta u_0^+, u_1^+, \delta u_1^+$ for the upper branch u_α^+ .

On a decomposition $0 = \alpha_0 < \alpha_1 < \dots < \alpha_N = 1$ we can proceed piecewise. For example, a differentiable shape function requires $4(N + 1)$ parameters

$$\begin{aligned} u &= (\alpha_i; u_i^-, \delta u_i^-, u_i^+, \delta u_i^+);_{i=0,1,\dots,N} \text{ with} \\ u_0^- &\leq u_1^- \leq \dots \leq u_N^- \leq u_N^+ \leq u_{N-1}^+ \leq \dots \leq u_0^+ \text{ (data)} \\ \delta u_i^- &\geq 0, \delta u_i^+ \leq 0 \text{ (slopes)}, \end{aligned} \quad (38)$$

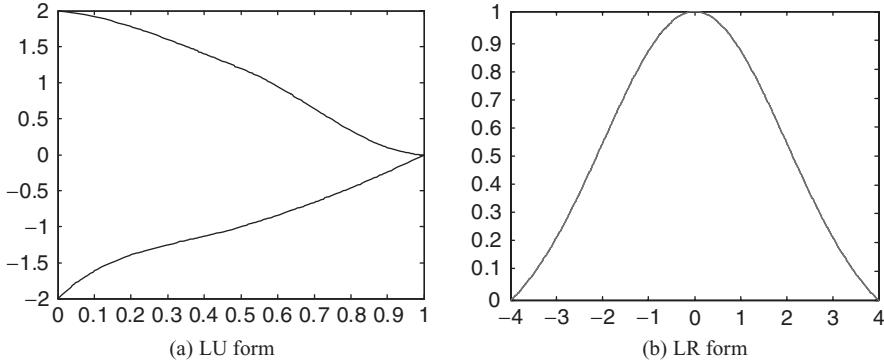


Figure 12.4 LU and LR parametric Fuzzy numbers. (a) Fuzzy number in LU representation; the parameters are reported in (39) and the construction is obtained by the mixed spline with $N = 2$. (b) Quasi-Gaussian fuzzy number; the parameters are reported in (43) and the membership function is obtained by the mixed spline with $N = 4$.

and the branches are computed according to (35) and (36). An example with $N = 4$ is in (39) and is plotted in Figure 12.4a.

LU parametrization of a fuzzy number (39)

α_i	u_i^-	δu_i^-	u_i^+	δu_i^+
0.0	-2.0	5.0	2.0	-0.5
0.5	-1.0	1.5	1.2	-2.0
1.0	0.0	2.5	0.0	-0.1

12.3.2 Parametric LR Fuzzy Numbers

The (parametric) monotonic splines can be used as models for the shape functions L and R ; in fact, if $\beta_0, \beta_1 \geq 0$ are given and we consider

$$q(t; \beta_0, \beta_1) = 1 - p(t; \beta_0, \beta_1), \quad (40)$$

then $q(0) = 1$, $q(1) = 0$, $q'(0) = -\beta_0$, $q'(1) = -\beta_1$ and we can write the L and R shapes as

$$\begin{cases} L(t) = q(t; \beta_{0,L}, \beta_{1,L}) \\ R(t) = q(t; \beta_{0,R}, \beta_{1,R}). \end{cases} \quad (41)$$

An LR fuzzy number can be obtained by using (40) and (41) with the parameters

$$u_{LR} = (u_{0,L}, \delta u_{0,L}, u_{0,R}, \delta u_{0,R}; \quad u_{1,L}, \delta u_{1,L}, u_{1,R}, \delta u_{1,R}), \quad (42)$$

provided that $u_{0,L} \leq u_{1,L} \leq u_{1,R} \leq u_{0,R}$ and the slopes $\delta u_{0,L}, \delta u_{1,L} \geq 0$ and $\delta u_{0,R}, \delta u_{1,R} \leq 0$ are the first derivatives of the membership function μ in (20) at the points $x = u_{0,L}$, $x = u_{1,L}$, $x = u_{0,R}$, and $x = u_{1,R}$, respectively. The β parameters in (41) are related to the slopes by the equations

$$\begin{aligned} \delta u_{0,L} &= \frac{\beta_{1,L}}{u_{1,L} - u_{0,L}} \geq 0, & \delta u_{1,L} &= \frac{\beta_{0,L}}{u_{1,L} - u_{0,L}} \geq 0; \\ \delta u_{0,R} &= \frac{\beta_{1,R}}{u_{1,R} - u_{0,R}} \leq 0, & \delta u_{1,R} &= \frac{\beta_{0,R}}{u_{1,R} - u_{0,R}} \leq 0. \end{aligned}$$

On a decomposition $0 = \alpha_0 < \alpha_1 < \dots < \alpha_N = 1$ we proceed similarly to (38).

As two examples, the LR parametrization of a fuzzy Quasi-Gaussian number ($m = 0$, $\sigma = 2$, and $k = 2$) approximated with $N = 4$ (five points) is (see Figure 12.4b),

LR parametrization of fuzzy number (26) (43)

α_i	$u_{i,L}$	$\delta u_{i,L}$	$u_{i,R}$	$\delta u_{i,R}$
0.0	-4.0	0.156518	4.0	-0.156518
0.25	-2.8921	0.293924	2.8921	-0.293924
0.5	-2.1283	0.349320	2.1283	-0.349320
0.75	-1.3959	0.316346	1.3959	-0.316346
1.0	0.0	0.0	0.0	0.0

and of a hyperbolic tangent fuzzy number ($m = 0$, $\sigma = 3$, and $k = 1$) is

LR parametrization of fuzzy number (27) (44)

α_i	$u_{i,L}$	$\delta u_{i,L}$	$u_{i,R}$	$\delta u_{i,R}$
0.0	-3.0	0.367627	3.0	-0.367627
0.25	-2.4174	0.475221	2.4174	-0.475221
0.5	-1.8997	0.473932	1.8997	-0.473932
0.75	-1.3171	0.370379	1.3171	-0.370379
1.0	0.0	0.0	0.0	0.0

The representations are exact at the nodes α_i and the average absolute errors in the membership functions (calculated in 1000 uniform x values of the corresponding supports $[-4, 4]$ and $[-3, 3]$) are 0.076% and 0.024% respectively.

12.3.3 Switching LR and LU

The LU and LR parametric representations of fuzzy numbers produce subspaces of the space of fuzzy numbers. Denote by \mathbb{F}^{LU} and by \mathbb{F}^{LR} the sets of (differentiable shaped) fuzzy numbers defined by (37) and (42), respectively, and by \mathbb{F}_N^{LU} and by \mathbb{F}_N^{LR} the corresponding extensions to a uniform decomposition $\alpha_i = \frac{i}{N} \in [0, 1]$, $i = 0, 1, \dots, N$, into N subintervals. By using equations (46) there is a one-to-one correspondence between \mathbb{F}_N^{LU} and \mathbb{F}_N^{LR} so that we can go equivalently from a representation to the other.

For example, for the case $N = 1$, let $u_\alpha^- = u_0^- + (u_1^- - u_0^-)p(\alpha; \beta_0^-, \beta_1^-)$ and $u_\alpha^+ = u_1^+ + (u_1^+ - u_0^+)p(\alpha; \beta_0^+, \beta_1^+)$ be the lower and upper functions of the LU representation of a fuzzy number $u \in \mathbb{F}^{LU}$; the LR representation of u has the membership function

$$\mu(x) = \begin{cases} p^{-1}\left(\frac{x-u_0^-}{u_1^- - u_0^-}; \beta_0^-, \beta_1^-\right) & \text{if } x \in [u_0^-, u_1^-] \\ 1 & \text{if } x \in [u_1^-, u_1^+] \\ p^{-1}\left(\frac{u_1^+ - x}{u_1^+ - u_0^+}; \beta_0^+, \beta_1^+\right) & \text{if } x \in [u_1^+, u_0^+] \\ 0 & \text{otherwise,} \end{cases} \quad (45)$$

where $\alpha = p^{-1}(t; \beta_0, \beta_1)$ is the inverse function of $t = p(\alpha; \beta_0, \beta_1)$. If we model the LU fuzzy numbers by a (2,2)-rational spline $p(\alpha; \beta_0, \beta_1)$ like (30), the inverse $p^{-1}(t; \beta_0, \beta_1)$ can be computed analytically as we have to solve the quadratic equation (with respect to α)

$$\begin{aligned} \alpha^2 + \beta_0\alpha(1 - \alpha) &= t[1 + (\beta_0 + \beta_1 - 2)\alpha(1 - \alpha)], \text{ i.e.,} \\ (1 + A(t))\alpha^2 - A(t)\alpha - t &= 0, \text{ where} \end{aligned}$$

$$A(t) = -\beta_0 + \beta_0 t + \beta_1 t - 2t.$$

If $A(t) = -1$, then the equation is linear and the solution is $\alpha = t$. If $A(t) \neq -1$, then there exist two real solutions and we choose the one belonging to $[0, 1]$.

We can also switch the two representations: for example, for a given LR fuzzy number $u \in \mathbb{F}^{LR}$ given by (42), its approximated LU representation $u \in \mathbb{F}^{LU}$ corresponding to (37) is

$$\left\{ \begin{array}{l} u_{LU} = (u_0^-, \delta u_0^-, u_0^+, \delta u_0^+; \quad u_1^-, \delta u_1^-, u_1^+, \delta u_1^+) \\ \text{with} \\ u_0^- = u_{0,L}, \delta u_0^- = \frac{1}{\delta u_{0,L}} \\ u_1^- = u_{1,L}, \delta u_1^- = \frac{1}{\delta u_{1,L}} \\ u_0^+ = u_{0,R}, \delta u_0^+ = \frac{1}{\delta u_{0,R}} \\ u_1^+ = u_{1,R}, \delta u_1^+ = \frac{1}{\delta u_{1,R}}. \end{array} \right. \quad (46)$$

(If some $\delta u_{i,L}, \delta u_{i,R}$ is zero, the corresponding infinite $\delta u_i^-, \delta u_i^+$ slope can be assigned a BIG number.)

Following [37] we can define a geometric distance $D_p(u, v)$ between fuzzy numbers $u, v \in \mathbb{F}^{LU}$, given by

$$D_p^{LU}(u, v) = [|u_0^- - v_0^-|^p + |u_1^- - v_1^-|^p + |u_0^+ - v_0^+|^p + |u_1^+ - v_1^+|^p + |\delta u_0^- - \delta v_0^-|^p + |\delta u_1^- - \delta v_1^-|^p + |\delta u_0^+ - \delta v_0^+|^p + |\delta u_1^+ - \delta v_1^+|^p]^{1/p}$$

and

$$D_\infty^{LU}(u, v) = \max \{ |u_0^- - v_0^-|, |u_1^- - v_1^-|, |u_0^+ - v_0^+|, |u_1^+ - v_1^+|, \\ |\delta u_0^- - \delta v_0^-|, |\delta u_1^- - \delta v_1^-|, |\delta u_0^+ - \delta v_0^+|, |\delta u_1^+ - \delta v_1^+| \}.$$

Analogous formulas can be introduced for \mathbb{F}^{LR} and for \mathbb{F}_N^{LU} and \mathbb{F}_N^{LR} .

In very recent years, a particular attention has been dedicated to the class of trapezoidal and triangular fuzzy numbers, as they are one of the simplest representations of fuzzy uncertainty and only four parameters are sufficient to characterize them. For this reason, several methods have been proposed to approximate fuzzy numbers by the trapezoidal family (see [25–29, 38].)

The families of fuzzy numbers \mathbb{F}^{LR} and \mathbb{F}^{LU} , which include triangular and trapezoidal fuzzy numbers, are (in the simpler form) characterized by eight parameters and it appears that the inclusion of the slopes of lower and upper functions, even without generating piecewise monotonic approximations over subintervals (i.e., working with $N = 1$) is able to capture much more information than the linear approximation (see [20, 21]).

12.4 Fuzzy Arithmetic: Basic Elements

The fuzzy extension principle introduced by Zadeh in [32, 39] is the basic tool for fuzzy calculus; it extends functions of real numbers to functions of non-interactive fuzzy quantities and it allows the extension of arithmetic operations and calculus to fuzzy arguments. We have already defined the addition (9) and the scalar multiplication (10).

Let $\circ \in \{+, -, \times, /\}$ be one of the four arithmetic operations and let $u, v \in \mathbb{F}_{\mathbb{I}}$ be given fuzzy intervals (or numbers), having $\mu_u(\cdot)$ and $\mu_v(\cdot)$ as membership functions and level cuts representations $u = (u^-, u^+)$ and $v = (v^-, v^+)$; the extension principle for the extension of \circ defines the membership function of $w = u \circ v$ by

$$\mu_{uv}(z) = \sup \{ \min\{\mu_u(x), \mu_v(y)\} \mid z = x \circ y \}. \quad (47)$$

In terms of the α -cuts, the four arithmetic operations and the scalar multiplication for $k \in \mathbb{R}$ are obtained by the well-known *interval arithmetic*:

Addition:

$$\left\{ \begin{array}{l} u + v = (u^- + v^-, u^+ + v^+) \\ \alpha \in [0, 1], [u + v]_\alpha = [u_\alpha^- + v_\alpha^-, u_\alpha^+ + v_\alpha^+]. \end{array} \right.$$

Scalar multiplication:

$$\left\{ \begin{array}{ll} ku = (ku^-, ku^+) & \text{if } k > 0 \\ ku = (ku^+, ku^-) & \text{if } k < 0 \\ \alpha \in [0, 1], [ku]_\alpha = [\min\{ku_\alpha^-, ku_\alpha^+\}, \max\{ku_\alpha^-, ku_\alpha^+\}]. \end{array} \right.$$

Subtraction:

$$\left\{ \begin{array}{ll} u - v = u + (-v) = (u^- - v^+, u^+ - v^-) \\ \alpha \in [0, 1], [u - v]_\alpha = [u_\alpha^- - v_\alpha^+, u_\alpha^+ - v_\alpha^-]. \end{array} \right.$$

Multiplication:

$$\left\{ \begin{array}{ll} u \times v = ((uv)^-, (uv)^+) \\ \alpha \in [0, 1], \left\{ \begin{array}{l} (uv)_\alpha^- = \min\{u_\alpha^- v_\alpha^-, u_\alpha^- v_\alpha^+, u_\alpha^+ v_\alpha^-, u_\alpha^+ v_\alpha^+\} \\ (uv)_\alpha^+ = \max\{u_\alpha^- v_\alpha^-, u_\alpha^- v_\alpha^+, u_\alpha^+ v_\alpha^-, u_\alpha^+ v_\alpha^+\} \end{array} \right. \end{array} \right.$$

Division:

$$\left\{ \begin{array}{ll} \frac{u}{v} = \left(\left(\frac{u}{v} \right)^-, \left(\frac{u}{v} \right)^+ \right) \text{ if } 0 \notin [v_0^-, v_0^+] \\ \alpha \in [0, 1], \left\{ \begin{array}{l} \left(\frac{u}{v} \right)_\alpha^- = \min \left\{ \frac{u_\alpha^-}{v_\alpha^-}, \frac{u_\alpha^-}{v_\alpha^+}, \frac{u_\alpha^+}{v_\alpha^-}, \frac{u_\alpha^+}{v_\alpha^+} \right\} \\ \left(\frac{u}{v} \right)_\alpha^+ = \max \left\{ \frac{u_\alpha^-}{v_\alpha^-}, \frac{u_\alpha^-}{v_\alpha^+}, \frac{u_\alpha^+}{v_\alpha^-}, \frac{u_\alpha^+}{v_\alpha^+} \right\}. \end{array} \right. \end{array} \right.$$

From an algebraic point of view, addition and multiplication are commutative and associative and have a neutral element. If we include the crisp numbers 0 and 1 into the set of fuzzy numbers with $[0]_\alpha = [0, 0] = \{0\}$ and $[1]_\alpha = [1, 1] = \{1\}$, it is easy to see that, for every $u \in \mathbb{F}_1$, $u + 0 = 0 + u = u$ (additive neutral element) and $u \times 1 = 1 \times u = u$ (multiplicative neutral element). But addition and multiplication of fuzzy numbers do not admit an inverse element:

$$\begin{aligned} u + v = w &\nrightarrow u = w - v \\ u/v = w &\nrightarrow u = vw. \end{aligned}$$

For given $u \in \mathbb{F}_1$ and real numbers p and q , it is $pu + qu \neq (p + q)u$ unless p and q have the same sign ($pq \geq 0$) so that, in particular, $u - u \neq (1 - 1)u = 0$. Analogously, $u/u \neq 1$. This implies that the ‘inversion’ of fuzzy operations, like in the cases $u + v - u$ or uv/u , is not possible in terms of what is expected of crisp numbers; e.g., $(3u^2 - 2u^2)v/u \neq uv$. It is clear that the direct application of the fuzzy extension principle to the computation of expressions $u + v - u$ and uv/u always produces the correct result v ; but it cannot generally be obtained by the iterative application of the extension principle to ‘parts’ of the expressions. For example, the two-step procedure

Step 1. $w_1 = u + v$ or $w_1 = uv$

Step 2. $w = w_1 - u$ or $w = w_1/u$

will always produce $w \neq v$ (exception if some numbers are crisp).

Also the distributivity property $(u + v)w = uw + vw$ is valid only in special cases, e.g., (for recent results see [40, 41])

$$\begin{aligned} w &\in \mathbb{F} \text{ and } u, v \in \mathbb{F}_+, \text{ or} \\ w &\in \mathbb{F} \text{ and } u, v \in \mathbb{F}_-, \text{ or} \\ w &\in \mathbb{F} \text{ and } u, v \in \mathbb{S}_0, \text{ or} \\ w &\in \mathbb{F}_+ \cup \mathbb{F}_- \text{ and } u, v \in \mathbb{F}_0. \end{aligned}$$

The simple examples given above suggest that fuzzy arithmetic has to be performed very carefully and, in particular, we cannot mimic the rules of standard crisp situations. This does not mean that fuzzy arithmetic (based on extension principle) is not compatible with crisp arithmetic; we cannot use in the fuzzy context the same algorithms and rules as for crisp calculations. Further investigation (in particular [16]) has pointed out that critical cases are related to the multiple occurrence of some fuzzy quantities in the expression to be calculated. From a mathematical point of view this is quite obvious as $\min\{f(x_1, x_2) \mid x_1 \in A, x_2 \in A\}$ is not the same as $\min\{f(x, x) \mid x \in A\}$ and, e.g.,

$$\begin{aligned}[u^2]_\alpha &= [\min\{x^2 \mid x \in [u]_\alpha\}, \max\{x^2 \mid x \in [u]_\alpha\}] \\ &\text{is not the same as} \\ [u \cdot u]_\alpha &= [\min\{xy \mid x, y \in [u]_\alpha\}, \max\{xy \mid x, y \in [u]_\alpha\}].\end{aligned}$$

In the fuzzy or in the interval arithmetic contexts, equation $u = v + w$ is not equivalent to $w = u - v = u + (-1)v$ or to $v = u - w = u + (-1)w$, and this has motivated the introduction of the following Hukuhara difference (H-difference) (see [8]):

Definition 11. Given $u, v \in \mathbb{F}$, the H-difference of u and v is defined by $u \ominus v = w \Leftrightarrow u = v + w$; if $u \ominus v$ exists, it is unique and its α -cuts are $[u \ominus v]_\alpha = [u_\alpha^- - v_\alpha^-, u_\alpha^+ - v_\alpha^+]$. Clearly, $u \ominus u = \{0\}$.

The H-difference is also motivated by the problem of inverting the addition: if x, y are crisp numbers then $(x + y) - y = x$, but this is not true if x, y are fuzzy. It is possible to see that (see [42]) if u and v are fuzzy numbers (and not in general fuzzy sets), then $(u + v) \ominus v = u$; i.e., the H-difference inverts the addition of fuzzy numbers.

Note that in defining the H-difference, also the following case can be taken into account $u \ominus' v = w \Leftrightarrow v = u + (-1)w$ and the H-difference can be generalized to the following definition.

Definition 12. Given $u, v \in \mathbb{F}$, the generalized H-difference can be defined as the fuzzy number w , if it exists, such that

$$u \ominus_g v = w \Leftrightarrow \begin{cases} \text{either} & (i) u = v + w \\ \text{or} & (ii) v = u + (-1)w \end{cases}.$$

If $u \ominus_g v$ exists, it is unique and its α -cuts are given by

$$[u \ominus_g v]_\alpha = [\min\{u_\alpha^- - v_\alpha^-, u_\alpha^+ - v_\alpha^+\}, \max\{u_\alpha^- - v_\alpha^-, u_\alpha^+ - v_\alpha^+\}].$$

If $u \ominus v$ exists, then $u \ominus v = u \ominus_g v$ and if (i) and (ii) are satisfied simultaneously, then w is a crisp number. Also, $u \ominus_g u = u \ominus u = \{0\}$.

Two simple examples on real (crisp) compact intervals illustrate the generalization (from [8, p. 8]; $[-1, 1] \ominus [-1, 0] = [0, 1]$ as in fact (i) is $[-1, 0] + [0, 1] = [-1, 1]$, but $[0, 0] \ominus_g [0, 1] = [-1, 0]$ and $[0, 1] \ominus_g [-\frac{1}{2}, 1] = [0, \frac{1}{2}]$ satisfy (ii). Note that $[a, b] \ominus_g [c, d] = [\min\{a - c, b - d\}, \max\{a - c, b - d\}]$ is always defined for real intervals).

The generalized H-difference is (implicitly) used by Bede and Gal (see [43]) in their definition of generalized differentiability of a fuzzy-valued function.

Consider now the extension of function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ to a vector of n (non-interactive) fuzzy numbers $u = (u_1, \dots, u_n) \in (\mathbb{F})^n$, with k th component $u_k \in \mathbb{F}$

$$\begin{aligned}[u_k]_\alpha &= [u_{k,\alpha}^-, u_{k,\alpha}^+] \text{ for } k = 1, 2, \dots, n \text{ (α-cuts)} \\ \mu_{u_k} &: \text{supp}(u_k) \rightarrow [0, 1] \text{ for } k = 1, 2, \dots, n \text{ (membership function)}\end{aligned}$$

and denote $v = f(u_1, \dots, u_n)$ with membership function μ_v and LU representation $v = (v^-, v^+)$; the extension principle states that μ_v is given by

$$\mu_v(y) = \begin{cases} \sup\{\min\{\mu_{u_1}(x_1), \dots, \mu_{u_n}(x_n)\} \mid y = f(x_1, \dots, x_n)\} & \text{if } y \in \text{Range}(f) \\ 0 & \text{otherwise,} \end{cases} \quad (48)$$

where $\text{Range}(f) = \{y \in \mathbb{R} \mid \exists (x_1, \dots, x_n) \in \mathbb{R}^n \text{ s.t. } y = f(x_1, \dots, x_n)\}$.

For a continuous function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, the α -cuts of the fuzzy extension v are obtained by solving the following box-constrained global optimization problems ($\alpha \in [0, 1]$):

$$v_\alpha^- = \min\{f(x_1, \dots, x_n) \mid x_k \in [u_k]_\alpha, k = 1, 2, \dots, n\}; \quad (49)$$

$$v_\alpha^+ = \max\{f(x_1, \dots, x_n) \mid x_k \in [u_k]_\alpha, k = 1, 2, \dots, n\}; \quad (50)$$

The lower and upper values v_α^- and v_α^+ of v define equivalently (as f is assumed to be continuous) the image of the cartesian product $\prod_{k=1}^n [u_k]_\alpha$ via f , i.e. (see Figure 12.5),

$$[v_\alpha^-, v_\alpha^+] = f([u_1]_\alpha, \dots, [u_n]_\alpha).$$

If function $f(x_1, \dots, x_n)$ is sufficiently simple, the analytical expressions for v_α^- and v_α^+ can be obtained, as it is the case for many unidimensional elementary functions (see, e.g., [44]). For general functions, such as polynomials or trigonometric functions, for which many min/max global points exist, we need to solve numerically the global optimization problems (49) and (50) above; general methods for global optimization have been proposed and a very extended scientific literature is available.

It is clear that in these cases we have only the possibility of fixing a finite set of values $\alpha \in \{\alpha_0, \dots, \alpha_M\}$ and obtain the corresponding v_α^- and v_α^+ pointwise; a sufficiently precise calculation requires M in the range from 10 to 100 or more (depending on the application and the required precision) and the computational time may become very high.

To reduce these difficulties, various specific heuristic methods have been proposed and all the specific methods try to take computational advantage from the specific structure of ‘nested’ optimizations (49)–(50) intrinsic in the property (4) of the α -cuts; among others, the vertex method and its variants

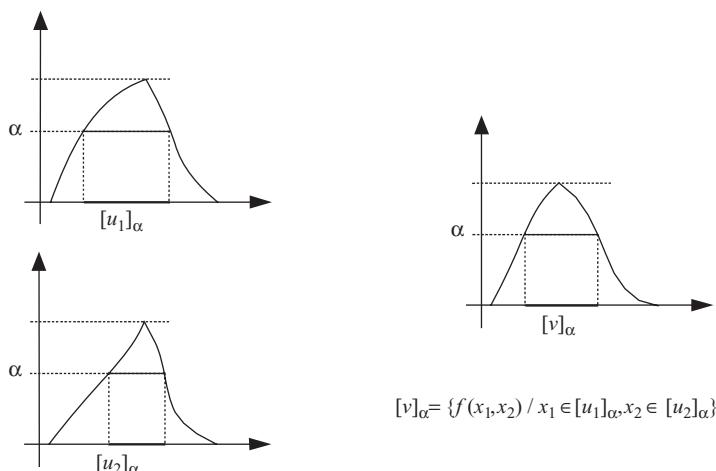


Figure 12.5 Interval view of fuzzy arithmetic. Each α -cut $[v]_\alpha$ of fuzzy number $v = f(u_1, u_2)$ is the image via function f of the α -cuts of u_1 and u_2 corresponding to the same membership level $\alpha \in [0, 1]$

(see [45–47]), the fuzzy weighted average method (see [48]), the general transformation method (see [49–51]), and the interval arithmetic optimization with sparse grids (see [52]).

The computational complexity of the algorithms is generally exponential in the number n of (distinct) operators and goes from $\mathcal{O}(M2^n)$ of the vertex method to $\mathcal{O}(M^n)$ of the complete version of the transformation method.

Since its origins, fuzzy calculus has been related and has received improvements from interval analysis (see [53, 54] and the references therein); the overestimation effect that arises in interval arithmetic when a variable has more than one occurrence in the expression to be calculated is also common to fuzzy calculus and ideas to overcome this are quite similar ([23, 55]).

At least in the differentiable case, the advantages of the LU representation appear to be quite interesting, based on the fact that a small number of α points is in general sufficient to obtain good approximations (this is the essential gain in using the slopes to model fuzzy numbers), so reducing the number of constrained \min (49) and \max (50) problems to be solved directly. On the other hand, finding computationally efficient extension solvers is still an open research field in fuzzy calculations.

12.4.1 Constrained Fuzzy Arithmetic

A research area in fuzzy calculations attains the so-called *overestimation effect* associated with the adoption of interval arithmetic for the calculation of fuzzy arithmetic expressions. The fuzzy literature is rich in examples and a general setting has been formulated and illustrated by Klir in [16, 17]; after his paper both ‘radical’ and ‘compromise’ solutions have been proposed.

The basic question is associated with the fact that, in standard interval arithmetic, addition and multiplication do not possess inverse elements and, in particular, $u - u \neq 0$ or $u/u \neq 1$ or the fuzzy extension (48) of $f(x) = x^n$ to a fuzzy argument u is not equivalent to the product $u \cdots u \cdots u$ (n times). In this context, a fuzzy expression like $z = 3x - (y + 2x) - (u^2 + v)(v + w^2)$, for given fuzzy numbers x, y, u, v , and w , if calculated by the application of the standard interval arithmetic (*INT*), produces a fuzzy number z^{INT} with α -cuts $[z_\alpha^-, z_\alpha^+]^{INT}$ that are much larger than the fuzzy extension principle (49) and (50) applied to $z = f(x, y, u, v, w)$. In particular, the *constrained arithmetic* requires that, in the expression above, $3x - (y + 2x)$ and $(u^2 + v)(v + w^2)$ be computed with constraints induced by the double occurrence of x (as $3x$ and $2x$), of u, w (as u^2 and w^2), and of v .

The ‘radical’ solution (constrained fuzzy arithmetic, CFA) produces the extension principle (48) result: in particular, it requires that $3x - (y + 2x) = x - y$ and $(u^2 + v)(v + w^2)$ be obtained by

$$\begin{aligned} [(u^2 + v)(v + w^2)]_\alpha^- &= \min\{(a^2 + b)(b + c^2) \mid a \in [u]_\alpha; b \in [v]_\alpha; c \in [w]_\alpha\} \\ [(u^2 + v)(v + w^2)]_\alpha^+ &= \max\{(a^2 + b)(b + c^2) \mid a \in [u]_\alpha; b \in [v]_\alpha; c \in [w]_\alpha\}. \end{aligned}$$

(Denote by z^{CFA} the corresponding results.) Observe, in particular, that $(u^2)^{CFA} \neq (uu)^{INT}$.

The full adoption of the CFA produces a great increase in computational complexity, as the calculations cannot be decomposed sequentially into binary operations and all the variables have to be taken globally and the dimension may grow very quickly with the number of distinct operands.

Also a mixed (compromise) approach (see [56]) is frequently used, e.g., $z^{MIX} = (3x - (y + 2x))^{CFA} - ((u^2)^{CFA} + v)(v + (w^2)^{CFA})$, where only isolated parts of the expression are computed via CFA (e.g., $3x - (y + 2x)$ is simplified to $x - y$ and u^2 and w^2 are obtained via the unary square operator) and the other operations are executed by interval arithmetic. It is well known that, in general, $[z^{CFA}]_\alpha \subseteq [z^{MIX}]_\alpha \subseteq [z^{INT}]_\alpha$.

In a recent paper, Chang and Hung (see [57]) have proposed a series of rules to simplify the calculation of algebraic fuzzy expressions, by identifying components to be solved by the direct use of the vertex method, such as products and sums of powers, and by isolating subfunctions that operate on partitions of the total variables so as to reduce the complexity or to calculate directly according to a series of catalogued cases to simplify the applications of the vertex-like methods.

12.5 Algorithmic Fuzzy Arithmetic

In [20] and [21] we have analyzed the advantages of the LU representation in the computation of fuzzy expressions, by the direct interval arithmetic operations (INT) or by the equality-constrained fuzzy arithmetic (CFA) method of Klir.

In this section we adopt an algorithmic approach to describe the application of the fuzzy extension principle to arithmetic operators and to fuzzy function calculation associated with the LU representation of the fuzzy quantities involved. For simplicity, we will illustrate the case of differentiable representations (38); if the functions are not differentiable or if the slopes are not used (i.e., only the values u_i^- and u_i^+ are used) then in each algorithm we can omit all the blocks referring to the $\delta u_i^-, \delta u_i^+$.

For fuzzy basic operations we have easy-to-implement algorithms, based on the application of exact fuzzy operations at the nodes of the α -subdivision.¹

Algorithm 1 (LU addition, subtraction, and H-difference). Let $u = (u_i^-, \delta u_i^-, u_i^+, \delta u_i^+)_{i=0,1,\dots,N}$ and $v = (v_i^-, \delta v_i^-, v_i^+, \delta v_i^+)_{i=0,1,\dots,N}$ be given; calculate $w = u + v$, $z = u - v$, and $y = u \ominus v$ with $w = (w_i^-, \delta w_i^-, w_i^+, \delta w_i^+)_{i=0,1,\dots,N}$, $y = (y_i^-, \delta y_i^-, y_i^+, \delta y_i^+)_{i=0,1,\dots,N}$, and $z = (z_i^-, \delta z_i^-, z_i^+, \delta z_i^+)_{i=0,1,\dots,N}$.

For $i = 0, 1, \dots, N$

$$\begin{aligned} w_i^- &= u_i^- + v_i^-, z_i^- = u_i^- - v_i^+, y_i^- = u_i^- - v_i^- \\ \delta w_i^- &= \delta u_i^- + \delta v_i^-, \delta z_i^- = \delta u_i^- - \delta v_i^+, \delta y_i^- = \delta u_i^- - \delta v_i^- \\ w_i^+ &= u_i^+ + v_i^+, z_i^+ = u_i^+ - v_i^-, y_i^+ = u_i^+ - v_i^+ \\ \delta w_i^+ &= \delta u_i^+ + \delta v_i^+, \delta z_i^+ = \delta u_i^+ - \delta v_i^-, \delta y_i^+ = \delta u_i^+ - \delta v_i^+ \end{aligned}$$

end

test if conditions (38) are satisfied for $(y_i^-, \delta y_i^-, y_i^+, \delta y_i^+)_{i=0,1,\dots,N}$.

Algorithm 2 (LU scalar multiplication). Let $k \in \mathbb{R}$ and $u = (u_i^-, \delta u_i^-, u_i^+, \delta u_i^+)_{i=0,1,\dots,N}$ be given; calculate $w = ku$ with $w = (w_i^-, \delta w_i^-, w_i^+, \delta w_i^+)_{i=0,1,\dots,N}$.

For $i = 0, 1, \dots, N$

$$\begin{aligned} \text{if } k \geq 0 \text{ then } w_i^- &= ku_i^-, \delta w_i^- = k\delta u_i^-, w_i^+ = ku_i^+, \delta w_i^+ = k\delta u_i^+ \\ \text{else } w_i^- &= ku_i^+, \delta w_i^- = k\delta u_i^+, w_i^+ = ku_i^-, \delta w_i^+ = k\delta u_i^- \end{aligned}$$

end

Algorithm 3 (LU multiplication). Let $u = (u_i^-, \delta u_i^-, u_i^+, \delta u_i^+)_{i=0,1,\dots,N}$ and $v = (v_i^-, \delta v_i^-, v_i^+, \delta v_i^+)_{i=0,1,\dots,N}$ be given; calculate $w = uv$ with $w = (w_i^-, \delta w_i^-, w_i^+, \delta w_i^+)_{i=0,1,\dots,N}$.

For $i = 0, 1, \dots, N$

$$\begin{aligned} m_i &= \min\{u_i^- v_i^-, u_i^- v_i^+, u_i^+ v_i^-, u_i^+ v_i^+\} \\ M_i &= \max\{u_i^- v_i^-, u_i^- v_i^+, u_i^+ v_i^-, u_i^+ v_i^+\} \\ w_i^- &= m_i, w_i^+ = M_i \end{aligned}$$

¹ In multiplication and division with symmetric fuzzy numbers, the *min* and the *max* values of products and ratios can be attained more than once, as for $[-3, 3] * [-2, 2]$, where $\min = (-3)(2) = (3)(-2)$ and $\max = (3)(2) = (2)(3)$; in these cases, the slopes are to be calculated carefully by avoiding improper use of branches. We suggest to keep the correct branches by working with $[-3 - \varepsilon, 3 + \varepsilon] * [-2 - \varepsilon, 2 + \varepsilon]$, where ε is a very small positive number (e.g., $\varepsilon \simeq 10^{-6}$). Similarly for cases like $[a, a] * [b, b]$.

```

if  $u_i^- v_i^- = m_i$  then  $\delta w_i^- = \delta u_i^- v_i^- + u_i^- \delta v_i^-$ 
elseif  $u_i^- v_i^+ = m_i$  then  $\delta w_i^- = \delta u_i^- v_i^+ + u_i^- \delta v_i^+$ 
elseif  $u_i^+ v_i^- = m_i$  then  $\delta w_i^- = \delta u_i^+ v_i^- + u_i^+ \delta v_i^-$ 
elseif  $u_i^+ v_i^+ = m_i$  then  $\delta w_i^- = \delta u_i^+ v_i^+ + u_i^+ \delta v_i^+$ 
endif
if  $u_i^- v_i^+ = M_i$  then  $\delta w_i^+ = \delta u_i^- v_i^+ + u_i^- \delta v_i^+$ 
elseif  $u_i^- v_i^+ = M_i$  then  $\delta w_i^+ = \delta u_i^- v_i^+ + u_i^- \delta v_i^+$ 
elseif  $u_i^+ v_i^- = M_i$  then  $\delta w_i^+ = \delta u_i^+ v_i^- + u_i^+ \delta v_i^-$ 
elseif  $u_i^+ v_i^+ = M_i$  then  $\delta w_i^+ = \delta u_i^+ v_i^+ + u_i^+ \delta v_i^+$ 
endif
end

```

Similar algorithms can be deduced for the division and the scalar multiplication.

Algorithm 4 (LU division). Let $u = (u_i^-, \delta u_i^-, u_i^+, \delta u_i^+)_{i=0,1,\dots,N}$ and $v = (v_i^-, \delta v_i^-, v_i^+, \delta v_i^+)_{i=0,1,\dots,N}$ be given with $v > 0$ or $v < 0$; calculate $w = u/v$ with $w = (w_i^-, \delta w_i^-, w_i^+, \delta w_i^+)_{i=0,1,\dots,N}$.

```

For  $i = 0, 1, \dots, N$ 
 $m_i = \min\{u_i^-/v_i^-, u_i^-/v_i^+, u_i^+/v_i^-, u_i^+/v_i^+\}$ 
 $M_i = \max\{u_i^-/v_i^-, u_i^-/v_i^+, u_i^+/v_i^-, u_i^+/v_i^+\}$ 
 $w_i^- = m_i, w_i^+ = M_i$ 
if  $u_i^-/v_i^- = m_i$  then  $\delta w_i^- = (\delta u_i^- v_i^- - u_i^- \delta v_i^-)/[v_i^-]^2$ 
elseif  $u_i^-/v_i^+ = m_i$  then  $\delta w_i^- = (\delta u_i^- v_i^+ - u_i^- \delta v_i^+)/[v_i^+]^2$ 
elseif  $u_i^+/v_i^- = m_i$  then  $\delta w_i^- = (\delta u_i^+ v_i^- - u_i^+ \delta v_i^-)/[v_i^-]^2$ 
elseif  $u_i^+/v_i^+ = m_i$  then  $\delta w_i^- = (\delta u_i^+ v_i^+ - u_i^+ \delta v_i^+)/[v_i^+]^2$ 
endif
if  $u_i^-/v_i^+ = M_i$  then  $\delta w_i^+ = (\delta u_i^- v_i^+ - u_i^- \delta v_i^+)/[v_i^+]^2$ 
elseif  $u_i^-/v_i^+ = M_i$  then  $\delta w_i^+ = (\delta u_i^- v_i^+ - u_i^- \delta v_i^+)/[v_i^+]^2$ 
elseif  $u_i^+/v_i^- = M_i$  then  $\delta w_i^+ = (\delta u_i^+ v_i^- - u_i^+ \delta v_i^-)/[v_i^-]^2$ 
elseif  $u_i^+/v_i^+ = M_i$  then  $\delta w_i^+ = (\delta u_i^+ v_i^+ - u_i^+ \delta v_i^+)/[v_i^+]^2$ 
endif
end

```

If the fuzzy numbers are given in the LR form, then the (LU)–(LR) fuzzy relationship (46) can be used as an intermediate step for LR fuzzy operations.

Consider two LR fuzzy numbers u and v ($N = 1$ for simplicity)

$$\begin{aligned} u_{\text{LR}} &= (u_{0,L}, \delta u_{0,L}, u_{0,R}, \delta u_{0,R}; \quad u_{1,L}, \delta u_{1,L}, u_{1,R}, \delta u_{1,R}), \\ v_{\text{LR}} &= (v_{0,L}, \delta v_{0,L}, v_{0,R}, \delta v_{0,R}; \quad v_{1,L}, \delta v_{1,L}, v_{1,R}, \delta v_{1,R}), \end{aligned} \quad (51)$$

having the LU representations

$$\begin{aligned} u_{\text{LU}} &= (u_0^-, \delta u_0^-, u_0^+, \delta u_0^+; \quad u_1^-, \delta u_1^-, u_1^+, \delta u_1^+), \\ v_{\text{LU}} &= (v_0^-, \delta v_0^-, v_0^+, \delta v_0^+; \quad v_1^-, \delta v_1^-, v_1^+, \delta v_1^+), \end{aligned} \quad (52)$$

with $u_i^\pm, v_i^\pm, \delta u_i^\pm$, and δv_i^\pm ($i = 0, 1$) calculated according to (46).

Note that in the formulas below u and v are not constrained to have the same $L(\cdot)$ and $R(\cdot)$ shape functions and changing the slopes will change the form of the membership functions.

Addition is immediate:

$$(u + v)_{LR} = \left(u_{0,L} + u_{0,R}, \frac{\delta u_{0,L} \delta v_{0,L}}{\delta u_{0,L} + \delta v_{0,L}}, u_{0,R} + v_{0,R}, \frac{\delta u_{0,R} \delta v_{0,R}}{\delta u_{0,R} + \delta v_{0,R}}; \right. \\ \left. u_{1,L} + u_{1,R}, \frac{\delta u_{1,L} \delta v_{1,L}}{\delta u_{1,L} + \delta v_{1,L}}, u_{1,R} + v_{1,R}, \frac{\delta u_{1,R} \delta v_{1,R}}{\delta u_{1,R} + \delta v_{1,R}} \right). \quad (53)$$

The sum of u given by (43) and v given by (44) has LR representation (at $\alpha_i = 1$ use $\frac{0}{0} = 0$)

LR form of the addition of two LR fuzzy numbers

α_i	$(u + v)_{i,L}$	$\delta(u + v)_{i,L}$	$(u + v)_{i,R}$	$\delta(u + v)_{i,R}$
0.0	-7.0	0.1098	7.0	-0.1098
0.25	-5.3095	0.1816	5.3095	-0.1816
0.5	-4.0280	0.2011	4.0280	-0.2011
0.75	-2.7130	0.1706	2.7130	-0.1706
1.0	0.0	0.0	0.0	0.0

and, with respect to the exact addition, the absolute average error is 0.3%.

The multiplication $w = uv$ of two positive LR fuzzy numbers is given, in LU form, by

$$w_{LU} = (u_0^- v_0^-, \delta u_0^- v_0^- + u_0^- \delta v_0^-, u_0^+ v_0^+, \delta u_0^+ v_0^+ + u_0^+ \delta v_0^+; \\ u_1^- v_1^-, \delta u_1^- v_1^- + u_1^- \delta v_1^-, u_1^+ v_1^+, \delta u_1^+ v_1^+ + u_1^+ \delta v_1^+) \quad (54)$$

and back to the LR form of w , we obtain

$$w_{LR} = \left(u_{0,L} v_{0,L}, \frac{\delta u_{0,L} \delta v_{0,L}}{\delta u_{0,L} + u_{0,L} \delta u_{0,L}}, u_{0,R} v_{0,R}, \frac{\delta u_{0,R} \delta v_{0,R}}{\delta u_{0,R} + u_{0,R} \delta u_{0,R}}; \right. \\ \left. u_{1,L} v_{1,L}, \frac{\delta u_{1,L} \delta v_{1,L}}{\delta u_{1,L} + u_{1,L} \delta u_{1,L}}, u_{1,R} v_{1,R}, \frac{\delta u_{1,R} \delta v_{1,R}}{\delta u_{1,R} + u_{1,R} \delta u_{1,R}} \right).$$

The corresponding algorithm is immediate.

Algorithm 5 (LR multiplication). Let $u = (u_{i,L}, \delta u_{i,L}, u_{i,R}, \delta u_{i,R})_{i=0,1,\dots,N}$ and $v = (v_{i,L}, \delta v_{i,L}, v_{i,R}, \delta v_{i,R})_{i=0,1,\dots,N}$ be given LR fuzzy numbers in parametric form; calculate $w = uv$ with $w = (w_{i,L}, \delta w_{i,L}, w_{i,R}, \delta w_{i,R})_{i=0,1,\dots,N}$. (If necessary, set $\frac{0}{0} = 0$.)

For $i = 0, 1, \dots, N$

```

 $m_i = \min\{u_{i,L} v_{i,L}, u_{i,L} v_{i,R}, u_{i,R} v_{i,L}, u_{i,R} v_{i,R}\}$ 
 $M_i = \max\{u_{i,L} v_{i,L}, u_{i,L} v_{i,R}, u_{i,R} v_{i,L}, u_{i,R} v_{i,R}\}$ 
 $w_{i,L} = m_i, w_{i,R} = M_i$ 
if  $u_{i,L} v_{i,L} = m_i$  then  $\delta w_{i,L} = \delta u_{i,L} \delta v_{i,L} / [v_{i,L} \delta v_{i,L} + u_{i,L} \delta u_{i,L}]$ 
elseif  $u_{i,L} v_{i,R} = m_i$  then  $\delta w_{i,L} = \delta u_{i,L} \delta v_{i,R} / [v_{i,R} \delta v_{i,R} + u_{i,L} \delta u_{i,L}]$ 
elseif  $u_{i,R} v_{i,L} = m_i$  then  $\delta w_{i,L} = \delta u_{i,R} \delta v_{i,L} / [v_{i,L} \delta v_{i,L} + u_{i,R} \delta u_{i,R}]$ 
elseif  $u_{i,R} v_{i,R} = m_i$  then  $\delta w_{i,L} = \delta u_{i,R} \delta v_{i,R} / [v_{i,R} \delta v_{i,R} + u_{i,R} \delta u_{i,R}]$ 
endif

```

```

if  $u_{i,L}v_{i,L} = M_i$  then  $\delta w_{i,R} = \delta u_{i,L}\delta v_{i,L}/[v_{i,L}\delta v_{i,L} + u_{i,L}\delta u_{i,L}]$ 
elseif  $u_{i,L}v_{i,R} = M_i$  then  $\delta w_{i,R} = \delta u_{i,L}\delta v_{i,R}/[v_{i,R}\delta v_{i,R} + u_{i,L}\delta u_{i,L}]$ 
elseif  $u_{i,R}v_{i,L} = M_i$  then  $\delta w_{i,R} = \delta u_{i,R}\delta v_{i,L}/[v_{i,L}\delta v_{i,L} + u_{i,R}\delta u_{i,R}]$ 
elseif  $u_{i,R}v_{i,R} = M_i$  then  $\delta w_{i,R} = \delta u_{i,R}\delta v_{i,R}/[v_{i,R}\delta v_{i,R} + u_{i,R}\delta u_{i,R}]$ 
endif
end

```

As pointed out by the experimentation reported in [20] and [21] the operations above are exact at the nodes α_i and have very small global errors on $[0, 1]$. Further, it is easy to control the error by using a sufficiently fine α -decomposition and the results have shown that both the rational (30) and the mixed (31) models perform well.

Some parametric membership functions in the LR framework are present in many applications and the use of non-linear shapes is increasing. Usually, one defines a given family, e.g., linear, quadratic (see the extended study on piecewise parabolic functions in [58]), sigmoid, or quasi-Gaussian, and the operations are performed within the same family.

Our proposed parametrization (linking directly LR and LU representations) allows an extended set of flexible fuzzy numbers and is able to approximate all other forms with acceptable small errors, with the additional advantage of producing good approximations to the results of the arithmetic operations even between LU or LR fuzzy numbers having very different original shapes.

12.6 Computation of Fuzzy-Valued Functions

Let $v = f(u_1, u_2, \dots, u_n)$ denote the fuzzy extension of a continuous function f in n variables; it is well known that the fuzzy extension of f to normal upper-semicontinuous fuzzy intervals (with compact support) has the level-cutting commutative property (see [13]); i.e. the α -cuts $[v_\alpha^-, v_\alpha^+]$ of v are the images of the α -cuts of (u_1, u_2, \dots, u_n) and are obtained by solving the box-constrained optimization problems

$$(EP)_\alpha : \begin{cases} v_\alpha^- = \min\{f(x_1, x_2, \dots, x_n) \mid x_k \in [u_{k,\alpha}^-, u_{k,\alpha}^+], k = 1, 2, \dots, n\} \\ v_\alpha^+ = \max\{f(x_1, x_2, \dots, x_n) \mid x_k \in [u_{k,\alpha}^-, u_{k,\alpha}^+], k = 1, 2, \dots, n\}. \end{cases} \quad (54)$$

Except for simple elementary cases for which the optimization problems above can be solved analytically, the direct application of (EP) is difficult and computationally expensive.

Basically, the vertex method evaluates the objective function at the 2^n vertices of the hyper rectangular box

$$U_\alpha = [u_{1,\alpha}^-, u_{1,\alpha}^+] \times [u_{2,\alpha}^-, u_{2,\alpha}^+] \times \cdots \times [u_{n,\alpha}^-, u_{n,\alpha}^+]$$

and modifications have been proposed to take into account eventual internal or boundary optima (see [47, 59]) or to extend both a function and its inverse (see [60]).

The transformation method, in its general or reduced or extended versions (see [51] for a recent efficient implementation), evaluates the objective function in a sufficiently large number of points in a hierarchical selection of α -cuts $U_{i/m}$, with $\alpha_i = i/m$ for $i = m, m-1, \dots, 1, 0$ (including the vertices, midpoints of vertices, midpoints of midpoints, ...) and estimates the $m+1$ α -cuts of the solution $[v]_{i/m} = [v_i^-, v_i^+]$ by choosing recursively the best (*min* and *max*) values for each $i = m, m-1, \dots, 0$.

Recently, a sparse grids method for interval arithmetic optimization has been proposed (see [52]) to further improve the computational efficiency for general functions; the method starts with a hierarchical selection of α -cuts $U_{i/m}$ and constructs a linear (or polynomial) interpolation of the objective function $f(\cdot)$ over a grid of points (internal and on the boundary) which is sufficiently sparse (a strong selection of the possible points) and has optimal ‘covering’ properties. The optimizations are then performed by finding the *min* and the *max* of the interpolant $A_f(\cdot)$ (in general simpler than the original function) by using adapted (global) search procedures.

In the following subsections, we give the details of the fuzzy extension of general (piecewise) differentiable functions by the LU representation. In all the computations we will adopt the EP method, but also if other approaches are adopted, the representation still remains valid.

12.6.1 Univariate Functions

We consider first a single-variable differentiable function $f : \mathbb{R} \rightarrow \mathbb{R}$; its (EP)-extension $v = f(u)$ to a fuzzy argument $u = (u^-, u^+)$ has α -cuts

$$[v]_\alpha = [\min\{f(x) | x \in [u]_\alpha\}, \max\{f(x) | x \in [u]_\alpha\}]. \quad (56)$$

If f is monotonic increasing, we obtain $[v]_\alpha = [f(u_\alpha^-), f(u_\alpha^+)]$, while if f is monotonic decreasing, $[v]_\alpha = [f(u_\alpha^+), f(u_\alpha^-)]$; the LU representation of $v = (v_i^-, \delta v_i^-, v_i^+, \delta v_i^+)_{i=0,1,\dots,N}$ is obtained by the following.

Algorithm 6 (1-dim monotonic extension). Let $u = (u_i^-, \delta u_i^-, u_i^+, \delta u_i^+)_{i=0,1,\dots,N}$ be given and $f : \text{supp}(u) \rightarrow \mathbb{R}$ be differentiable monotonic; calculate $v = f(u)$ with $v = (v_i^-, \delta v_i^-, v_i^+, \delta v_i^+)_{i=0,1,\dots,N}$.

For $i = 0, 1, \dots, N$

if (f is increasing) **then**

$$v_i^- = f(u_i^-), \delta v_i^- = f'(u_i^-)\delta u_i^-$$

$$v_i^+ = f(u_i^+), \delta v_i^+ = f'(u_i^+)\delta u_i^+$$

else

$$v_i^- = f(u_i^+), \delta v_i^- = f'(u_i^+)\delta u_i^+$$

$$v_i^+ = f(u_i^-), \delta v_i^+ = f'(u_i^-)\delta u_i^-$$

endif

end

As an example, the monotonic exponential function $f(x) = \exp(x)$ has LU fuzzy extension

$$\exp(u) = (\exp(u_i^-), \exp(u_i^-)\delta u_i^-, \exp(u_i^+), \exp(u_i^+)\delta u_i^+)_{i=0,1,\dots,N}.$$

In the non-monotonic (differentiable) case, we have to solve the optimization problems in (56) for each $\alpha = \alpha_i$, $i = 0, 1, \dots, N$; i.e.,

$$(\text{EP}_i): \begin{cases} v_i^- = \min\{f(x) | x \in [u_i^-, u_i^+]\} \\ v_i^+ = \max\{f(x) | x \in [u_i^-, u_i^+]\}. \end{cases}$$

The min (or the max) can occur either at a point which is coincident with one of the extremal values of $[u_i^-, u_i^+]$ or at a point which is internal; in the last case, the derivative of f is null and $\delta v_i^- = 0$ (or $\delta v_i^+ = 0$).

Algorithm 7 (1-dim non-monotonic extension). Let $u = (u_i^-, \delta u_i^-, u_i^+, \delta u_i^+)_{i=0,1,\dots,N}$ be given and $f : \text{supp}(u) \rightarrow \mathbb{R}$ be differentiable; calculate $v = f(u)$ with $v = (v_i^-, \delta v_i^-, v_i^+, \delta v_i^+)_{i=0,1,\dots,N}$.

For $i = 0, 1, \dots, N$

solve $\min\{f(x) | x \in [u_i^-, u_i^+]\}$, **let** $\hat{x}_i^- = \arg \min\{f(x) | x \in [u_i^-, u_i^+]\}$

if $\hat{x}_i^- = u_i^-$ **then** $v_i^- = f(u_i^-), \delta v_i^- = f'(u_i^-)\delta u_i^-$

elseif $\hat{x}_i^- = u_i^+$ **then** $v_i^- = f(u_i^+), \delta v_i^- = f'(u_i^+)\delta u_i^+$

else $v_i^- = f(\hat{x}_i^-), \delta v_i^- = 0$

endif

```

solve max{ $f(x) | x \in [u_i^-, u_i^+]$ }, let  $\hat{x}_i^+ = \arg \max\{f(x) | x \in [u_i^-, u_i^+]\}$ 
if  $\hat{x}_i^+ = u_i^-$  then  $v_i^+ = f(u_i^-)$ ,  $\delta v_i^+ = f'(u_i^-)\delta u_i^-$ 
elseif  $\hat{x}_i^+ = u_i^+$  then  $v_i^+ = f(u_i^+)$ ,  $\delta v_i^+ = f'(u_i^+)\delta u_i^+$ 
else  $v_i^+ = f(\hat{x}_i^+)$ ,  $\delta v_i^+ = 0$ 
endif
end

```

As an example of unidimensional non-monotonic function, consider the hyperbolic cosinusoidal function $y = \cosh(x) = \frac{e^x + e^{-x}}{2}$. Its fuzzy extension to u can be obtained as follows:

Example 1. Calculation of fuzzy $v = \cosh(u)$.

For $i = 0, 1, \dots, N$

if $u_i^+ \leq 0$ **then**

$$\begin{aligned} v_i^- &= \cosh(u_i^+), v_i^+ = \cosh(u_i^-) \\ \delta v_i^- &= \delta u_i^+ \sinh(u_i^+), \delta v_i^+ = \delta u_i^- \sinh(u_i^-) \\ \text{elseif } u_i^- \geq 0 \text{ then} \\ v_i^- &= \cosh(u_i^-), v_i^+ = \cosh(u_i^+) \\ \delta v_i^- &= \delta u_i^- \sinh(u_i^-), \delta v_i^+ = \delta u_i^+ \sinh(u_i^+) \end{aligned}$$

else

$$v_i^- = 1, \delta v_i^- = 0$$

if $\text{abs}(u_i^-) \geq \text{abs}(u_i^+)$

$$\begin{aligned} \text{then } v_i^+ &= \cosh(u_i^-), \delta v_i^+ = \delta u_i^- \sinh(u_i^-) \\ \text{else } v_i^+ &= \cosh(u_i^+), \delta v_i^+ = \delta u_i^+ \sinh(u_i^+) \end{aligned}$$

endif

endif

end

The fuzzy extension of elementary functions by the LU fuzzy representation are documented in [44] and an application to fuzzy dynamical systems is in [61].

12.6.2 Multivariate Functions

Consider now the extension of a multivariate differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ to a vector of n fuzzy numbers $u = (u_1, u_2, \dots, u_n)$ with k th component

$$u_k = (u_{k,i}^-, \delta u_{k,i}^-, u_{k,i}^+, \delta u_{k,i}^+)_{i=0,1,\dots,N} \text{ for } k = 1, 2, \dots, n.$$

Let $v = f(u_1, u_2, \dots, u_n)$ and $v = (v_i^-, \delta v_i^-, v_i^+, \delta v_i^+)_{i=0,1,\dots,N}$ be its LU representation; the α -cuts of v are obtained by solving the box-constrained optimization problems (EP).

For each $\alpha = \alpha_i, i = 0, 1, \dots, N$, the min and the max (EP) can occur either at a point whose components $x_{k,i}$ are internal to the corresponding intervals $[u_{k,i}^-, u_{k,i}^+]$ or are coincident with one of the extremal values; denote by $\hat{x}_i^- = (\hat{x}_{1,i}^-, \dots, \hat{x}_{n,i}^-)$ and $\hat{x}_i^+ = (\hat{x}_{1,i}^+, \dots, \hat{x}_{n,i}^+)$ the points where the min and the max take place; then

$$v_i^- = f(\hat{x}_{1,i}^-, \hat{x}_{2,i}^-, \dots, \hat{x}_{n,i}^-) \text{ and } v_i^+ = f(\hat{x}_{1,i}^+, \hat{x}_{2,i}^+, \dots, \hat{x}_{n,i}^+),$$

and the slopes δv_i^- and δv_i^+ are computed (as f is differentiable) by

$$\begin{aligned} \delta v_i^- &= \sum_{k=1}^n \frac{\partial f(\widehat{x}_{1,i}^-, \dots, \widehat{x}_{n,i}^-)}{\partial x_k} \delta u_{k,i}^- + \sum_{\substack{k=1 \\ \widehat{x}_{k,i}^- = u_{k,i}^-}}^n \frac{\partial f(\widehat{x}_{1,i}^-, \dots, \widehat{x}_{n,i}^-)}{\partial x_k} \delta u_{k,i}^+ \\ \delta v_i^+ &= \sum_{k=1}^n \frac{\partial f(\widehat{x}_{1,i}^+, \dots, \widehat{x}_{n,i}^+)}{\partial x_k} \delta u_{k,i}^+ + \sum_{\substack{k=1 \\ \widehat{x}_{k,i}^+ = u_{k,i}^+}}^n \frac{\partial f(\widehat{x}_{1,i}^+, \dots, \widehat{x}_{n,i}^+)}{\partial x_k} \delta u_{k,i}^+ \end{aligned} \quad (57)$$

If, for some reasons, the partial derivatives of f at the solution points are non-available (and the points are not internal), we can always produce an estimation of the shapes δv_i^- and δv_i^+ ; it is sufficient to calculate $v_{\Delta i}^-$ and $v_{\Delta i}^+$ corresponding to $\alpha = \alpha_i \pm \Delta\alpha$ ($\Delta\alpha$ small) and estimate the slopes by applying a least squares criterion like (32).

Algorithm 8 (n-dim extension). Let $u_k = (u_{k,i}^-, \delta u_{k,i}^-, u_{k,i}^+, \delta u_{k,i}^+)_{i=0,1,\dots,N}$ $k = 1, 2, \dots, n$ be given and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be differentiable; with $v = (v_i^-, \delta v_i^-, v_i^+, \delta v_i^+)_{i=0,1,\dots,N}$, calculate $v = f(u_1, \dots, u_n)$.

For $i = 0, 1, \dots, N$

```

solve  $\min\{f(x_1, \dots, x_n) | x_k \in [u_{k,i}^-, u_{k,i}^+] \forall k\}$ 
let  $(\widehat{x}_{1,i}^-, \dots, \widehat{x}_{n,i}^-) = \arg \min\{f(x_1, \dots, x_n) | x_k \in [u_{k,i}^-, u_{k,i}^+], \forall k\}$ 
let  $v_i^- = f(\widehat{x}_{1,i}^-, \dots, \widehat{x}_{n,i}^-)$ ,  $\delta v_i^- = 0$ 
for  $k = 1, 2, \dots, n$  (loop to calculate  $\delta v_i^-$ )
  if  $\widehat{x}_{k,i}^- = u_{k,i}^-$  then  $\delta v_i^- = \delta v_i^- + \frac{\partial f(\widehat{x}_{1,i}^-, \dots, \widehat{x}_{n,i}^-)}{\partial x_k} \delta u_{k,i}^-$ 
  elseif  $\widehat{x}_{k,i}^- = u_{k,i}^+$  then  $\delta v_i^- = \delta v_i^- + \frac{\partial f(\widehat{x}_{1,i}^-, \dots, \widehat{x}_{n,i}^-)}{\partial x_k} \delta u_{k,i}^+$ 
end
solve  $\max\{f(x_1, \dots, x_n) | x_k \in [u_{k,i}^-, u_{k,i}^+] \forall k\}$ 
let  $(\widehat{x}_{1,i}^+, \dots, \widehat{x}_{n,i}^+) = \arg \max\{f(x_1, \dots, x_n) | x_k \in [u_{k,i}^-, u_{k,i}^+], \forall k\}$ 
let  $v_i^+ = f(\widehat{x}_{1,i}^+, \dots, \widehat{x}_{n,i}^+)$ ,  $\delta v_i^+ = 0$ 
for  $k = 1, 2, \dots, n$  (loop to calculate  $\delta v_i^+$ )
  if  $\widehat{x}_{k,i}^+ = u_{k,i}^-$  then  $\delta v_i^+ = \delta v_i^+ + \frac{\partial f(\widehat{x}_{1,i}^+, \dots, \widehat{x}_{n,i}^+)}{\partial x_k} \delta u_{k,i}^-$ 
  elseif  $\widehat{x}_{k,i}^+ = u_{k,i}^+$  then  $\delta v_i^+ = \delta v_i^+ + \frac{\partial f(\widehat{x}_{1,i}^+, \dots, \widehat{x}_{n,i}^+)}{\partial x_k} \delta u_{k,i}^+$ 
end
end

```

The main and possibly critical steps in the above algorithm are the solution of the optimization problems (EP), depending on the dimension n of the solution space and on the possibility of many local optimal points. (If the min and the max points are not located with sufficient precision, an underestimation of the fuzziness may be produced and the propagation of the errors may grow without control.)

In many applications, a careful exploitation of the min and max problems can produce efficient solution methods. An example is offered in [62] and [63] for the fuzzy finite element analysis; the authors analyze the essential sources of uncertainty and define the correct form of the objective function (so avoiding unnecessary overestimation); then they propose a safe approximation of the objective function by a quadratic interpolation scheme and use a version of the corner method to determine the optimal solutions.

As we have mentioned, all existing general methods (in cases where the structure of the min and max subproblems do not suggest specific efficient procedures) try to take advantage of the nested structure of the box constraints for different values of α .

We suggest here a relatively simple procedure, based on the differential evolution (DE) method of Storn and Price (see [64–66]) and adapted to take into account both the nested property of α -cuts and the min and max problems over the same domains.

The general idea of DE to find *min* or *max* of $\{f(x_1, \dots, x_n) | (x_1, \dots, x_n) \in \mathbb{V} \subset \mathbb{R}^n\}$ is simple. Start with an initial ‘population’ $(x_1, \dots, x_n)^{(1)}, \dots, (x_1, \dots, x_n)^{(p)} \in \mathbb{V}$ of p feasible points; at each iteration obtain a new set of points by recombining randomly the individuals of the current population and by selecting the best generated elements to continue in the next generation. A typical recombination operates on a single component $j \in \{1, \dots, n\}$ and has the form (see [65, 67, 68])

$$x'_j = x_j^{(r)} + \gamma \left[x_j^{(s)} - x_j^{(t)} \right], \gamma \in]0, 1],$$

where $r, s, t \in \{1, 2, \dots, p\}$ are chosen randomly. The components of each individual of the current population are modified to x'_j by a given probability q . Typical values are $\gamma \in [0.2, 0.95]$ and $q \in [0.7, 1.0]$.

To take into account the particular mentioned nature of the problem, we modify the basic procedure: start with the ($\alpha = 1$)-cut back to the ($\alpha = 0$)-cut so that the optimal solutions at a given level can be inserted into the ‘starting’ populations of lower levels; use two distinct populations and perform the recombinations such that, during generations, one of the populations specializes to find the minimum and the other to find the maximum.

Algorithm 9 (DE procedure). Let $[u_{k,i}^-, u_{k,i}^+]$, $k = 1, 2, \dots, n$, and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be given; find, for $i = 0, 1, \dots, N$,

$$v_i^- = \min\{f(x_1, \dots, x_n) | x_k \in [u_{k,i}^-, u_{k,i}^+] \forall k\} \text{ and}$$

$$v_i^+ = \max\{f(x_1, \dots, x_n) | x_k \in [u_{k,i}^-, u_{k,i}^+] \forall k\}$$

Choose $p \approx 10n$, $g_{\max} \approx 200$, q , and γ . Function $\text{rand}(0, 1)$ generates a random uniform number between 0 and 1.

Select $(x_1^{(l)}, \dots, x_n^{(l)})$, $x_k^{(l)} \in [u_{k,N}^-, u_{k,N}^+] \forall k, l = 1, \dots, 2p$

let $y^{(l)} = f(x_1^{(l)}, \dots, x_n^{(l)})$

for $i = N, N - 1, \dots, 0$

for $g = 1, 2, \dots, g_{\max}$ (up to g_{\max} generations or other stopping rule)

for $l = 1, 2, \dots, 2p$

select (randomly) $r, s, t \in \{1, 2, \dots, 2p\}$ and $j^* \in \{1, 2, \dots, n\}$

for $j = 1, 2, \dots, n$

if $(j = j^* \text{ or } \text{random}(0, 1) < q)$

then $x'_j = x_j^{(r)} + \gamma[x_j^{(s)} - x_j^{(t)}]$

else $x'_j = x_j^{(l)}$

endif

if $(x'_j < u_{j,i}^-)$ **then** $x'_j = u_{j,i}^-$ (lower feasibility)

if $(x'_j > u_{j,i}^+)$ **then** $x'_j = u_{j,i}^+$ (upper feasibility)

end

let $y = f(x'_1, \dots, x'_n)$

if $l \leq p$ **and** $y < y^{(l)}$

then substitute $(x_1, \dots, x_n)^{(l)}$ with (x'_1, \dots, x'_n) (best min)

endif

```

if  $l > p$  and  $y > y^{(l)}$ 
    then substitute  $(x_1, \dots, x_n)^{(l)}$  with  $(x'_1, \dots, x'_n)$  (best max)
endif
end
end

 $v_i^- = y^{(l^*)} = \min\{y^{(l)} \mid l = 1, 2, \dots, p\}, (\hat{x}_{1,i}^-, \dots, \hat{x}_{n,i}^-) = (x_1, \dots, x_n)^{(l^*)}$ 
 $v_i^+ = y^{(l^{**})} = \max\{y^{(p+l)} \mid l = 1, 2, \dots, p\}, (\hat{x}_{1,i}^+, \dots, \hat{x}_{n,i}^+) = (x_1, \dots, x_n)^{(l^{**})}$ 

if  $i < N$ 
    select  $(x_1^{(l)}, \dots, x_n^{(l)}), x_k^{(l)} \in [u_{k,i-1}^-, u_{k,i-1}^+] \forall k, l = 1, \dots, 2p$ 
    including  $(\hat{x}_{1,i}^-, \dots, \hat{x}_{n,i}^-)$  and  $(\hat{x}_{1,i}^+, \dots, \hat{x}_{n,i}^+)$ ;
    eventually reduce  $g_{\max}$ .
endif
end

```

Extended experiments of the DE procedure (and some variants) are documented in [67], where two algorithms SPDE (single population) and MPDE (multiple populations) have been implemented and executed on a set of 35 test functions with different dimension $n = 2, 4, 8, 16, 32$.

If the extension algorithm is used in combinations with the LU fuzzy representation for differentiable membership functions (and differentiable extended functions), then the number $N + 1$ of α -cuts (and correspondingly of min/max optimizations) can be sufficiently small. Experiments in [20] and [21] motivated that $N = 10$ is in general quite sufficient to obtain good approximations. The number of function evaluations FE_{SPDE} and FE_{MPDE} needed to the two algorithms SPDE and MPDE to reach the solution of the nested min/max optimization problems corresponding to the 11 α -cuts of the uniform α -decomposition $\alpha_i = \frac{i}{10}, i = 0, 1, \dots, 10$ ($N = 10$ subintervals), is reported in Figure 12.6.

The graph (Figure 12.6) represents the logarithm of the number of function evaluations FE vs. the logarithm of the number n of arguments $\ln(FE_{\text{SPDE}}) = a + b \ln(n)$ and $\ln(FE_{\text{MPDE}}) = c + d \ln(n)$. The estimated coefficients are $a = 8.615$, $b = 1.20$ and $c = 7.869$, $d = 1.34$. The computational complexity of the DE algorithm (on average for the 35 test problems) grows less than quadratically ($FE_{\text{SPDE}} \approx 5513.8n^{1.2}$ and $FE_{\text{MPDE}} \approx 2614.9n^{1.34}$) with the dimension n . (SPDE is less efficient but grows slowly than MPDE.) This is an interesting result, as all the existing methods for the fuzzy extension of functions are essentially exponential in n .

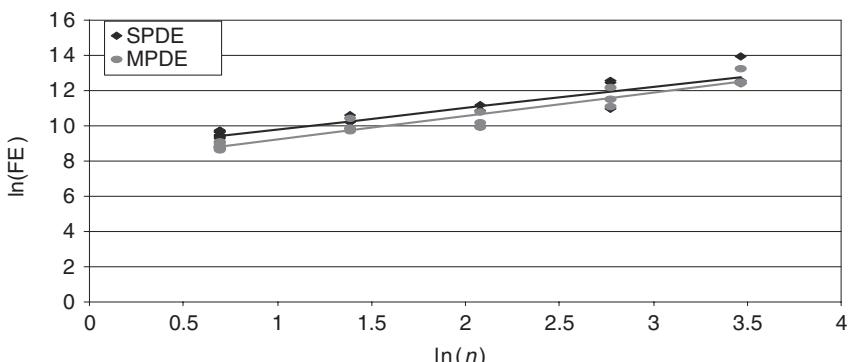


Figure 12.6 Number of function evaluations FE vs. number of variables n for two versions of DE algorithm for fuzzy extension of functions ($n = 2, 4, 8, 16, 32$)

12.7 Integration and Differentiation of Fuzzy-Valued Functions

Integrals and derivatives of fuzzy-valued functions have been established, among others, by Dubois and Prade [6], Kaleva ([69, 70]), and Puri and Ralescu [71]; see also [43] and [72] for recent results.

We consider here a fuzzy-valued function $u : [a, b] \rightarrow \mathbb{F}_{\mathbb{I}}$, where $u(t) = (u^-(t), u^+(t))$ for $t \in [a, b]$ is an LU fuzzy number of the form

$$u(t) = (u_i^-(t), \delta u_i^-(t), u_i^+(t), \delta u_i^+(t))_{i=0,1,\dots,N}.$$

The integral of $u(t)$ with respect to $t \in [a, b]$ is given by

$$[v]_{\alpha} := \left[\int_a^b u(t) dt \right]_{\alpha} = \left[\int_a^b u_{\alpha}^-(t) dt, \int_a^b u_{\alpha}^+(t) dt \right], \quad \alpha \in [0, 1] \quad (58)$$

and its LU representation $v = (v_i^-, \delta v_i^-, v_i^+, \delta v_i^+)_i=0,1,\dots,N$ is simply

$$v_i^{\pm} = \int_a^b u_i^{\pm}(t) dt \quad \text{and} \quad \delta v_i^{\pm} = \int_a^b \delta u_i^{\pm}(t) dt, \quad i = 0, 1, \dots, N. \quad (59)$$

The H-derivative [71] (and the generalized derivative [43]) of $u(t)$ at a point t_0 is obtained by considering the intervals defined by the derivatives of the lower and upper branches of the α -cuts

$$[u'(t_0)]_{\alpha} = \left[\frac{d}{dt} u_{\alpha}^-(t) \Big|_{t=t_0}, \frac{d}{dt} u_{\alpha}^+(t) \Big|_{t=t_0} \right] \quad (60)$$

$$\text{or } [u'(t_0)]_{\alpha} = \left[\frac{d}{dt} u_{\alpha}^+(t) \Big|_{t=t_0}, \frac{d}{dt} u_{\alpha}^-(t) \Big|_{t=t_0} \right], \quad (61)$$

provided that the intervals define a correct fuzzy number for each t_0 . Using the LU fuzzy representation, we obtain, in the first case (the ' means derivative w.r.t. t)

$$u'(t) = (u_i'^-(t), \delta u_i'^-(t), u_i'^+(t), \delta u_i'^+(t))_{i=0,1,\dots,N} \quad (62)$$

and the conditions for a valid fuzzy derivative are, for $i = 0, 1, \dots, N$,

$$\begin{aligned} u_0'^-(t) &\leq u_1'^-(t) \leq \dots \leq u_N'^-(t) \leq \\ &\leq u_N'^+(t) \leq u_{N-1}'^+(t) \leq \dots \leq u_0'^+(t) \\ \delta u_i'^-(t) &\geq 0, \\ \delta u_i'^+(t) &\leq 0. \end{aligned} \quad (63)$$

As an example, consider the fuzzy-valued function $[u(t)]_{\alpha} = [u_{\alpha}^-(t), u_{\alpha}^+(t)]$, $t \in [0, 2\pi]$, with

$$\begin{aligned} u_{\alpha}^-(t) &= \frac{t^2}{40} + \frac{(3\alpha^2 - 2\alpha^3) \sin^2(t)}{20} \\ u_{\alpha}^+(t) &= \frac{t^2}{40} + \frac{(2 - 3\alpha^2 + 2\alpha^3) \sin^2(t)}{20}. \end{aligned}$$

At $t \in \{0, \pi, 2\pi\}$ the function $u(t)$ has a crisp value.

The generalized fuzzy derivative exists at all the points of $]0, 2\pi[$ and is obtained by (60) for $t \in]0, \frac{1}{2}\pi] \cup [\pi, \frac{3}{2}\pi]$, by (61) for $t \in [\frac{1}{2}\pi, \pi] \cup [\frac{3}{2}\pi, 2\pi[$, and by both for $t \in \{\frac{1}{2}\pi, \pi, \frac{3}{2}\pi\}$. Note that at the points $t \in \{\frac{1}{2}\pi, \pi, \frac{3}{2}\pi\}$ the derivatives $\frac{d}{dt}u_\alpha^-(t)$ and $\frac{d}{dt}u_\alpha^+(t)$ change the relative position in defining the lower and the upper branches of the generalized fuzzy derivative, given for $t \in]0, \frac{1}{2}\pi] \cup [\pi, \frac{3}{2}\pi]$ by

$$\begin{aligned}[u'(t)]_\alpha^- &= \left(\frac{t}{2} + (3\alpha^2 - 2\alpha^3) \sin(t) \cos(t) \right) / 10 \\ [u'(t)]_\alpha^+ &= \left(\frac{t}{2} + (2 - 3\alpha^2 + 2\alpha^3) \sin(t) \cos(t) \right) / 10,\end{aligned}$$

and for $t \in [\frac{1}{2}\pi, \pi] \cup [\frac{3}{2}\pi, 2\pi[$ by

$$\begin{aligned}[u'(t)]_\alpha^- &= \left(\frac{t}{2} + (2 - 3\alpha^2 + 2\alpha^3) \sin(t) \cos(t) \right) / 10 \\ [u'(t)]_\alpha^+ &= \left(\frac{t}{2} + (3\alpha^2 - 2\alpha^3) \sin(t) \cos(t) \right) / 10.\end{aligned}$$

Observe that the two cases (60) and (61) can be formulated in a compact form by using the generalized H-difference to the incremental ratio, so defining $m_\alpha(t) = \min\{(u_\alpha^-)'(t), (u_\alpha^+)'(t)\}$ and $M_\alpha(t) = \max\{(u_\alpha^-)'(t), (u_\alpha^+)'(t)\}$ and $[u'(t)]_\alpha = [m_\alpha(t), M_\alpha(t)]$, $\alpha \in [0, 1]$, provided that we have a fuzzy (or a crisp) number; i.e.,

$$[u'(t)]_\alpha = \lim_{\Delta t \rightarrow 0} \left[\frac{[u(t + \Delta t)]_\alpha \ominus_g [u(t)]_\alpha}{\Delta t} \right], \quad (64)$$

provided that $[u(t + \Delta t)]_\alpha \ominus_g [u(t)]_\alpha$ and the limit exist.

To obtain the LU fuzzy parametrization of the generalized fuzzy derivative, we need to choose a decomposition of the membership interval $[0, 1]$ into N subintervals and define the values and the slopes as in (62). For simplicity of notation, we consider the trivial decomposition with only two points $0 = \alpha_0 < \alpha_1 = 1$ (i.e., $N = 1$) so that the parametrization of $u(t)$ is, for a given t ,

$$\begin{aligned}u(t) &= (u_\alpha^-(t), \delta u_\alpha^-(t), u_\alpha^+(t), \delta u_\alpha^+(t); u_1^-(t), \delta u_1^-(t), u_1^+(t), \delta u_1^+(t)) \\ &= \left(\frac{t^2}{40}, 0, \frac{t^2}{40} + \frac{1}{10} \sin^2(t), 0; \frac{t^2}{40} + \frac{1}{20} \sin^2(t), 0, \frac{t^2}{40} + \frac{1}{20} \sin^2(t), 0 \right).\end{aligned}$$

The slopes $\delta u_i^-(t)$ and $\delta u_i^+(t)$ are the derivatives of $u_\alpha^-(t)$ and $u_\alpha^+(t)$ with respect to α at $\alpha = 0$ and $\alpha = 1$, and they are null for any t . By applying the cases (60) and (61) the values of the generalized fuzzy derivative of u at point t (the derivatives $(\cdot)'$ are intended with respect to t) are given by the correct combinations of $(u_0^-)'(t) = \frac{t}{20}$, $(u_1^-)'(t) = \frac{t}{20} + \frac{1}{10} \sin(t) \cos(t)$, $(u_0^+)'(t) = \frac{t}{20} + \frac{1}{10} \sin(t) \cos(t)$, and $(u_1^+)'(t) = \frac{t}{20} + \frac{1}{5} \sin(t) \cos(t)$, i.e., depending on the sign of $\sin(t) \cos(t)$; it has the following LU fuzzy parametrization:

Generalized derivative in case (60): $t \in]0, \frac{1}{2}\pi] \cup [\pi, \frac{3}{2}\pi]$

$$u'(t) = \left(\frac{t}{20}, 0, \frac{t}{20} + \frac{1}{5} \sin(t) \cos(t), 0; \frac{t}{20} + \frac{1}{10} \sin(t) \cos(t), 0, \frac{t}{20} + \frac{1}{10} \sin(t) \cos(t), 0 \right).$$

Generalized derivative in case (61): $t \in [\frac{1}{2}\pi, \pi] \cup [\frac{3}{2}\pi, 2\pi[$

$$u'(t) = \left(\frac{t}{20} + \frac{1}{5} \sin(t) \cos(t), 0, \frac{t}{20}, 0; \frac{t}{20} + \frac{1}{10} \sin(t) \cos(t), 0, \frac{t}{20} + \frac{1}{10} \sin(t) \cos(t), 0 \right).$$

12.8 Applications and Concluding Remark

The characterization of uncertainty by fuzzy sets is an issue to manage information in modeling reasoning, processes, systems design and analysis, and decision making in many fields of application. In particular, as we have seen, fuzzy numbers (having the real numbers as the basic support) are special fuzzy sets whose definition can be based on intervals and can be parametrized to obtain a flexible representation modeling and for calculations.

A brief indication of major trends in the applications of fuzzy numbers (intervals) is described in the concluding section of [13], which contains the essential points and references. We follow an analogous frame to sketch the very recent essential literature where fuzzy numbers and fuzzy arithmetic are used for some applications (which are near to the research interests of the authors).²

In the field of *fuzzy analysis* (see [73] for the basic concepts and [8, 31] for a formal treatment), the very recent literature has dedicated great attention to various areas involving fuzzy numbers and corresponding arithmetic aspects. Two problems have been particularly addressed: (i) for the solution of fuzzy algebraic equations, in particular having the matrix form $Au = v$ with some or all elements being fuzzy (A , v , and/or u) (see [74–78]); (ii) the analysis and numerical treatment of fuzzy integral (see [79]) and differential equations are a second series of arguments addressed by a great research activity, both from a theoretical point of view (see [43, 80–82] and extensions to viability theory in [83]), with the discovery of important relations between the fuzzy and the differential inclusions settings [84, 85] and from the computational aspects (in some cases with commonly inspired algorithms, see [21, 86]). Also of interest are recent developments in the study of fuzzy chaotic dynamical systems, in the continuous or discrete (maps) time settings [22, 87], and the inclusion of randomness into fuzzy mappings or fuzziness into random mappings), see [88].

The increased knowledge of the properties of spaces of fuzzy numbers and their metric topology has produced an important spin-off on the methodology of *fuzzy statistics* and *probability* and on linear/non-linear *fuzzy regression* and forecasting. The theory of statistical estimation of fuzzy relationships, in the setting of fuzzy random variables [89] and hypothesis testing [90, 91], has been improved; the used approaches are based on the linear structure of the space of fuzzy random variables, on the extension principle applied to classical estimators, or on a fuzzy least squares linear regression) estimation theory [92–95]. Also probabilistic approaches to fuzzy regression using linear programming formulations similar to Tanaka method, [96] and to statistical testing [97] have been analyzed.

Most of proposed methods and algorithms are efficiently implemented [98–103] and allow many shapes of fuzzy numbers, not restricted to symmetric or to linear simplifications.

In the areas of *systems engineering*, applications of traditional interval analysis are increasingly approached by fuzzy arithmetic with two substantial benefits in terms of a better setting for sensitivity to uncertain data and relations (due to the higher flexibility of fuzzy vs. interval uncertainty) and in terms of an increased attention (by the ‘fuzzy’ community) to the overestimation effect, associated with improper use of fuzzy interval arithmetic or to the underestimation effect, depending on the use of heuristic calculations with scarce control of approximation errors [1, 24, 104].

In applications of *approximate reasoning*, some new methodologies have been proposed, based on fuzzy arithmetic to define or implement fuzzy IF–THEN rules, with non-linear shapes of the fuzzy rulebases [14, 15, 105–107]. In [107] a procedure for the estimation of a fuzzy number from data is also suggested.

In recent years, following the ideas of GrC and fuzzy logic, relevant importance of fuzzy numbers and arithmetic has been associated with the universal approximation of *soft computing systems* [108] with applications to knowledge discovery [109] and data mining, to fuzzy systems and machine learning techniques [110], to fuzzy systems simulation, to expert systems [111], and, very recently, to fuzzy, geographic information systems (GIS) and spatial analysis [112–114].

² The references in this section are very essential and non-exhaustive of the current work. We indicate some of the more recent results and publications; the interested reader is addressed to the references therein.

In the various fields of *operations research*, recent work is addressing the inclusion of fuzzy methodologies for well-analyzed problems, e.g., scheduling [115, 116], assignment, location, graph-based optimization [117–119], network analysis, distribution and supply chain management [120, 121], vehicle routing [122], and for the fuzzy linear or non-linear programming problem [36, 123–126]. It is also increasing the number and quality of connections between fuzzy concepts and evolutionary methods for solving hard computational problems in global optimization [53, 127–129], integer programming, combinatorial optimization, or multiple criteria optimization.

Finally (and we omit many others) an emerging field of application of fuzzy tools is in *Economics*, covering fuzzy game theory [130–136], fuzzy preferences and decision making [137, 139], fuzzy Pareto optimality, and fuzzy DEA; in *Business*, with emphasis on knowledge intelligent support systems, project evaluation and investment decisions [139, 140]; in *Finance*, for financial pricing [141], fuzzy stochastic differential equations and option valuation [142–145], portfolio selection [146], and trading strategies.

It appears that the use of fuzzy numbers, possibly of general and flexible shape, and the search for sufficiently precise arithmetic algorithms is one of the current research fields of general attention by the scientific community and the number of applications is high and increasing.

References

- [1] H. Bandemer. *Mathematics of Uncertainty: Ideas, Methods, Application Problems*. Springer, New York, 2006.
- [2] L.A. Zadeh. Some reflections on soft computing, granular computing, and their roles in the computation, design and utilization of information/intelligent systems. *Soft Comput.* 2 (1998) 23–25.
- [3] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*, Kluwer, Dordrecht, 2003.
- [4] L.A. Zadeh. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.
- [5] L.A. Zadeh. Towards a generalized theory of uncertainty (GTU): an outline. *Inf. Sci.* 172 (2005) 1–40.
- [6] D. Dubois and H. Prade. Towards fuzzy differential calculus. *Fuzzy Sets Syst.* 8 (1982) 1–17(I), 105–116(II), 225–234(III).
- [7] R. Goetschel and W. Voxman. Elementary fuzzy calculus. *Fuzzy Sets Syst.* 18 (1986) 31–43.
- [8] P. Diamond and P. Kloden. *Metric Spaces of Fuzzy Sets*. World Scientific, Singapore, 1994.
- [9] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, 1980.
- [10] D. Dubois and H. Prade. Ranking fuzzy numbers in a setting of possibility theory. *Inf. Sci.* 30 (1983) 183–224.
- [11] D. Dubois and H. Prade. *Possibility Theory. An Approach to Computerized Processing of Uncertainty*. Plenum Press, New York, 1988.
- [12] D. Dubois and H. Prade (eds). *Fundamentals of Fuzzy Sets, The Handbooks of Fuzzy Sets Series*. Kluwer, Boston, 2000.
- [13] D. Dubois, E. Kerre, R. Mesiar, and H. Prade. Fuzzy interval analysis. In: D. Dubois and H. Prade (eds), *Fundamentals of Fuzzy Sets, The Handbooks of Fuzzy Sets Series*. Kluwer, Boston, 2000, pp. 483–581.
- [14] Y. Xu, E.E. Kerre, D. Ruan, and Z. Song. Fuzzy reasoning based on the extension principle. *Int. J. Intell. Syst.* 16 (2001) 469–495.
- [15] Y. Xu, J. Liu, D. Ruan, and W. Li. Fuzzy reasoning based on generalized fuzzy if-then rules. *Int. J. Intell. Syst.* 17 (2002) 977–1006.
- [16] G.J. Klir. Fuzzy arithmetic with requisite constraints. *Fuzzy Sets Syst.* 91 (1997) 165–175.
- [17] G.J. Klir. *Uncertainty Analysis in Engineering and Science*, Kluwer, Dordrecht, 1997.
- [18] G.J. Klir and Y. Pan. Constrained fuzzy arithmetic, basic questions and some answers. *Soft Comput.* 2 (1998) 100–108.
- [19] G.J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1995.
- [20] M.L. Guerra and L. Stefanini. Approximate fuzzy arithmetic operations using monotonic interpolations. *Fuzzy Sets Syst.* 150 (2005) 5–33.
- [21] L. Stefanini, L. Sorini, and M.L. Guerra. Parametric representation of fuzzy numbers and application to fuzzy calculus. *Fuzzy Sets Syst.* 157 (2006) 2423–2455.
- [22] L. Stefanini, L. Sorini, and M.L. Guerra. Simulation of fuzzy dynamical systems using the LU-representation of fuzzy numbers. *Chaos Solitons Fractals* 29(3) (2006) 638–652.
- [23] A. Kaufmann and M.M. Gupta. *Introduction to Fuzzy Arithmetic – Theory and Applications*. Van Nostrand Reinhold, New York, 1985.
- [24] H.-J. Zimmermann. *Fuzzy Set Theory and Its Applications*, 4th edn. Kluwer, Dordrecht, 2001.

- [25] S. Abbasbandy and M. Amirkhahrian. The nearest trapezoidal form of a generalized left right fuzzy number. *Int. J. Approx. Reason.* 43 (2006) 166–178.
- [26] P. Grzegorzewski. Nearest interval approximation of a fuzzy number. *Fuzzy Sets Syst.* 130 (2002) 321–330.
- [27] P. Grzegorzewski and E. Mrowka. Trapezoidal approximations of fuzzy numbers. *Fuzzy Sets Syst.* 153 (2005) 115–135. Revisited in *Fuzzy Sets Syst.* 158 (2007) 757–768.
- [28] M. Oussalah. On the compatibility between defuzzification and fuzzy arithmetic operations. *Fuzzy Sets Syst.* 128 (2002) 247–260.
- [29] M. Oussalah and J. De Schutter. Approximated fuzzy LR computation. *Inf. Sci.* 153 (2003) 155–175.
- [30] R.R. Yager. On the lack of inverses in fuzzy arithmetic. *Fuzzy Sets Syst.* 4 (1980) 73–82.
- [31] P. Diamond and P. Kloden. Metric Topology of fuzzy numbers and fuzzy analysis. In: D. Dubois and H. Prade (eds), *Fundamentals of Fuzzy Sets, The Handbooks of Fuzzy Sets Series*. Kluwer, Boston, 2000, pp. 583–641.
- [32] L.A. Zadeh. Concept of a linguistic variable and its application to approximate reasoning. I. *Inf. Sci.* 8 (1975) 199–249.
- [33] L.A. Zadeh. Concept of a linguistic variable and its application to approximate reasoning. II. *Inf. Sci.* 8 (1975) 301–357.
- [34] L.A. Zadeh. Concept of a linguistic variable and its application to approximate reasoning, III. *Inf. Sci.* 9 (1975) 43–80.
- [35] R. Fuller and P. Majlender. On interactive fuzzy numbers. *Fuzzy Sets Syst.* 143 (2004) 355–369;
- [36] M. Inuiguchi, J. Ramik, and T. Tanino. Oblique fuzzy vectors and their use in possibilistic linear programming. *Fuzzy Sets Syst.* 135 (2003) 123–150.
- [37] S. Heilpern. Representation and application of fuzzy numbers. *Fuzzy Sets Syst.* 91 (1997) 259–268.
- [38] C.-T. Yeh. A note on trapezoidal approximations of fuzzy numbers. *Fuzzy Sets Syst.* 158 (2007) 747–754.
- [39] L.A. Zadeh. Fuzzy Sets. *Inf. Control* 8 (1965) 338–353.
- [40] M.L. Guerra and L. Stefanini. On fuzzy arithmetic operations: some properties and distributive approximations. *Int. J. Appl. Math.* 19 (2006) 171–199. Extended version in *Working Paper Series EMS*, University of Urbino, 2005, available online by the RePEC Project (www.repec.org).
- [41] D. Ruiz and J. Torrens. Distributivity and conditional distributivity of a uniform and a continuous t-conorm. *IEEE Trans. Fuzzy Syst.* 14 (2006) 180–190.
- [42] B. Bouchon-Meunier, O. Kosheleva, V. Kreinovich, and H.T. Nguyen. Fuzzy numbers are the only fuzzy sets that keep invertible operations invertible. *Fuzzy Sets Syst.* 91 (1997) 155–163.
- [43] B. Bede and S.G. Gal. Generalizations of the differentiability of fuzzy number valued functions with applications to fuzzy differential equations. *Fuzzy Sets Syst.* 151 (2005) 581–599.
- [44] L. Sorini and L. Stefanini. *An LU-Fuzzy Calculator for the Basic Fuzzy Calculus*. Working Paper Series EMS No. 101. University of Urbino, Urbino, Italy, 2005. Revised and extended version available online by the RePEC Project (www.repec.org).
- [45] H.K. Chen, W.K. Hsu, and W.L. Chiang. A comparison of vertex method with JHE method. *Fuzzy Sets Syst.* 95 (1998) 201–214.
- [46] W.M. Dong and H.C. Shah. Vertex method for computing functions of fuzzy variables. *Fuzzy Sets Syst.* 24 (1987) 65–78.
- [47] E.N. Otto, A.D. Lewis, and E.K. Antonsson. Approximating α -cuts with the vertex method. *Fuzzy Sets Syst.* 55 (1993) 43–50.
- [48] W.M. Dong and F.S. Wong. Fuzzy weighted averages and implementation of the extension principle. *Fuzzy Sets Syst.* 21 (1987) 183–199.
- [49] M. Hanss. The transformation method for the simulation and analysis of systems with uncertain parameters. *Fuzzy Sets Syst.* 130 (2002) 277–289.
- [50] M. Hanss and A. Klimke. On the reliability of the influence measure in the transformation method of fuzzy arithmetic. *Fuzzy Sets Syst.* 143 (2004) 371–390.
- [51] A. Klimke. *An Efficient Implementation of the Transformation Method of Fuzzy Arithmetic*. Extended Preprint Report, 2003/009. Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, Germany, 2003.
- [52] A. Klimke and B. Wohlmuth. Computing expensive multivariate functions of fuzzy numbers using sparse grids. *Fuzzy Sets Syst.* 153 (2005) 432–453.
- [53] W.A. Lodwick and K.D. Jamison. Interval methods and fuzzy optimization. *Int. J. of Uncertain., Fuzziness and Knowl.-Based Reason.* 5 (1997) 239–250.
- [54] R. Moore and W.A. Lodwick. Interval analysis and fuzzy set theory. *Fuzzy Sets Syst.* 135 (2003) 5–9.
- [55] R.B. Kearfott and V. Kreinovich (eds). *Applications of Interval Analysis*. Kluwer, Dordrecht, 1996.

- [56] M. Navara and Z. Zabokrtsky. How to make constrained fuzzy arithmetic efficient. *Soft Comput.* 6 (2001) 412–417.
- [57] P.-T. Chang and K.-C. Hung. α -cut fuzzy arithmetic: simplifying rules and a fuzzy function optimization with a decision variable. *IEEE Trans. Fuzzy Syst.* 14 (2006) 496–510.
- [58] R. Hassine, F. Karray, A.M. Alimi, and M. Selmi. Approximation properties of piecewise parabolic functions fuzzy logic systems. *Fuzzy Sets Syst.* 157 (2006) 501–515.
- [59] K.L. Wood, K.N. Otto, and E.K. Antonsson. Engineering design calculations with fuzzy parameters. *Fuzzy Sets Syst.* 52 (1992) 1–20.
- [60] O.G. Duarte, M. Delgado, and I. Requena. Algorithms to extend crisp functions and their inverse functions to fuzzy numbers. *Int. J. Intell. Syst.* 18 (2003) 855–876.
- [61] L. Stefanini, L. Sorini, and M.L. Guerra. A parametrization of fuzzy numbers for fuzzy calculus and application to the fuzzy Black-Scholes option pricing. In: *Proceedings of the 2006 IEEE International Conference on Fuzzy Systems*, Vancouver, Canada, 2006, pp. 587–594. Extended version *Working Paper Series EMS*, No. 106. University of Urbino, Urbino, Italy, 2006.
- [62] M. De Munck, D. Moens, W. Desmet, and D. Vandepitte. An automated procedure for interval and fuzzy finite element analysis. *Proceedings of ISMA*, Leuven, Belgium, September, 20–22, 2004, pp. 3023–3033.
- [63] D. Moens and D. Vandepitte. Fuzzy finite element method for frequency response function analysis of uncertain structures. *AIAA J.* 40 (2002) 126–136.
- [64] K. Price. An introduction to differential evolution. In: D. Corne, M. Dorigo and F. Glover (eds). *New Ideas in Optimization*. McGraw-Hill, New York, 1999, pp. 79–108.
- [65] R. Storn and K. Price. *Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*. ICSI Technical Report TR-95-012. Berkeley University, Berkeley, CA, 1995. Also. *J. Glob. Optim.* 11 (1997) 341–359.
- [66] R. Storn. System design by constraint adaptation and differential evolution. *IEEE Trans. Evol. Comput.* 3 (1999) 22–34.
- [67] L. Stefanini. *Differential Evolution Methods for Arithmetic with Fuzzy Numbers*. Working Paper Series EMS No. 104. University of Urbino, Urbino, Italy, 2006. Available online by the RePEc Project (www.repec.org).
- [68] M.M. Ali and A. Törn. Population set-based global optimization algorithms and some modifications and numerical studies. *Comput. Oper. Res.* 31 (2004) 1703–1725.
- [69] O. Kaleva. Fuzzy differential equations. *Fuzzy Sets Syst.* 24 (1987) 301–317.
- [70] O. Kaleva. The calculus of fuzzy valued functions. *Appl. Math. Lett.* 3 (1990) 55–59.
- [71] M. Puri and D. Ralescu. Differentials of fuzzy functions. *J. Math. Anal. Appl.* 91 (1983) 552–558.
- [72] Y.L. Kim and B.M. Ghil. Integrals of fuzzy-number-valued functions. *Fuzzy Sets Syst.* 86 (1997) 213–222.
- [73] J.J. Buckley and A. Yan. Fuzzy functional analysis (I): basic concepts. *Fuzzy Sets Syst.* 115 (2000) 393–402.
- [74] T. Allahviranloo. The Adomian decomposition method for fuzzy systems of linear equations. *Appl. Math. Comput.* 163 (2005) 553–563.
- [75] T. Allahviranloo. Successive over relaxation iterative method for fuzzy system of linear equations. *Appl. Math. Comput.* 162 (2005) 189–196.
- [76] B. Asady, S. Abbasbandy, and M. Alavi. Fuzzy general linear systems. *Appl. Math. Comput.* 169 (2005) 34–40.
- [77] A. Vroman, G. Deschrijver, and E.E. Kerre. Solving systems of linear fuzzy equations by parametric functions; an improved algorithm. *Fuzzy Sets Syst.* 158 (2007) 1515–1534.
- [78] S.M. Wang, S.C. Fang, and H.L.W. Nuttle. Solution sets of interval valued fuzzy relational equations. *Fuzzy Optim. Decis. Mak.* 2 (2003) 41–60.
- [79] P. Diamond. Theory and applications of fuzzy Volterra integral equations. *IEEE Trans. Fuzzy Syst.* 10 (2002) 97–102.
- [80] D.N. Georgiou, J.J. Nieto, and R. Rodriguez-Lopez. Initial value problems for higher order fuzzy differential equations. *Nonlinear Anal.* 63 (2005) 587–600.
- [81] J.J. Nieto. The Cauchy problem for continuous fuzzy differential equations. *Fuzzy Sets Syst.* 102 (1999) 259–262.
- [82] V. Lakshmikantham. Set differential equations versus fuzzy differential equations. *Appl. Math. Comput.* 164 (2005) 277–294.
- [83] R.P. Agarwal, D. O'Regan, and V. Lakshmikantham. Viability theory and fuzzy differential equations. *Fuzzy Sets Syst.* 151 (2005) 563–580.
- [84] T.G. Bhaskar, V. Lakshmikantham, and V. Devi. Revisiting fuzzy differential equations. *Nonlinear Anal.* 58 (2004) 351–358.
- [85] V. Lakshmikantham and A.A. Tolstonogov. Existence and interrelation between set and fuzzy differential equations. *Nonlinear Anal.* 55 (2003) 255–268.

- [86] K.R. Jackson and N.S. Nedialkov. Some recent advances in validated methods for IVPs and ODEs. *Appl. Numer. Math.* 42 (2002) 269–284.
- [87] S.M. Pederson. Fuzzy homoclinic orbits and commuting fuzzifications. *Fuzzy Sets Syst.* 155 (2005) 361–371.
- [88] R. Ahmad and F.F. Bazan. An interactive algorithm for random generalized nonlinear mixed variational inclusions for random fuzzy mappings. *Appl. Math. Comput.* 167 (2005) 1400–1411.
- [89] V. Krätschmer. A unified approach to fuzzy random variables. *Fuzzy Sets Syst.* 123 (2001) 1–9.
- [90] W. Näther. Random fuzzy variables of second order and applications to statistical inference. *Inf. Sci.* 133 (2001) 69–88.
- [91] H.C. Wu. Statistical hypothesis testing for fuzzy data. *Inf. Sci.* 175 (2005) 30–56.
- [92] R. Alex. A new kind of fuzzy regression modeling and its combination with fuzzy inference. *Soft Comput.* 10 (2006) 618–622.
- [93] V. Krätschmer. Strong consistency of least squares estimation in linear regression model with vague concepts. *J. Multivariate Anal.* 97 (2006) 633–654.
- [94] V. Krätschmer. Least squares estimation in linear regression models with vague concepts. *Fuzzy Sets Syst.* 157 (2006) 2579–2592.
- [95] A. Wünsche and W Näther. Least squares fuzzy regression with fuzzy random variable. *Fuzzy Sets Syst.* 130 (2002) 43–50.
- [96] M. Modares, E. Nasrabadi, and M.M. Nasrabadi. Fuzzy linear regression models with least squares errors. *Appl. Math. Comput.* 163 (2005) 977–989.
- [97] O. Hryniewicz. Possibilistic decisions and fuzzy statistical tests. *Fuzzy Sets Syst.* 157 (2006) 2665–2673.
- [98] P. D'Urso. Linear regression analysis for fuzzy/crisp input and fuzzy/crisp output data. *Comput. Stat. Data Anal.* 42 (2003) 47–72.
- [99] P. D'Urso and A. Santoro. Goodness of fit and variable selection in the fuzzy multiple linear regression. *Fuzzy Sets Syst.* 157 (2006) 2627–2647.
- [100] M. Hojati, C.R. Bector, and K. Smimou. A simple method for computation of fuzzy linear regression. *Eur. J. Oper. Res.* 166 (2005) 172–184.
- [101] D.H. Hong, J.-K. Song, and H.Y. Do. Fuzzy least squares linear regression analysis using shape preserving operations. *Inf. Sci.* 138 (2001) 185–193.
- [102] C. Kao and C.-L. Chyu. Least squares estimates in fuzzy regression analysis. *Eur. J. Oper. Res.* 148 (2003) 426–435.
- [103] H.K. Yu. A refined fuzzy time series model for forecasting. *Physica A* 346 (2005) 347–351.
- [104] O. Wolkenhauser. *Data Engineering: Fuzzy Mathematics in Systems Theory and Data Analysis*. Wiley, New York, 2001.
- [105] M. Delgado, O. Duarte, and I. Requena. An arithmetic approach for the computing with words paradigm. *Int. J. Intell. Syst.* 21 (2006) 121–142.
- [106] V.G. Kaburlasos. FINs: lattice theoretic tools for improving prediction of sugar production from populations of measurement. *IEEE Trans. Syst. Man, Cybern. B* 34 (2004) 1017–1030.
- [107] V.G. Kaburlasos and A. Kehagias. Novel fuzzy inference system (FIS) analysis and design based on lattice theory. *IEEE Trans. Fuzzy Syst.* 15 (2007) 243–260.
- [108] S. Wang and H. Lu. Fuzzy system and CMAC network with B-spline membership/basis functions are smooth approximators. *Soft Comput.* 7 (2003) 566–573.
- [109] Y.Q. Zhang. Constructive granular systems with universal approximation and fast knowledge discovery. *IEEE Trans. Fuzzy Syst.* 13 (2005) 48–57.
- [110] H.K. Lam, S.H. Ling, F.H.F. Leung, and P.K.S. Tam. Function estimation using a neural fuzzy network and an improved genetic algorithm. *Int. J. Approx. Reason.* 36 (2004) 243–260.
- [111] S.H. Liao. Expert system methodologies and applications: a decade review from 1995 to 2004. *Expert Syst. Appl.* 28 (2005) 93–103.
- [112] G. Bordogna, S. Chiesa, and D. Geneletti. Linguistic modeling of imperfect spatial information as a basis for simplifying spatial analysis. *Inf. Sci.* 176 (2006) 366–389.
- [113] Y. Li and S. Li. A fuzzy set theoretic approach to approximate spatial reasoning. *IEEE Trans. Fuzzy Syst.* 13 (2005) 745–754.
- [114] S.Q. Ma, J. Feng, and H.H. Cao. Fuzzy model of regional economic competitiveness in Gis spatial analysis: case study of Gansu, Western China. *Fuzzy Optim. Deci. Mak.* 5 (2006) 99–112.
- [115] W. Herroelen and R. Leus. Project scheduling under uncertainty: survey and research potentials. *Eur. J. Oper. Res.* 165 (2005) 289–306.
- [116] S. Petrovic and X.Y. Song. A new approach to two machine flow shop problem with uncertain processing times. *Optim. Eng.* 7 (2006) 329–342.
- [117] S. Muñoz, M.T. Otuño, J. Ramirez, and J. Yañez. Coloring fuzzy graphs. *Omega* 33 (2005) 211–221.

- [118] T. Savsek, M. Vezjah, and N. Pavesic. Fuzzy trees in decision support systems. *Eur. J. Oper. Res.* 174 (2006) 293–310.
- [119] A. Sengupta and T.K. Pal. Solving the shortest path problem with interval arcs. *Fuzzy Optim. Decis. Mak.* 5 (2006) 71–89.
- [120] R. Alex. Fuzzy point estimation and its application on fuzzy supply chain analysis. *Fuzzy Sets Syst.* 158 (2007) 1571–1587.
- [121] J. Wang and Y.F. Shu. Fuzzy decision modeling for supply chain management. *Fuzzy Sets Syst.* 150 (2005) 107–127.
- [122] E.E. Ammar and E.A. Youness. Study of multiobjective transportation problem with fuzzy numbers. *Appl. Math. Comput.* 166 (2005) 241–253.
- [123] G. Facchinetti, S. Giove, and N. Pacchiarotti. Optimisation of a nonlinear fuzzy function. *Soft Comput.* 6 (2002) 476–480.
- [124] K. Ganeshan and P. Veeramani. Fuzzy linear programs with trapezoidal fuzzy numbers. *Ann. Oper. Res.* 143 (2006) 305–315.
- [125] F.F. Guo and Z.Q. Xia. An algorithm for solving optimization problems with one linear objective function and finitely many constraints of fuzzy relation inequalities. *Fuzzy Optim. Decis. Mak.* 5 (2006) 33–48.
- [126] J. Ramik. Duality in fuzzy linear programming: some new concepts and results. *Fuzzy Optim. Decis. Mak.* 4 (2005) 25–40.
- [127] J. Alami, A. El Imrani, and A. Bouroumi. A multipopulation cultural algorithm using fuzzy clustering. *Appl. Soft Comput.* 7 (2007) 506–519.
- [128] J. Liu and J. Lampinen. A fuzzy adaptive differential evolution algorithm. *Soft Comput.* 9 (2005) 448–462.
- [129] W.A. Lodwick and K.A. Bachman. Solving large scale fuzzy and possibilistic optimization problems. *Fuzzy Optim. Deci. Mak.* 4 (2005) 257–278.
- [130] B. Arfi. Linguistic fuzzy logic game theory. *J. Confl. Resolut.* 50 (2006) 28–57.
- [131] D. Butnariu. Fuzzy games: a description of the concept. *Fuzzy Sets Syst.* 1 (1978) 181–192.
- [132] D. Garagic and J.B. Cruz. An approach to fuzzy noncooperative Nash games. *J. Optim. Theory Appl.* 118 (2003) 475–491.
- [133] M. Mares. *Fuzzy Cooperative Games: Cooperation with Vague Expectations (Studies in Fuzziness and Soft Computing)*, Physica-Verlag, Heidelberg, 2001.
- [134] M. Mares. On the possibilities of fuzzification of the solution in fuzzy cooperative games. *Mathw. Soft Comput.* IX (2002) 123–127.
- [135] Q. Song and A. Kandel. A fuzzy approach to strategic games. *IEEE Trans. Fuzzy Syst.* 7 (1999) 634–642.
- [136] L. Xie and M. Grabisch. The core of bicapacities and bipolar games. *Fuzzy Sets Syst.* 158 (2007) 1000–1012.
- [137] B. Matarazzo and G. Munda. New approaches for the comparison of LR numbers: a theoretical and operational analysis. *Fuzzy Sets Syst.* 118 (2001) 407–418.
- [138] R.R. Yager. Perception based granular probabilities in risk modeling and decision making. *IEEE Trans. Fuzzy Syst.* 14 (2006) 329–339.
- [139] E.E. Ammar and H.A. Khalifa. Characterization of optimal solutions of uncertainty investment problems. *Appl. Math. Comput.* 160 (2005) 111–124.
- [140] C. Kahraman, D. Ruan, and E. Tolga. Capital budgeting techniques using discounted fuzzy versus probabilistic cash flows. *Inf. Sci.* 142 (2002) 57–76.
- [141] J. de A. Sanchez and A.T. Gomez. Estimating a term structure of interest rates for fuzzy financial pricing by using fuzzy regression methods. *Fuzzy Sets Syst.* 139 (2003) 313–331.
- [142] S. Li and A. Ren. Representation theorems, set valued and fuzzy set valued Ito integral. *Fuzzy Sets Syst.* 158 (2007) 949–962.
- [143] I. Skrjanc, S. Blazic, and O. Agamennoni. Interval fuzzy modeling applied to Wiener models with uncertainties. *IEEE Trans. Syst. Man Cybernet. B* 35 (2005) 1092–1095.
- [144] Y. Yoshida, M. Yasuda, J.I. Nakagami, and M. Kurano. A discrete time american put option model with fuzziness of stock process. *Fuzzy Optim. Decis. Mak.* 4 (2005) 191–208.
- [145] H.C. Wu. European option pricing under fuzzy environments. *Int. J. Intell. Syst.* 20 (2005) 89–102.
- [146] X.X. Huang. Fuzzy chance-constrained portfolio selection. *Appl. Math. Comput.* 177 (2006) 500–507.

13

Rough-Granular Computing

Andrzej Skowron and James F. Peters

13.1 Introduction

This chapter briefly introduces the highlights of the rough set approach to knowledge discovery by means of various forms of information granulation. This approach has its roots in the seminal work began during the early 1980s and which continued until very recently by Zdzisław Pawlak.¹ The fulcrum of the rough set approach is the indiscernibility relation introduced by Zdzisław Pawlak during the early 1980s. This relation makes it possible to partition sets of sample perceptual objects or sets of conceptual objects into collections of classes called elementary sets. An *elementary set* is an equivalence class that is a penultimate example of an information granule, i.e., a set of sample objects with matching descriptions that reveal affinities between the objects in the sample. A by-product of this introduction to the rough set approach to knowledge discovery is a rather complete view of what is known as rough-granular computing (RGC).

The basic ideas of rough set theory and its extensions as well as many interesting applications can be found in a number of books (see, e.g., [1–20]), issues of the transactions on rough sets [21–27], special issues of other journals (see, e.g., [28–38]), proceedings of international conferences (see, e.g., [39–56]), and surveys (see, e.g., [28–30]).²

The basic notions of rough sets and approximation spaces were introduced during the early 1980s (see, e.g., [57–59]). In this chapter, the basic concepts of rough set theory are presented. We also point out some research directions and applications based on rough sets.

We outline an approach to information granulation and granular computing (GC) based on rough sets.

Information granulation can be viewed as a human way of achieving data compression and it plays a key role in the implementation of the strategy of divide and conquer in human problem solving [60]. Objects obtained as the result of granulation are information granules. Examples of elementary information granules are indiscernibility or tolerance (similarity) classes (see, e.g., [30]). In reasoning about data and knowledge under uncertainty and imprecision many other more compound information granules are used (see, e.g., papers by Skowron and others in [11, 31], in this handbook, and papers [61, 62]). Examples

¹ Zdzisław Pawlak passed away on April 7, 2006.

² For more information one can also visit a number of Web pages; see, e.g., <http://www.roughsets.org> and <http://logic.mimuw.edu.pl>.

of such granules are decision rules, sets of decision rules, or classifiers. More compound information granules are defined by means of less compound ones. Note that inclusion or closeness measures between information granules should be considered rather than their strict equality. Such measures are also defined recursively for information granules.

In GC we search for granules satisfying some criteria. These criteria can be based on the minimal length principle, can express acceptable risk degrees of granules, or can use some utility functions. We discuss the role of approximation spaces in modeling granules satisfying such criteria.

Granules are obtained in the process of information granulation. GC is based on processing of complex information entities called granules. Generally speaking, granules are collection of entities, which are arranged together because of their similarity, functional adjacency, or indistinguishability [60].

One of the main branch of GC is computing with words and perceptions (CWP). GC ‘derives from the fact that it opens the door to computation and reasoning with information which is perception – rather than measurement based. Perceptions play a key role in human cognition and underlie the remarkable human capability to perform a wide variety of physical and mental tasks without any measurements and any computations. Everyday examples of such tasks are driving a car in city traffic, playing tennis, and summarizing a story’ [60].

In GC we solve optimization tasks based on searches for optimal solutions satisfying some constraints to a satisfactory degree. The optimization criteria are related to the minimization of the description size of the solutions. The constraints are often vague, imprecise, and/or specifications of concepts and dependencies between them involved in the constraints are incomplete. Decision tables [13] are examples of such constraints. Another example of constraints can be found, e.g., in papers by Skowron and others in [11, 46] where a specification is given by a domain knowledge and data sets. Domain knowledge is represented by ontology of vague concepts and dependencies between them. In a more general case, the constraints can be specified in a simplified fragment of a natural language [60]. The solutions are represented by composite granules constructed from some elementary ones. Both elementary granules and the schemes of construction of more composite granules should be discovered in the process of searching for solutions. Moreover, in this process one should be able to measure matching (inclusion) degrees of different granules. All these components define what is known as granule systems. In the optimization process for a given task, e.g., construction of a classifier for a complex concept, it is necessary to discover a relevant granule system.

In this book, several topics of the rough-granular approach are covered. Among them are general aspects of rough fuzzy granulation, modeling granules in hierarchical learning, ontology approximation and planning, construction of granules satisfying a given specification, GC for ordered data, GC and rough set methods for dealing with missing values, wisdom granular computing (WGC), granulation by rough clustering, rough-neuro hybridization in the GC framework, granulation for Web documents, outlier and exception analysis in rough sets and GC, approximation spaces and perceptual granulation in reinforcement learning, GC based on rough mereology, conflict analysis in the framework of rough sets and GC, rough and granular case-based reasoning, general aspects of RGC, granulation in spatiotemporal reasoning, GC in bioinformatics, and granulation on analogy-based classification.

13.2 Objects, Granules, and Vague Concepts

In this section, we give some general remarks about objects, granules, sets, the notion of vagueness, and the place of rough sets in set theory. In the sequel, two basic types of objects are identified, namely, percepts and concepts. These objects provide the basic building blocks for information granules. The description of an object is a pivotal notion in rough set theory. Each object description is in the form of a vector of function values representing either features of perceptual objects or attributes of conceptual objects. For this reason, a brief consideration of features and attributes is also briefly considered in this chapter. Granulation and the discovery of information granules of interest is viewed in the context of sets, i.e., collections of objects. Hence, a brief consideration of sets is important for an understanding of GC. One of the strengths of the approach to approximation in rough set theory is a provision for measuring

the extent that a set of objects can be considered vague. Vagueness is also briefly considered in this chapter.

Object [L. *objectum*],

a thing thrown before or

presented to the senses or mind.

—Walter W. Skeat, Etymological Dictionary, 1879.

Perceptible [L. *perceptibil-is* (Cassiod, Boethius)],

capable of being perceived by the senses

or intellect, cognizable, apprehensible; observable.

—The Oxford English Dictionary, 1933.

Concept [ad. late L. *concept-um*].

2. Logic and Philos.

The product of the faculty of conception; an idea of a class of objects, a general notion or idea.

—The Oxford English Dictionary, 1933.

Granule [from *granum*, Lat.].

A small compact particle.

—Samuel Johnson, 1816.

To Granulate. v.a.

1. To break into small masses or granules.

—Samuel Johnson, 1816.

Granular, a. [f. late L. *granulat-um*].

1. Consisting grains or granules;

existing in the condition of grains or granules.

—The Oxford English Dictionary, 1933.

13.2.1 Objects and Granules

An *object* is understood to be something perceptible to the senses or knowable by the mind. Basically, a *granule* is a set of objects that have something in common, i.e., an affinity (close relationship). An *information granule* is a set of objects that have matching descriptions.

The natural world provides a treasure trove of examples of objects that can be perceived by the senses. Consider, for example, the sample conches in Figures 13.1a and 13.1b, where one can discern many tiny conical projections arranged in tiers in the conch surface (see Figure 13.1c). Each tier in the conch surface can be viewed as a granule (collection of conical projections in a conch tier). The conches in Figure 13.1 come from the coral reef along the coastline of the Fiji islands. Constructs from either logic or mathematics provide a rich lore of examples of objects knowable by the mind.

A rough set approach to approximation tends to be either percept centered or concept centered depending on the application domain.

A *percept* is an object of perception. A percept-centered approach to approximation focuses on objects perceived by the senses. This approach is common in science and engineering. Let a percept be denoted by $\langle \text{descriptor} \rangle^P$. Examples of percepts are coral pores^P, pixel-windows^P of an image^P, observed behavior^P of an organism^P, wind direction^P, characters^P on a page in a book, observed patients^P in a hospital, and observed weights^P and observed measurements^P from laboratory experiments. Characteristic features such as shape, habitat, or color are typically associated with visual perceptions of sample objects in the neighborhood of an observer's environment (see, e.g., paper by Peters in [22, 63]).

A *concept* is basically a notion or idea about objects. A concept-centered approach to approximation focuses on objects knowable by the mind. Let a concept be denoted by $\langle \text{descriptor} \rangle^C$. Examples

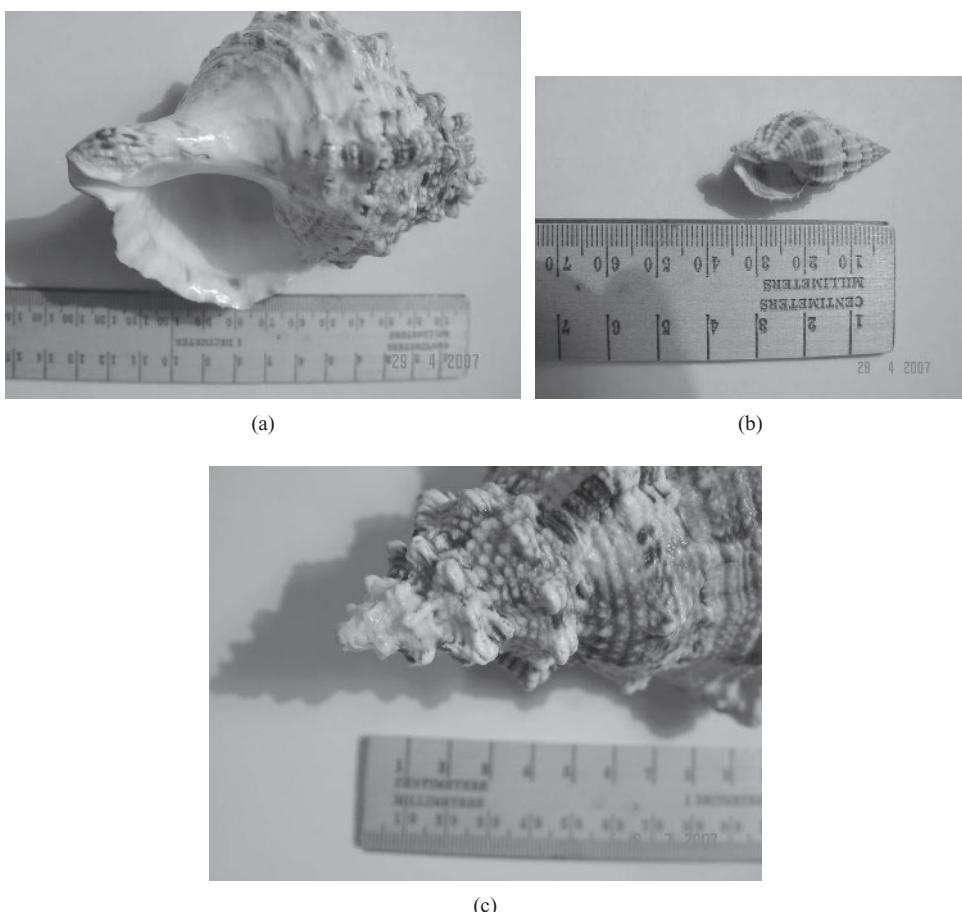


Figure 13.1 Sample objects: Fiji conch shells: (a) large Fiji conch, (b) small Fiji conch and (c) conical projections

of concepts are sets of objects representing an idea (e.g., coral pores that exhibit deterioration of the exoskeleton of a marine animal, where the concept of interest is *deterioration*^c), animation^c in a Web page, *income bracket*^c in a tax table, *disease symptom*^c in clinical data from physicians' reports, and *rhyming words*^c in a dictionary. We use the term *attribute* to denote properties inherent in things knowable by the mind. Attributes tend to reflect abstract notions such as time, grade point average, speed, terrain familiarity [13], or temperature, humidity, windy, outlook, forecast [64], or city, state, population [65] with associated values organized in a table. Attributes are commonly associated with data in either database theory [65] or data mining [64], where the focus is on characterizing abstract objects such as city (Winnipeg), province (Manitoba), and population (e.g., 648,600 in Winnipeg in 2006) in the form of numbers and symbols. Characteristic attributes such as quantity, quality, relation, and modality³ are typically associated with abstract objects.

Objects are compared by considering measurements that *represent* either features or attributes. Feature language is commonly used in science, pattern recognition, and machine learning as a means of characterizing perceived objects. By contrast, attribute language has a long history in rough set theory that originates with Zdzisław Pawlak's lifelong interest in information systems and knowledge representation with rules that are attribute based. Attributes are commonly associated with data rather than percepts. The

³ Emmanuel Kant calls these categories pure concepts.

important thing to notice at the outset is that both schools of thought solve the approximation problem by introducing functions that *represent* either features of percepts or attributes of abstract objects. These functions map objects to values (measurements) that make it possible to compare and group objects. The paradox in rough set theory is that variables such as x are used to represent either concrete objects (percepts) or abstractions (concepts). This is a direct result of Stanisław Leśniewski's mereology [66], where there is place for variables that represent either mathematical (knowable) objects such as numbers, functions, points, sets, and surfaces or physical (perceivable) objects such as icicles, snow flakes, coral, rocks, trees, and biological organisms. It was Leśniewski who suggested that theorems should harmonize with 'common sense.' One of the strengths of the rough set approach to approximation is that it has utility in either the percept-oriented or concept-oriented approach.

13.2.2 Sets and Information Granules

The notion of a set (collection of objects) is a basic one in mathematics and provides a means of representing information granules.

The definition of this notion and the creation of set theory are due to German mathematician Georg Cantor (1845–1918), who laid the foundations of contemporary set theory over 100 years ago.

The birth of set theory can be traced back to Georg Cantor's 1873 proof of the uncountability of real line (i.e., the set of all real numbers is not countable). It was Bernhard Bolzano (1781–1848) who coined the term *Menge* ('set'), which Cantor used to refer to objects in his theory. According to Cantor, a set is a collection of any objects, which can be considered as a whole according to some law. As one can see, the notion of set is very intuitive and simple.

Mathematical objects such as relations, functions, and numbers are examples of sets. In fact, set theory is needed in mathematics to provide rigor.

The notion of a set is not only fundamental for the whole of mathematics but it also plays an important role in natural language. We often speak about sets (collections) of various objects of interest, such as collection of books, paintings, and people.

The intuitive meaning of a set according to some dictionaries is the following:

'A number of things of the same kind that belong or are used together.'

—Webster's Dictionary

'Number of things of the same kind, that belong together because they are similar or complementary to each other.'

—The Oxford English Dictionary

Thus a set is a collection of things which are somehow related to each other but the nature of this relationship is not specified in these definitions. Sets of objects with matching descriptions are of particular interest because such sets constitute information granules. In a nutshell, an *information granule* is a set of objects with matching descriptions, e.g., objects belonging to the same interval, part of the same fuzzy set, and objects having the same features. Examples of granules are sets of conch conical projections with the same height and diameter (at the base), sets of novels by the same author with the same characters, and so on.

13.2.3 Antinomies

In 1903, the renowned English philosopher Bertrand Russell (1872–1970) observed that the intuitive notion of a set given by Cantor leads to logical *antinomies* (contradictions); i.e., Cantor's set theory is contradictory. (There are other kinds of antinomies, which are outside the scope of this chapter.) A logical antinomy (for simplicity, we refer to 'antinomy' in the rest of this chapter) arises whenever correct logical reasoning leads to a contradiction, i.e., to propositions A and $\text{non-}A$.

As an example let us discuss briefly the so-called Russell's antinomy. Consider the set X containing all the sets Y , which are not the elements of themselves. If we assume that X is its own element then X , by

definition, cannot be its own element; while if we assume that X is not its own element then, according to the definition of the set X , it must be its own element. Thus while applying each assumption we obtain contradiction.

Antinomies show that a set cannot be a collection of arbitrary elements, as was stipulated by Cantor.

One could think that antinomies are ingenuous logical play, but it is not so. They question the essence of logical reasoning. That is why there have been attempts to ‘repair’ Cantor’s theory for over 100 years or to substitute another set theory for it, but the results have not been good so far. Is then all mathematics based on doubtful foundations?

As a remedy for this defect several axiomatizations of set theory have been proposed (e.g., Zermelo [67]).

Instead of improvements of Cantor’s set theory by its axiomatization, some mathematicians proposed escape from classical set theory by creating a completely new idea of a set, which would free the theory from antinomies.

No doubt the most interesting proposal was given by Polish logician Stanisław Leśniewski, who introduced the relation of ‘being a part’ instead of the membership relation between elements and sets employed in classical set theory. In his set theory called mereology, *being a part* is a fundamental relation [66]. Mereology is a significant part of recent studies on the foundations of mathematics, artificial intelligence, cognitive science, natural language, and research in rough set theory (see, e.g., [14] and papers by Polkowski *et al.* in [15, 68]).

The deficiency of sets, mentioned above, has rather philosophical than practical meaning, since sets used in contemporary mathematics tend to be free from antinomies. The antinomies themselves are associated with very ‘artificial’ sets of all sets but are not found in sets commonly used in mathematics. For this reason, using set theory as a means of gathering together objects in information granules is plausible.

13.2.4 Vagueness

Another issue discussed in connection with sets of objects is vagueness, especially in the context of information granulation. Mathematics requires that all mathematical notions (including set) must be exact; otherwise precise reasoning would be impossible. However, philosophers [69] and recently many other researchers have become interested in (imprecise) concepts (see, e.g., [70]).

In classical set theory, a set is uniquely determined by its elements. In other words, this means that every element must be uniquely classified as belonging to the set or not. That is to say the notion of a set is a *crisp* (precise) one. For example, the set of odd numbers is crisp because every number is either odd or even.

In contrast to odd numbers, the notion of a beautiful painting is vague, because we are unable to classify uniquely all paintings into two classes: beautiful and not beautiful. Some paintings cannot be decided whether they are beautiful or not and thus they remain in the doubtful area. Thus, *beauty* is not a precise but a vague concept.

Almost all concepts we are using in natural language are vague. Therefore, commonsense reasoning based on natural language must be based on vague concepts and not on classical logic. Interesting discussion of this issue can be found in [71].

The idea of vagueness can be traced back to the ancient Greek philosopher Eubulides of Megara (ca. 400 BC), who first formulated so-called sorites (heap) and falakros (bald man) paradoxes (see, e.g., [69]). The bald man paradox goes as follows: suppose a man has 100,000 hairs on his head. Removing one hair from his head surely cannot make him bald. Repeating this step we arrive at the conclusion that a man without any hair is not bald. Similar reasoning can be applied to a heap of stones.

Vagueness is usually associated with the boundary region approach (i.e., existence of objects which cannot be uniquely classified relative to a set or its complement), which was first formulated in 1893 by the father of modern logic, German logician, Gottlob Frege (1848–1925) (see [72]). Consideration of the boundary region in the rough set approach to approximation has been the focus of a number of recent studies (see, e.g., paper by Peters in [54]).

According to Frege the concept must have a sharp boundary. To the concept without a sharp boundary there would correspond an area that would not have any sharp boundary line all around. This means that mathematics must use crisp, not vague, concepts; otherwise it would be impossible to reason precisely.

Perception [L. *perception-em*],
 ‘sensuous or mental apprehension, intelligence,
 knowledge’,

6. In strict philosophical language:
 The action of the mind by which it refers its
 sensations to an external object as their cause.

—The Oxford English Dictionary, 1933.

13.3 Rough Sets

This section briefly delineates basic concepts in rough set theory.

13.3.1 Rough Sets: An Introduction

Rough set theory, proposed by Pawlak in 1982 [13, 59], can be seen as a new mathematical approach to vagueness.

The rough set philosophy is founded on the assumption that with every object of the universe of discourse we associate some information (object descriptions and knowledge derived from study of classes of objects of interest). For example, if perceptible objects are patients suffering from a certain disease, symptoms of the disease are a source of information about the patients. Objects characterized by the same information are indiscernible (similar) in view of the available information about them. The indiscernibility relation generated in this way is the mathematical basis of rough set theory. This understanding of indiscernibility is related to the idea of Gottfried Wilhelm Leibniz’s identity of indiscernibles [73], which is a principle of analytic ontology. The basic assumption is that no two distinct substances exactly resemble each other. Two objects x and y are indiscernible if, for every property F , an object x has F if and only if object y has F . However, in the rough set approach, indiscernibility is defined relative to selected sets of functions representing attributes or features of objects of interest. Objects x and y are deemed indiscernible in the case the objects have matching descriptions, i.e., matching function values.

Any set of all indiscernible (similar) objects is called an elementary set, and forms a basic granule (atom) of knowledge about the universe. Any union of some elementary sets is referred to as crisp (precise) set – otherwise the set is rough (imprecise, vague).

Consequently, each rough set has boundary line cases, i.e., objects which cannot with certainty be classified either as members of the set or as members of its complement. Obviously crisp sets have no boundary line elements at all. This means that boundary line cases cannot be properly classified by employing available knowledge.

Thus, the assumption that objects can be ‘seen’ only through the information available about them leads to the view that knowledge has granular structure. Due to the granularity of knowledge, some objects of interest cannot be discerned and appear as the same (or similar). As a consequence, vague perception in contrast to precise perception cannot be characterized in terms of information about their elements. Therefore, in the proposed approach, we assume that any vague concept is replaced by a pair of precise perceptions – called the lower and the upper approximation of the vague perception. The lower approximation consists of all objects which surely belong to the concept and the upper approximation contains all objects which possibly belong to the perception. The difference between the upper and the lower approximation constitutes the boundary region of the vague perception. Approximations are two basic operations in rough set theory.

Hence, rough set theory expresses vagueness not by means of membership, but by employing a boundary region of a set. If the boundary region of a set is empty, it means that the set is crisp; otherwise the set is rough (inexact). A non-empty boundary region of a set means that our knowledge about the set is not sufficient to define the set precisely.

Rough set theory is not an alternative to classical set theory but it is embedded in it. Rough set theory can be viewed as a specific implementation of Frege’s idea of vagueness; i.e., imprecision in this approach is expressed by a boundary region of a set.

Rough set theory has attracted attention of many researchers and practitioners all over the world, who have contributed essentially to its development and applications. Rough set theory overlaps with many other theories. Despite this overlap, rough set theory may be considered as an independent discipline in its own right. The rough set approach seems to be of fundamental importance in artificial intelligence and cognitive sciences, especially in research areas such as machine learning, intelligent systems, inductive reasoning, pattern recognition, mereology, knowledge discovery, decision analysis, and expert systems.

13.3.2 Hallmarks of Rough Set Theory

Rough set theory has a kinship with fuzzy set theory. This kinship emerges if one considers functions that measure the appearance of objects with behavioral features expressible by words such as *low*, *medium*, and *high*. Typically, object behavior is in the form of signals with measurable amplitudes. Functions representing signal amplitudes provide a basis for describing, granulating, and analyzing object behavior. In both settings, functions are chosen to measure the appearance of sample objects. Selected functions provide a basis for constructing tables of values associated with sample objects. Notice that the design of functions in the rough set approach to representing object features does suggest the helpfulness of guesswork as well as prior knowledge about probability distributions of objects (events) in random samples. The fuzzy set approach to object description and information granulation is very helpful at the threshold of every consideration of the roughness of sets of sample objects. This is so because the fuzzy set approach focuses on how to get started in formulating object descriptions, i.e., selection of functions that capture high-level features of objects of interest.

Rough set theory distinguishes itself from traditional fuzzy set theory by ushering in an approach to granulating object universes and opening the door to a study of just how near each complete set of sample objects is to designated objects of interest. The hallmark of the rough set approach is the introduction of the indiscernibility relation as a means of granulating sets of sample objects. Thanks to the rough set approach to information granulation, it is possible to gauge the efficacy of functions used to represent object features. This suggests the utility of rough set methods in evaluating chosen functions representing object features. Obviously, chosen features merit revision and redesign in cases where there is significant separation (lack of description-based commonality) between available sample objects and objects of interest. This is important, since it is widely recognized that feature selection and feature function construction are two of the great challenges in information-theoretic methods [13].

One can observe the following high water marks in the rough set approach:

- granulation of sets of sample objects,
- description-based information granules,
- measures of nearness of information granules to sets of objects of interest,
- frameworks to facilitate perception of sample objects,
- feature selection aided by perception of nearness of information granules to sets of objects of interest,
- efficient algorithms for finding hidden patterns in sample sets of objects,
- methods to identify superfluous functions in object descriptions,
- determination of optimal sets of objects (object reduction),
- evaluation of the significance of sample objects,
- generation of sets of decision rules from object descriptions,
- high-yield information granulation,
- straightforward interpretation of results, and
- suitability of many of its algorithms for parallel processing.

13.3.3 Object Description

Perceptual as well as conceptual objects are known by their descriptions. An *object description* is defined by means of a tuple of function values associated with either type of object. The important thing to notice

is the paramount importance of the choice functions used to describe an object of interest.⁴ In defining what is meant by the description of an object, the focus here is on the functions that provide a basis for an object description. This can only be done by understanding the objects associated with a problem domain, such as sampling organisms by a biologist or signals from an electronic device or observations from a medical clinical study.

In combination, the functions representing object features provide a basis for an *object description* in the form of a vector $\phi : \mathcal{O} \rightarrow \mathbb{R}^L$, where \mathbb{R} is the set of real numbers, containing measurements (returned values) associated with each functional value $\phi_i(x)$ in (1).

$$\textbf{Object description: } \phi(x) = (\phi_1(x), \phi_2(x), \phi_3(x), \dots, \phi_L(x)). \quad (1)$$

Example 1. Sample object description.

By way of illustration, consider the behavior of an organism (living object) represented by a tuple

$$(s, a, r, V(s), \dots),$$

where s , a , r , and $V(s)$ denote organism functions representing state, action, reward for an action, and value of state, respectively. Typically, $V(s) \approx \sum_i r_i$, where r_i is the reward observed in state i for an action performed in state s_{i-1} . In combination, tuples of behavior function values form the following description of an object x relative to its observed behavior:

$$\textbf{Organism behaviour: } \phi(x) = (s(x), a(x), r(x), V(s(x))).$$

For example, in paper by Peters in [22], a set of objects X with observed interesting (i.e., acceptable) behavior is approximated after the set of available sample objects has been granulated using rough set approximation methods. Observed organism behavior is episodic and behavior tuples are stored in a decision table called an ethogram, where each observed behavior is assessed with an acceptability decision, i.e., $d(x) = 1$ (acceptable) and $d(x) = 0$ (unacceptable) based on evaluation of $V(s)$ for each behavior.

13.3.4 Indiscernibility and Approximation

The starting point of rough set theory is the indiscernibility relation, which is generated from information about objects of interest (see Section 13.3.1). The indiscernibility relation expresses the fact that due to a lack of information (or knowledge) we are unable to discern some objects employing available information (or knowledge). This means that, in general, we are unable to deal with each particular object but we have to consider granules (clusters) of indiscernible objects as a fundamental basis for our theory.

From a practical point of view, it is better to define basic concepts of this theory in terms of data. Therefore we will start our considerations with a tabular representation of available information about sample objects. This tabular representation is called an *information system*. Such a table contains rows labeled by objects of interest, columns labeled by functions representing object attributes, and entries of the table are function values representing object attributes. For example, an information table can describe a sample set of patients in a hospital. The patients can be characterized by some attributes such as *age*, *sex*, *blood pressure*, and *body temperature*. Every attribute is associated with a set of representative function values, e.g., *young*, *middle-aged*, and *elderly* for a function representing the attribute *age*. Attribute function values can also be numerical. In data analysis, the basic problem we are interested in is to find patterns in the function-value vectors associated with the sample objects; e.g., we might look for relationships between function values representing *blood pressure* and function values representing *age* and *sex*.

⁴ There are a number of reasons why one should distinguish between the features and attributes of objects. This is explained in detail in [74]. Consideration of this distinction is outside the scope of this chapter.

Suppose we are given a pair $\mathcal{A} = (U, A)$ of non-empty, finite sets U and A , where U is the *universe of objects* and A is a set consisting of functions *representing* attributes of objects in U . That is, $f \in A$ is a function $f : U \rightarrow V_a$, where V_a is the set of values called the *domain* of f . The pair $\mathcal{A} = (U, A)$ is called an *information system* (see, e.g., [75]). Any information system can be represented by an information table with rows labeled by objects and columns labeled by attributes. Any pair (x, f) , where $x \in U$ and $f \in A$, defines the table entry consisting of the value $f(x)$.⁵

Any subset B of A determines a binary relation \sim_B on U , called an *indiscernibility relation*, defined by

$$x \sim_B y \text{ if and only if } f(x) = f(y) \text{ for every } f \in B, \quad (2)$$

where $f(x)$ is a function value representing an attribute a for object x . The notation \sim_B closely resembles $\tilde{\sim}$ originally suggested by Zdzisław Pawlak for the indiscernibility relation in rough set theory, where attention was drawn to the role of the set B in partitioning a set U into elementary sets.⁶ The basic idea here is that the relation \sim_B provides a classification of objects according to knowledge contained in the system (U, B) , $X \subseteq U$, $B \subseteq A$ [78].

Obviously, \sim_B is an equivalence relation. The family of all equivalence classes of \sim_B , i.e., the partition determined by B , will be denoted by U/\sim_B , or simply U/B ; an equivalence class of \sim_B , i.e., a block of the partition U/B , containing x will be denoted by $B(x)$ (also denoted by $[x]_B$), where

$$B(x) = \{x' \in U \mid x' \sim_B x\}.$$

If $\in \sim_B x, y$ we will say that x and y are *B -indiscernible*. Equivalence classes of the relation \sim_B (or blocks of the partition U/B) are referred to as *B -elementary sets* or *B -elementary granules*. In the rough set approach, elementary sets are the basic building blocks of our knowledge about reality. The union of *B -elementary sets* are called *B -definable sets*.

Example 2. B -Elementary granules of perceptible objects.

For an illustration of perceived objects separated into equivalence classes, consider a set of objects \mathcal{O} and a set of functions $B \subseteq \mathcal{F}$ representing object features defined in the following way:

$$\begin{aligned} \mathcal{O} &= \{p \mid p \text{ equals a labeled conch surface cone in Figure 13.2a}\}, \\ &= \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}\}, \\ B &= \{f_1, f_2\}, \text{ where} \\ f_1 &: \mathcal{O} \rightarrow \mathbb{R}, \quad f_1(c_i) = \text{height (in mm) of conch cone } c_i \in \mathcal{O}, \\ f_2 &: \mathcal{O} \rightarrow \mathbb{R}, \quad f_2(c_i) = \text{diameter (in mm) of conch cone base } c_i \in \mathcal{O}, \\ \mathcal{O}/B &= [c_1]_B \cup [c_5]_B \cup [c_7]_B \cup [c_9]_B \text{ shown in Figure 13.2b.} \end{aligned}$$

Assume that the conical projections in the encircled areas in the conch in Figure 13.2a have matching heights and base diameters; i.e., assume $f_1(c_5) = f_1(c_6)$ and $f_2(c_5) = f_2(c_6)$. This leads to the partition of \mathcal{O} shown in Figure 13.2b. Notice, also, that $[c_1]_B$, $[c_5]_B$, $[c_7]_B$, $[c_9]_B$ constitute information granules containing conical projections with matching description relative to the functions in B .

For $B \subseteq A$, we denote by $Inf_B(x)$ the *B -signature* of $x \in U$, i.e., the set $\{(a, a(s)) : a \in A\}$. Let $INF(B) = \{Inf_B(s) : s \in U\}$. Then for any objects $x, y \in U$, the following equivalence holds: $x \sim_B y$ if and only if $Inf_B(x) = Inf_B(y)$.

⁵ Note that in statistics or machine learning such an information table is called a sample [76].

⁶ The notation $\tilde{\sim}$ denoted an equivalence relation $x \tilde{\sim} y \Leftrightarrow \rho(x, a) = \rho(y, a)$, where $\rho : A \rightarrow V$ such that $\rho_x(a) = \rho(x, a)$ and \tilde{B} denoted the relation $\bigcap_{b \in B} \tilde{\sim}$ [57, 77, 78].

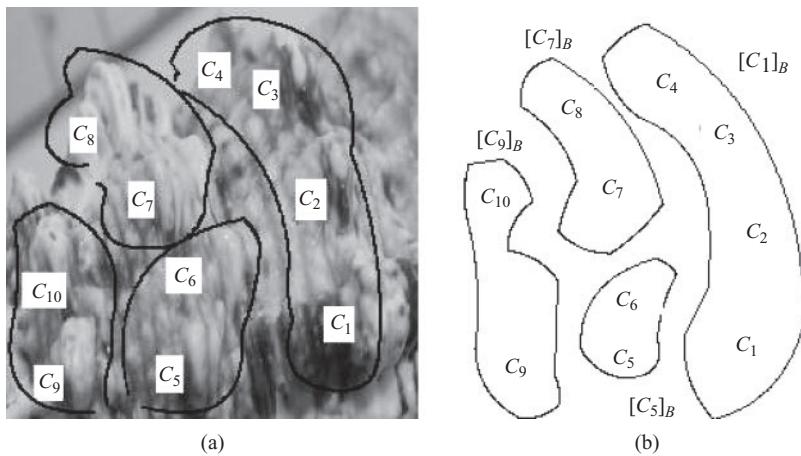


Figure 13.2 Sample partition: Conical projections in Fiji conch: (a) sample conch cones and (b) sample partition

The indiscernibility relation will be further used to define basic concepts of rough set theory. Let us now define the following two operations on sets $X \subseteq U$:

$$B_*(X) = \{x \in U : B(x) \subseteq X\}, \quad (3)$$

$$B^*(X) = \{x \in U : B(x) \cap X \neq \emptyset\}, \quad (4)$$

assigning to every subset X of the universe U two sets $B_*(X)$ and $B^*(X)$ called the *B-lower* and the *B-upper approximation* of X , respectively. The set

$$BN_B(X) = B^*(X) - B_*(X) \quad (5)$$

will be referred to as the *B-boundary region* of X .

From the definition we obtain the following interpretation:

- The *lower approximation* of a set X with respect to B is the set of all objects which can be for *certain* classified as X using B (are *certainly in* X in view of B).
- The *upper approximation* of a set X with respect to B is the set of all objects which can be *possibly* classified as X using B (are *possibly in* X in view of B).
- The *boundary region* of a set X with respect to B is the set of all objects which can be classified neither as X nor as $\text{not-}X$ using B .

Example 3. Approximation of perceptible objects.

By way of illustration of approximation of a set of perceived objects of interest and the discovery of composite information granules, consider the following approximations of the set X containing objects of interest in Figure 13.3.

The question to ask is to what extent is our knowledge about the objects in X based on the partition of available sample objects and on the choice of features in set B ? This question is answered by considering how near the lower approximation $B_*(X)$ and upper approximation $B^*(X)$ are to the set X . The nearness of these approximations to X and, by implication, the extent of the vagueness of our knowledge about X can be gauged by the size of the boundary set $BN_B(X)$ (see paper by Peters in Yao et al. [79]). Here

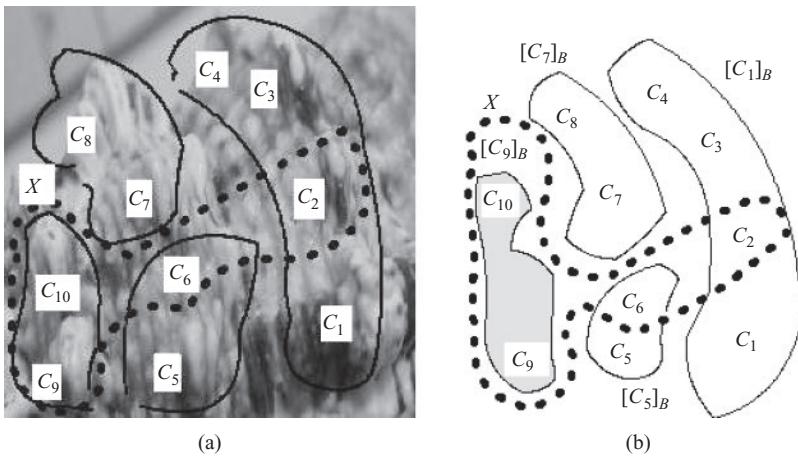


Figure 13.3 Sample approximation of conical projections: (a) selected conical projections and (b) sample set X

are the details.

$$\begin{aligned}
 \mathcal{O} &= \{c \mid c \text{ equals a labeled conical projection in Figure 13.3a}\}, \\
 &= \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}\}, \\
 X &= \{c_2, c_6, c_9, c_{10}\}, \\
 B &= \{f_1, f_2\}, \text{ defined in Example 2}, \\
 B_*(X) &= [c_9]_B, \\
 B^*(X) &= [c_1]_B \cup [c_5]_B \cup [c_9]_B, \\
 BN_B(X) &= [c_1]_B \cup [c_5]_B \text{ in Figure 13.3b}, \\
 &= \{c_1, c_2, c_3, c_4, c_5, c_6\}.
 \end{aligned}$$

In Figure 13.3b, notice that $B_*(X)$, $B^*(X)$, and $BN_B(X)$ are examples of composite information granules. Since $BN_B(X)$ is not empty, X is an example of a rough set. The size of boundary $BN_B(X)$ (it contains six objects or more than 50% of the available objects) indicates that there is a high level of vagueness associated with our knowledge about X .

In other words, due to the granularity of knowledge, rough sets cannot be characterized by using available knowledge. Therefore with every rough set we associate two *crisp* sets, called *lower* and *upper approximation*. Intuitively, the lower approximation of a set consists of all elements that *surely* belong to the set, whereas the upper approximation of the set constitutes of all elements that *possibly* belong to the set, and the *boundary region* of the set consists of all elements that cannot be classified uniquely to the set or its complement, by employing available knowledge. The approximation definition is clearly depicted in Figure 13.4.

The approximations have the following properties:

$$B_*(X) \subseteq X \subseteq B^*(X). \quad (6)$$

$$B_*(\emptyset) = B^*(\emptyset) = \emptyset, B_*(U) = B^*(U) = U.$$

$$B^*(X \cup Y) = B^*(X) \cup B^*(Y).$$

$$B_*(X \cap Y) = B_*(X) \cap B_*(Y).$$

$$X \subseteq Y \text{ implies } B_*(X) \subseteq B_*(Y) \text{ and } B^*(X) \subseteq B^*(Y).$$

$$B_*(X \cup Y) \supseteq B_*(X) \cup B_*(Y).$$

$$B^*(X \cap Y) \subseteq B^*(X) \cap B^*(Y).$$

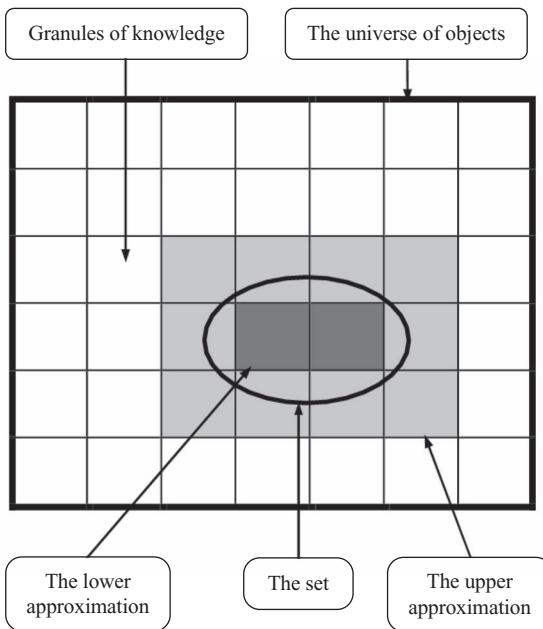


Figure 13.4 A rough set

$$\begin{aligned}
 B_*(-X) &= -B^*(X). \\
 B^*(-X) &= -B_*(X). \\
 B_*(B_*(X)) &= B^*(B_*(X)) = B_*(X). \\
 B^*(B^*(X)) &= B_*(B^*(X)) = B^*(X).
 \end{aligned}$$

Let us note that the inclusions in (6) concerning union and intersection cannot be in general substituted by the equalities. This has some important algorithmic and logical consequences.

Now we are ready to give the definition of rough sets.

If the boundary region of X is the empty set, i.e., $BN_B(X) = \emptyset$, then the set X is *crisp (exact)* with respect to B ; in the opposite case, i.e., if $BN_B(X) \neq \emptyset$, the set X is referred to as *rough (inexact)* with respect to B . Thus any rough set, in contrast to a crisp set, has a non-empty boundary region.

One can define the following four basic classes of rough sets, i.e., four categories of vagueness:

- $$\begin{aligned}
 B_*(X) \neq \emptyset \text{ and } B^*(X) \neq U, \text{ iff } X \text{ is } &\text{roughly } B\text{-definable}. \\
 B_*(X) = \emptyset \text{ and } B^*(X) \neq U, \text{ iff } X \text{ is } &\text{internally } B\text{-indefinable}. \\
 B_*(X) \neq \emptyset \text{ and } B^*(X) = U, \text{ iff } X \text{ is } &\text{externally } B\text{-indefinable}. \\
 B_*(X) = \emptyset \text{ and } B^*(X) = U, \text{ iff } X \text{ is } &\text{totally } B\text{-indefinable}.
 \end{aligned} \tag{7}$$

The intuitive meaning of this classification is the following.

If X is roughly B -definable, this means that we are able to decide for some elements of U that they belong to X and for some elements of U we are able to decide that they belong to $-X$, using B .

If X is internally B -indefinable, this means that we are able to decide for some elements of U that they belong to $-X$, but we are unable to decide for any element of U that it belongs to X , using B .

If X is externally B -indefinable, this means that we are able to decide for some elements of U that they belong to X , but we are unable to decide for any element of U that it belongs to $-X$, using B .

If X is totally B -indefinable, we are unable to decide for any element of U whether it belongs to X or $-X$, using B .

Thus a set is *rough* (imprecise) if it has non-empty boundary region; otherwise the set is *crisp* (precise). This is exactly the idea of vagueness proposed by Frege.

Let us observe that the definition of rough sets refers to data (knowledge), and is *subjective*, in contrast to the definition of classical sets, which is in some sense an *objective* one.

A rough set can also be characterized numerically by the following coefficient:

$$\alpha_B(X) = \frac{\text{card}(B_*(X))}{\text{card}(B^*(X))}, \text{ if } X \neq \emptyset; \text{ and } 1 \text{ otherwise,} \quad (8)$$

called the *accuracy of approximation*, where $\text{card}(X)$ denotes the cardinality of X .⁷ Obviously, $0 \leq \alpha_B(X) \leq 1$. If $\alpha_B(X) = 1$, then X is *crisp* with respect to B (X is *precise* with respect to B), and otherwise, if $\alpha_B(X) < 1$, then X is *rough* with respect to B (X is *vague* with respect to B). The accuracy of approximation can be used to measure the quality of approximation of decision classes on the universe U . One can use another measure of accuracy defined by $1 - \alpha_B(X)$ or by $1 - \frac{\text{card}(BN_B(X))}{\text{card}(U)}$. Some other measures of approximation accuracy are also used, e.g., based on entropy or some more specific properties of boundary regions (see, e.g., [80, 81]). The choice of a relevant accuracy of approximation depends on a particular data set. Observe that the accuracy of approximation of X can be tuned by B . Another approach to accuracy of approximation can be based on the variable-precision rough set model (VPRSM) [82].

In the next section, we discuss decision rules (constructed over a selected set B of features or a family of sets of features) which are used in inducing classification algorithms (classifiers), making it possible to classify to decision classes unseen objects. Parameters which are tuned in searching for a classifier with the high quality are its description size (defined using decision rules) and its quality of classification (measured by the number of misclassified objects on a given set of objects). By selecting a proper balance between the accuracy of classification and the description size we expect to find the classifier with the high quality of classification also on unseen objects. This approach is based on the minimal description length principle [81, 83].

13.3.5 Decision Systems and Decision Rules

Sometimes we distinguish in an information system $\mathcal{A} = (U, A)$ a partition of A into two classes $C, D \subseteq A$ of attributes, called *condition* and *decision (action)* attributes, respectively. The tuple $\mathcal{A} = (U, C, D)$ is called a *decision system*.⁸

Let $V = \bigcup\{V_a \mid a \in C\} \bigcup\{V_d \mid d \in D\}$. Atomic formulas over $B \subseteq C \cup D$ and V are expressions $a = v$ called *descriptors (selectors)* over B and V , where $a \in B$ and $v \in V_a$. The set $\mathcal{F}(B, V)$ of formulas over B and V is the least set containing all atomic formulas over B and V and closed with respect to the propositional connectives \wedge (conjunction), \vee (disjunction), and \neg (negation).

By $\|\varphi\|_{\mathcal{A}}$ we denote the meaning of $\varphi \in \mathcal{F}(B, V)$ in the decision table \mathcal{A} , which is the set of all objects in U with the property φ . These sets are defined by $\|a = v\|_{\mathcal{A}} = \{x \in U \mid a(x) = v\}$, $\|\varphi \wedge \varphi'\|_{\mathcal{A}} = \|\varphi\|_{\mathcal{A}} \cap \|\varphi'\|_{\mathcal{A}}$; $\|\varphi \vee \varphi'\|_{\mathcal{A}} = \|\varphi\|_{\mathcal{A}} \cup \|\varphi'\|_{\mathcal{A}}$; and $\|\neg\varphi\|_{\mathcal{A}} = U - \|\varphi\|_{\mathcal{A}}$. The formulas from $\mathcal{F}(C, V)$ and $\mathcal{F}(d, V)$ are called *condition formulas of \mathcal{A}* and *decision formulas of \mathcal{A}* , respectively.

Any object $x \in U$ belongs to the *decision class* $\|\bigwedge_{d \in D} d = d(x)\|_{\mathcal{A}}$ of \mathcal{A} . All decision classes of \mathcal{A} create a partition U/D of the universe U .

A *decision rule* for \mathcal{A} is any expression of the form $\varphi \Rightarrow \psi$, where $\varphi \in \mathcal{F}(C, V)$, $\psi \in \mathcal{F}(D, V)$, and $\|\varphi\|_{\mathcal{A}} \neq \emptyset$. Formulas φ and ψ are referred to as the *predecessor* and the *successor* of decision rule

⁷ $\text{card}(X)$ is also denoted by $|X|$.

⁸ Presenting decision procedures in a tabular form goes back at least to ancient Babylon. Tabular forms for computer programming dates back to the late 1950s. Next tabular forms became popular in databases (see <http://www.catalyst.com/products/logicgem/overview.html>). Some relationships of clustering in databases and rough sets are discussed in paper by Lin *et al.* in [84].

$\varphi \Rightarrow \psi$. Decision rules are often called ‘*IF . . . THEN . . .*’ rules. Such rules are used in machine learning (see, e.g., [76]).

Decision rule $\varphi \Rightarrow \psi$ is *true* in \mathcal{A} if and only if $\|\varphi\|_{\mathcal{A}} \subseteq \|\psi\|_{\mathcal{A}}$. Otherwise, one can measure its *truth degree* by introducing some inclusion measure of $\|\varphi\|_{\mathcal{A}}$ in $\|\psi\|_{\mathcal{A}}$.

Given two unary predicate formulas $\alpha(x)$ and $\beta(x)$, where x runs over a finite set U , Lukasiewicz [85] proposes to assign to $\alpha(x)$ the value $\frac{\text{card}(\{\alpha(x)\})}{\text{card}(U)}$, where $\|\alpha(x)\| = \{x \in U : x \text{ satisfies } \alpha\}$. The fractional value assigned to the implication $\alpha(x) \Rightarrow \beta(x)$ is then $\frac{\text{card}(\{\alpha(x) \wedge \beta(x)\})}{\text{card}(\{\alpha(x)\})}$ under the assumption that $\|\alpha(x)\| \neq \emptyset$. Proposed by Lukasiewicz, that fractional part was much later adapted by machine learning and data mining literature.

Each object x of a decision system determines a *decision rule*

$$\bigwedge_{a \in C} a = a(x) \Rightarrow \bigwedge_{d \in D} d = d(x). \quad (9)$$

For any decision table $\mathcal{A} = (U, C, D)$, one can consider a *generalized decision function* $\partial_A : U \longrightarrow \text{Pow}(\times_{d \in D} V_d)$ defined by

$$\partial_A(x) = \left\{ i : \exists x' \in U \left[(x', x) \in I(A) \text{ and } d(x') = i \right] \right\}, \quad (10)$$

where $\text{Pow}(V_d)$ is the powerset of the Cartesian product $\times_{d \in D} V_d$ of the family $\{V_d\}_{d \in D}$.

\mathcal{A} is called *consistent (deterministic)* if $\text{card}(\partial_A(x)) = 1$ for any $x \in U$. Otherwise \mathcal{A} is said to be *inconsistent (non-deterministic)*. Hence, a decision table is inconsistent if it consists of some objects with different decisions but indiscernible with respect to condition attributes. Any set consisting of all objects with the same generalized decision value is called a *generalized decision class*. Now, one can consider certain (possible) rules (see, e.g. paper by Grzymała-Busse in [19]) for decision classes defined by the lower (upper) approximations of such generalized decision classes of \mathcal{A} . This approach can be extended using the relationships of rough sets with the Dempster–Shafer theory by considering rules relative to decision classes defined by the lower approximations of unions of decision classes of \mathcal{A} .

Numerous methods have been developed for different decision rule generation that the reader can find in the literature on rough sets. Usually, one is searching for decision rules (semi) optimal with respect to some optimization criteria describing quality of decision rules in concept approximations.

In the case of searching for concept approximation in an extension of a given universe of objects (sample), the following steps are typical. When a set of rules has been induced from a decision table containing a set of training examples, they can be inspected to see if they reveal any novel relationships between functions representing attributes that are worth pursuing for further research. Furthermore, the rules can be applied to a set of unseen cases in order to estimate their classificatory power. For a systematic overview of rule application methods the reader is referred to the literature (see, e.g., paper by Bazan *et al.* in [15, 86]).

13.3.6 Dependency of Attributes

Another important issue in data analysis is to discover dependencies between attributes in a given decision system $\mathcal{A} = (U, C, D)$. Intuitively, a set of attributes D depends totally on a set of attributes C , denoted $C \Rightarrow D$, if the values of the functions representing attributes from C uniquely determine the values of the functions representing attributes from D . In other words, D depends totally on C if there exists a functional dependency between the function values of C and D . Hence, $C \Rightarrow D$ if and only if the rule (9) is true on \mathcal{A} for any $x \in U$. D can depend partially on C . Formally such a dependency can be defined in the following way.

We will say that D depends on C to a degree k ($0 \leq k \leq 1$), denoted $C \Rightarrow_k D$, if

$$k = \gamma(C, D) = \frac{\text{card}(POS_C(D))}{\text{card}(U)}, \quad (11)$$

where

$$POS_C(D) = \bigcup_{X \in U/D} C_*(X), \quad (12)$$

called a *positive region* of the partition U/D with respect to C , is the set of all elements of U that can be uniquely classified to blocks of the partition U/D , by means of C .

If $k = 1$, we say that D depends totally on C , and if $k < 1$, we say that D depends partially (to degree k) on C . If $k = 0$, then the *positive region* of the partition U/D with respect to C is empty.

The coefficient k expresses the ratio of all elements of the universe, which can be properly classified to blocks of the partition U/D , employing attributes C and will be called the *degree of the dependency*.

It can be easily seen that if D depends totally on C , then $I(C) \subseteq I(D)$. It means that the partition generated by C is finer than the partition generated by D . Notice that the concept of dependency discussed above corresponds to that considered in relational databases.

Summing up, D is *totally (partially)* dependent on C if *all (some)* elements of the universe U can be uniquely classified to blocks of the partition U/D , employing C .

Observe that (11) defines only one of possible measures of dependency between attributes (see, e.g., [87]). One can also compare the dependency discussed in this section with dependencies considered in databases.

In this section, we have considered granules being partitions and some measures of inclusion between such granules.

13.3.7 Reduction of Attributes

We often face a question whether we can remove some data from a data table preserving its basic properties, i.e., whether a table contains some superfluous data.

Let us express this idea more precisely.

Let $C, D \subseteq A$ be sets of condition and decision attributes respectively. We will say that $C' \subseteq C$ is a *D-reduct* (reduct with respect to D) of C if C' is a minimal subset of C such that

$$\gamma(C, D) = \gamma(C', D). \quad (13)$$

The intersection of all D -reducts is called a *D-core* (core with respect to D). Because the core is the intersection of all reducts, it is included in every reduct; i.e., each element of the core belongs to some reduct. Thus, in a sense, the core is the most important subset of attributes, since none of its elements can be removed without affecting the classification power of attributes. Certainly, the geometry of reducts can be more compound. For example, the core can be empty but there can exist a partition of reducts into a few sets with non-empty intersection.

Many other kinds of reducts and their approximations are discussed in the literature (see, e.g., papers by Bazan in [16], by Nguyen [88], by Nguyen *et al.* in [43], and in [81, 89, 90]). For example, if one change the condition (13) to $\partial_A(x) = \partial_B(x)$, then the defined reducts are preserving the generalized decision. Other kinds of reducts are preserving: (i) the distance between attribute value vectors for any two objects if this distance is greater than a given threshold [89]; (ii) the distance between entropy distributions between any two objects if this distance exceeds a given threshold [90]; or (iii) the so-called reducts relative to object used for generation of decision rules (see, e.g., paper by Bazan in [16]). There are some relationships between different kinds of reducts. If B is a reduct preserving the generalized decision, than in B is included a reduct preserving the positive region. For mentioned above reducts based on distances and thresholds one can find analogous dependency between reducts relative to different thresholds. By choosing different kinds of reducts we select different degrees to which information encoded in data is preserved. Reducts are used for building data models. Choosing a particular reduct or a set of reducts has impact on the model size as well as on its quality in describing a given data set. The model size together with the model quality are two basic components tuned in selecting relevant data models. This is known as the minimal length principle (see, e.g., [81, 83]). Selection of relevant kinds of reducts is an important

step in building data models. It turns out that the different kinds of reducts can be efficiently computed using heuristics based, e.g., on the Boolean reasoning approach [88, 91].

13.3.8 Discernibility and Boolean Reasoning

Methodologies devoted to data mining, knowledge discovery, decision support, pattern classification, and approximate reasoning require tools for discovering *templates (patterns)* in data and classifying them into certain *decision classes*. Templates are in many cases most frequent sequences of events, most probable events, regular configurations of objects, the decision rules of high quality, and standard reasoning schemes. Tools for discovering and classifying templates are based on *reasoning schemes* rooted in various paradigms [63]. Such patterns can be extracted from data by means of methods based, e.g., on Boolean reasoning and discernibility.

The discernibility relations are closely related to indiscernibility and belong to the most important relations considered in rough set theory.

The ability to discern between perceived objects is important for constructing many entities like reducts, decision rules, or decision algorithms. In the classical rough set approach, the discernibility relation $DIS(B) \subseteq U \times U$ is defined by $xDIS(B)y$ if and only if $\text{non}(x \sim_B y)$. However, this is, in general, not the case for the generalized approximation spaces.

The idea of Boolean reasoning is based on construction for a given problem P of a corresponding Boolean function f_P with the following property: the solutions for the problem P can be decoded from prime implicants of the Boolean function f_P . Let us mention that to solve real-life problems it is necessary to deal with Boolean functions having large number of variables.

A successful methodology based on the discernibility of objects and Boolean reasoning has been developed for computing many important ingredients for applications. These applications include generation of reducts and their approximations, decision rules, association rules, discretization of real value attributes, symbolic value grouping, searching for new features defined by oblique hyperplanes or higher order surfaces, pattern extraction from data, as well as conflict resolution or negotiation.

Most of the problems related to generation of the above-mentioned entities are NP-complete or NP-hard. However, it was possible to develop efficient heuristics returning suboptimal solutions of the problems. The results of experiments on many data sets show very good quality of solutions generated by the heuristics in comparison with other methods reported in literature (e.g., with respect to the classification quality of unseen objects). Moreover, they are very efficient from the point of view of time necessary for computing the solution. Many of these methods are based on discernibility matrices. Note that it is possible to compute the necessary information about these matrices using directly⁹ information or decision systems (e.g., sorted in preprocessing, see [79, 88, 92] and paper by Bazan *et al.* in [15]), which significantly improves the efficiency of algorithms.

It is important to note that the methodology makes it possible to construct heuristics having a very important *approximation property* which can be formulated as follows: expressions generated by heuristics (i.e., implicants) *close* to prime implicants define approximate solutions for the problem.

In supervised machine learning paradigm [64, 76, 86], a learning algorithm is given a training data set, usually in the form of a decision system $\mathcal{A} = (U, A, d)$,¹⁰ prepared by an expert. Every such decision system classifies elements from U into decision classes. The purpose of the algorithm is to return a set of decision rules together with matching procedure and conflict resolution strategy, called a classifier, which makes it possible to classify unseen objects, i.e., objects that are not described in the original decision table. Several rough set methods were developed for construction of classifiers. For more information the reader is referred, e.g., to references in [28], and for papers on hierarchical learning and ontology approximation, e.g., to papers by Bazan *et al.* in [46], by S.H. Nguyen *et al.* [21], by T. T. Nguyen in [93], and by Skowron *et al.* in [11, 20, 50] and for papers in this handbook.

⁹ That is, without the necessity of generation and storing of the discernibility matrices.

¹⁰ For simplicity, we consider decision systems with one decision.

Many of these methods are based on computing prime implicants for computing different kinds of reducts. Unfortunately, they are computationally hard. However, many heuristics have been developed, which turned out to be very promising. The results of experiments on many data sets, reported in the literature, show a very good quality of classification of unseen objects using these heuristics. A variety of methods for computing reducts and their applications can be found in the literature (see, e.g., references in [28]).

The fact that the problem of finding a minimal reduct of a given information system is NP-hard was proved in [94].

To summarize, there exists a number of good heuristics that compute sufficiently many reducts in an acceptable time. Moreover, a successful methodology, based on different reducts, has been developed for solution of many problems like attribute selection, decision rule generation, association rule generation, discretization of real-valued attributes, and symbolic value grouping. For further readings the reader is referred to the bibliography in [28–30].

Many of these methods are based on discernibility matrices [88, 94]. It is possible to compute the necessary information about these matrices using information or decision systems directly what significantly improves the efficiency of algorithms.

The results based on Boolean reasoning have been implemented, e.g., in the RSES and ROSETTA software systems (see <http://logic.mimuw.edu.pl/~rses/> for RSES and <http://rosetta.lcb.uu.se/general/> for ROSETTA; see also the bibliography in [28–30]). For links to other rough set software systems, the reader is referred to <http://rsds.wsiz.rzeszow.pl>.

13.3.9 Rough Membership

Let us observe that rough sets can also be defined employing the rough membership function (see equation 14) [95]. That is, consider

$$\mu_X^B : U \rightarrow <0, 1>,$$

defined by

$$\mu_X^B(x) = \frac{\text{card}(B(x) \cap X)}{\text{card}(X)}, \quad (14)$$

where $x \in X \subseteq U$.

The value $\mu_X^B(x)$ can be interpreted as the degree that x belongs to X in view of knowledge about x expressed by B or the degree to which the elementary granule $B(x)$ is included in the set X . This means that the definition reflects a subjective knowledge about elements of the universe, in contrast to the classical definition of a set.

The rough membership function can also be interpreted as the conditional probability that x belongs to X given B . This interpretation was used by several researchers in the rough set community (see, e.g., the bibliography in [28–30]). Note also that the ratio on the right-hand side of the equation (14) is known as the confidence coefficient in data mining [64, 76]. It is worthwhile to mention that set inclusion to a degree has been considered by Łukasiewicz [85] in studies on assigning fractional truth values to logical formulas.

It can be shown that the rough membership function has the following properties [95]:

- (1) $\mu_X^B(x) = 1$ iff $x \in B_*(X)$.
- (2) $\mu_X^B(x) = 0$ iff $x \in U - B^*(X)$.
- (3) $0 < \mu_X^B(x) < 1$ iff $x \in BN_B(X)$.
- (4) $\mu_{U-X}^B(x) = 1 - \mu_X^B(x)$ for any $x \in U$.
- (5) $\mu_{X \cup Y}^B(x) \geq \max(\mu_X^B(x), \mu_Y^B(x))$ for any $x \in U$.
- (6) $\mu_{X \cap Y}^B(x) \leq \min(\mu_X^B(x), \mu_Y^B(x))$ for any $x \in U$.

From the properties it follows that the rough membership differs essentially from the fuzzy membership [96], for properties (5) and (6) show that the membership for union and intersection of sets, in general, cannot be computed – as in the case of fuzzy sets – from their constituents membership. Thus, formally, the rough membership is more general than fuzzy membership. Moreover, the rough membership function depends on an available knowledge (represented by attributes from B). Besides, the rough membership function, in contrast to fuzzy membership function, has a probabilistic flavor.

Let us also mention that rough set theory, in contrast to fuzzy set theory, clearly distinguishes two very important concepts, vagueness and uncertainty, very often confused in the artificial intelligence (AI) literature. Vagueness is the property of sets and can be described by approximations, whereas uncertainty is the property of elements of a set and can be expressed by the rough membership function.

Both fuzzy and rough set theories represent two different approaches to vagueness. Fuzzy set theory addresses *gradualness* of knowledge, expressed by the fuzzy membership, whereas rough set theory addresses *granularity* of knowledge, expressed by the indiscernibility relation. A nice illustration of this difference has been given by Dider Dubois and Henri Prade [97] in the following example. In image processing, fuzzy set theory refers to gradualness of gray level, whereas rough set theory is about the size of pixels.

Consequently, both theories are not competing but are rather complementary. In particular, the rough set approach provides tools for approximate construction of fuzzy membership functions. The rough-fuzzy hybridization approach proved to be successful in many applications (see, e.g., [12, 98]).

An interesting discussion of fuzzy and rough set theory in the approach to vagueness can be found in [71]. Let us also observe that fuzzy set and rough set theories are not a remedy for classical set theory difficulties.

One of the consequences of perceiving objects by information about them is that for some objects one cannot decide if they belong to a given set or not. However, one can estimate the degree to which objects belong to sets. This is a crucial observation in building foundations for approximate reasoning. Dealing with imperfect knowledge implies that one can only characterize satisfiability of relations between objects to a degree, not precisely. One of the fundamental relations on objects is a rough inclusion relation describing that objects are parts of other objects to a degree. The rough mereological approach (see, e.g., papers by Polkowski *et al.* in [11, 15, 16, 68]) based on such a relation is an extension of the Leśniewski mereology [66].

13.4 Conflicts

Knowledge discovery in databases considered in the previous sections reduces to searching for functional dependencies in the data set. In this section, we will discuss another kind of relationship in the data – not dependencies, but conflicts. Formally, the conflict relation can be seen as a negation (not necessarily classical) of indiscernibility relation, which was used as a basis of rough set theory. Thus indiscernibility and conflict are closely related from logical point of view. It turns out that the conflict relation can be used to the conflict analysis study. Conflict analysis and resolution play an important role in business, governmental, political, and lawsuits disputes, labor-management negotiations, military operations, and others. To this end, many mathematical formal models of conflict situations have been proposed and studied (for references to the bibliography, see [28–30]). Various mathematical tools, e.g., graph theory, topology, and differential equations, have been used for that purpose. Needless to say, game theory can also be considered as a mathematical model of conflict situations. In fact, there is no “universal” theory of conflicts yet, and mathematical models of conflict situations are strongly domain dependent. In the following section we outline yet another approach to conflict analysis – based on some ideas of rough set theory – along the lines of [99]. The considered model is simple enough for easy computer implementation and seems to be adequate for many real-life applications.

13.4.1 Basic Concepts of Conflict Theory

In this section, we give definitions of basic concepts of the proposed approach in lines with [99].

Let us assume that we are given a finite, non-empty set Ag called the *universe*. Elements of Ag will be referred to as *agents*. Let a *voting function* $v : Ag \rightarrow \{-1, 0, 1\}$ be given, assigning to every agent one of the number -1 , 0 , or 1 , which represents his opinion, view, voting result, etc., about some discussed issue and means *against*, *neutral*, and *favorable*, respectively.

Voting functions correspond to situations. Hence, let us assume that there is given a set of situations U and a set of voting functions *Voting-Fun*, as well as a *conflict function* $Conflict : U \longrightarrow Voting_Fun$. Any pair $S = (s, v)$ where $s \in U$ and $v = Conflict(s)$ will be called a *conflict situation*.

In order to express relations between agents from Ag , defined by a given voting function v , we define three basic binary relations in Ag^2 : *conflict*, *neutrality*, and *alliance*.

To this end, we first define the following auxiliary function:

$$\phi_v(ag, ag') = \begin{cases} 1, & \text{if } v(ag)v(ag') = 1 \text{ or } v(ag) = v(ag') = 0 \\ 0, & \text{if } v(ag)v(ag') = 0 \text{ and } non(v(ag)) = v(ag') = 0 \\ -1, & \text{if } v(ag)v(ag') = -1. \end{cases} \quad (15)$$

This means that if $\phi_v(ag, ag') = 1$, then agents ag and ag' have the same opinion about issue v (are *allied* on v); $\phi_v(ag, ag') = 0$ means that one of agents ag or ag' has neutral approach to issue v (is *neutral* on v), and $\phi_v(ag, ag') = -1$, means that both agents have different opinions about issue v (are in *conflict* on v).

In what follows we will define three basic binary relations $R_v^+, R_v^0, R_v^- \subseteq Ag^2$ called *alliance*, *neutrality*, and *conflict* relations, respectively, and defined by

$$\begin{aligned} R_v^+(ag, ag') &\text{ iff } \phi_v(ag, ag') = 1; \\ R_v^0(ag, ag') &\text{ iff } \phi_v(ag, ag') = 0; \\ R_v^-(ag, ag') &\text{ iff } \phi_v(ag, ag') = -1. \end{aligned} \quad (16)$$

It is easily seen that the alliance relation has the following properties:

$$\begin{aligned} R_v^+(ag, ag). \\ R_v^+(ag, ag'). &\text{ implies } R_v^+(ag', ag). \\ R_v^+(ag, ag') \text{ and } R_v^+(ag', ag'') &\text{ imply } R_v^+(ag, ag''); \end{aligned} \quad (17)$$

i.e., R_v^+ is an *equivalence* relation. Each equivalence class of alliance relation will be called a *coalition* with respect to v . Let us note that the last condition in (17) can be expressed as ‘a friend of my friend is my friend.’

For the conflict relation we have the following properties:

$$\begin{aligned} \text{Not } R_v^-(ag, ag). \\ R_v^-(ag, ag') &\text{ implies } R_v^-(ag', ag). \\ R_v^-(ag, ag') \text{ and } R_v^-(ag', ag'') &\text{ imply } R_v^+(ag, ag''). \\ R_v^-(ag, ag') \text{ and } R_v^+(ag', ag'') &\text{ imply } R_v^-(ag, ag''). \end{aligned} \quad (18)$$

The last two conditions in (18) refer to well-known sayings ‘an enemy of my enemy is my friend’ and ‘a friend of my enemy is my enemy.’

For the neutrality relation, we have

$$\begin{aligned} \text{Not } R_v^0(ag, ag) \\ R_v^0(ag, ag') = R_v^0(ag', ag). \end{aligned} \quad (19)$$

Let us observe that there are no coalitions in the conflict and neutrality relations.

We have $R_v^+ \cup R_v^0 \cup R_v^- = Ag^2$ because for any pair of agents $(ag, ag') \in Ag^2$, $\Phi_v(ag, ag') = 1$ or $\Phi_v(ag, ag') = 0$ or $\Phi_v(ag, ag') = -1$, so $(ag, ag') \in R_v^+$ or $(ag, ag') \in R_v^0$ or $(ag, ag') \in R_v^-$. All the three relations R_v^+ , R_v^0 , and R_v^- are pairwise disjoint; i.e., every pair of objects (ag, ag') belongs to exactly one of relation (is in conflict, is allied, or is neutral).

With every conflict situation $S = (s, v)$ we will associate a *conflict graph*

$$G_S = (R_v^+, R_v^0, R_v^-). \quad (20)$$

A conflict degree $Con(S)$ of the conflict situation $S = (s, v)$ is defined by

$$Con(S) = \frac{\sum_{\{(ag, ag'): \Phi_v(ag, ag') = -1\}} |\Phi_v(ag, ag')|}{2 \lceil \frac{n}{2} \rceil \times (n - \lceil \frac{n}{2} \rceil)}, \quad (21)$$

where $n = \text{card}(Ag)$.

One can consider a more general case of a conflict function, viz., a mapping of the form $\text{Conflict} : U \longrightarrow \text{Voting_Fun}^k$, where k is a positive integer. Then, a conflict situation is any pair $S = (s, (v_1, \dots, v_k))$, where $(v_1, \dots, v_k) = \text{Conflict}(s)$, and the conflict degree in S can be defined by

$$Con(S) = \frac{\sum_{i=1}^k Con(S_i)}{k}, \quad (22)$$

where $S_i = (s, v_i)$ for $i = 1, \dots, k$. Each function v_i is called a voting function on the i th issue in s .

13.4.2 Conflicts and Rough Sets

There are strong relationships between the approach to conflicts presented in Section 13.4.1 and the rough set approach. In this section, we discuss examples of such relationships. The approach presented in this section seems to be very promising for solving problems related to conflict resolution and negotiations (for references see, e.g., the bibliography in [28–30]).

The application of rough sets can bring new results in the area related to conflict resolution and negotiations between agents because this makes it possible to introduce approximate reasoning about vague concepts into the area.

Let us outline this possibility. First, let us observe that any conflict situation $S = (s, V)$, where $V = (v_1, \dots, v_k)$ and each v_i is defined on the set of agents $Ag = \{ag_1, \dots, ag_n\}$, can be treated as an information system $\mathcal{A}(S)$ with the set of objects Ag and the set of attributes $\{v_1, \dots, v_k\}$. The discernibility degree between agents ag and ag' in S can be defined by

$$disc_S(ag, ag') = \frac{\sum_{\{i: \Phi_{v_i}(ag, ag') = -1\}} |\Phi_{v_i}(ag, ag')|}{k}, \quad (23)$$

where $ag, ag' \in Ag$ and $|\cdot|$ denotes the absolute value.

Now, one can consider reducts of $\mathcal{A}(S)$ relative to the discernibility degrees defined by $disc_S$. For example, one can consider agents ag and ag' as discernible if

$$disc_S(ag, ag') \geq tr,$$

where tr is a given threshold.¹¹ Any reduct $R \subseteq V$ of S is a minimal set of voting functions preserving on R to a degree at least tr the discernibility in voting between any two agents which are discernible on V to degree at least tr . All voting functions from $V - R$ are dispensable with respect to such a preservation of discernibility degrees between agents.

¹¹ To compute such reducts one can follow the method presented in [94], assuming that any entry of the discernibility matrix corresponding to (ag, ag') with $disc(ag, ag') < tr$ is empty and the remaining entries are families of all minimal subsets of V on which the discernibility between (ag, ag') is at least equal to tr [100].

Reducts of the information system $\mathcal{A}_T(S)$ with the universe of objects equal to $\{v_1, \dots, v_k\}$ and attributes defined by agents and voting functions by $ag(v) = v(ag)$, for $ag \in Ag$ and $v \in V$, can be considered in an analogous way. The discernibility degree between voting functions can be defined, e.g., by

$$disc(v, v') = |Con(S_v) - Con(S_{v'}|, \quad (24)$$

and it can be used to measure the difference between voting functions v and v' .

Any reduct R of $\mathcal{A}_T(S)$ is a minimal set of agents preserving on R to a degree at least tr the discernibility degree between any two voting functions which are discernible on Ag to degree at least tr .

In our next example, we extend the model of conflict by adding a set A of (condition) attributes used to describe the situations in terms of values of attributes from A . The set of given situations is denoted by U . In this way, we have defined an information system (U, A) . Let us assume that there is also given a set of agents Ag . Each agent $ag \in Ag$ has access to a subset $A_{ag} \subseteq A$ of condition attributes. Moreover, we assume that $A = \bigcup_{ag \in Ag} A_{ag}$. We assume that there is also defined a decision attribute d on U such that $d(s)$ is a conflict situation $S = (s, V)$, where $V = (v_1, \dots, v_k)$. Observe that $S = (s, V)$ can be represented by a matrix

$$[v_i(ag_j)]_{i=1,\dots,n; j=1,\dots,k},$$

where $v_i(ag_j)$ is the result of voting by j th agent on the i th issue. Such a matrix is a compound decision¹² corresponding to s . For the constructed decision system (U, A, d) one can use, e.g., the function given by (22) to measure conflict degrees of situations from U . Let us mention one more kind of reducts which have a natural interpretation in conflict analysis. Such reducts preserve the discernibility between any two A -discernible situations such that the absolute value of the difference between corresponding to them conflict degrees is at least equal tr , where tr is a given threshold.

The described decision table can also be used in conflict resolution. We would like to illustrate this possibility. First, let us recall some notation. For $B \subseteq A$, we denote by $Inf_B(s)$ the B -signature of the situation s , i.e., the set $\{(a, a(s)) : a \in A\}$. Let $INF(B) = \{Inf_B(s) : s \in U\}$. Let us also assume that for each agent $ag \in Ag$, there is given a similarity relation $\tau_{ag} \subseteq INF(A_{ag}) \times INF(A_{ag})$. In terms of these similarity relations one can consider the problem of conflict resolution relative to a given threshold tr in a given situation s described by $Inf_A(s)$. This is a searching problem for a situation s' , if such a situation exists, satisfying the following conditions:

1. $Inf_A(s')|A_{ag} \in \tau_{ag}(Inf_{A_{ag}}(s))$, where $\tau_{ag}(Inf_{A_{ag}}(s))$ is the tolerance class of $Inf_{A_{ag}}(s)$ with respect to τ_{ag} and $Inf_A(s')|A_{ag}$ denotes the restriction of $Inf_A(s')$ to A_{ag} .
2. $Inf_A(s')$ satisfies given local constraints (e.g., specifying coexistence of local situations [100]; paper by Suraj in [15]) and given global constraints (e.g., specifying quality of global situations [100]).
3. The conflict degree in the conflict situation $d(s')$,¹³ measured by means of the chosen conflict measure,¹⁴ is at most tr .

In searching for conflict resolution one can apply methods based on Boolean reasoning and [100].

We have proposed changes in the acceptability by agents to be expressed by similarity relations. Observe that in real-life applications, these similarities are more compound than it was suggested above; i.e., they are not defined directly by sensory concepts describing situations. However, they are often specified by high-level concepts (see, e.g., [29, 101]). These high-level concepts can be vague, and they are linked with the sensory concepts describing situations by means of a hierarchy of other vague concepts. Approximation of vague concepts in the hierarchy and dependencies between them (see [29])

¹² For references to other papers on compound decision the reader is referred, e.g., to the paper by Bazan *et al.* in [39].

¹³ Let us observe that s' is not necessarily in U . In such a case, the value $d(s')$ should be predicted by the classifier induced from (U, A, d) .

¹⁴ For example, one can consider (22).

makes it possible to approximate the similarity relations. This allows us to develop searching methods for acceptable value changes of sensory concepts preserving similarities (constraints) specified over high-level vague concepts. One can also introduce some costs of changes of local situations by agents and search for new situations accessible under minimal or subminimal costs.

Using the rough set approach to conflict resolution and negotiations between agents, one can also consider more advanced models in which actions and plans performed by agents or their teams are involved in negotiations and conflict resolution. This is one of many interesting directions for further research study of the relationships between rough sets and conflicts.

13.5 Extensions

The rough set concept can be defined quite generally by means of topological operations, *interior* and *closure*, called *approximations* [14]. It was observed in [59] that the key to the presented approach is provided by the exact mathematical formulation of the concept of approximative (rough) equality of sets in a given approximation space. In [13], an approximation space is represented by the pair (U, R) , where U is a universe of objects and $R \subseteq U \times U$ is an indiscernibility relation defined by an attribute set (i.e., $R = I(A)$ for some attribute set A). In this case R is the equivalence relation. Let $[x]_R$ denote an equivalence class of an element $x \in U$ under the indiscernibility relation R , where $[x]_R = \{y \in U : xRy\}$.

In this context, R -approximations of any set $X \subseteq U$ are based on the exact (crisp) containment of sets. Then set approximations are defined as follows:

- $x \in U$ belongs with certainty to $X \subseteq U$ (i.e., x belongs to the R -lower approximation of X) if $[x]_R \subseteq X$.
- $x \in U$ possibly belongs to $X \subseteq U$ (i.e., x belongs to the R -upper approximation of X) if $[x]_R \cap X \neq \emptyset$.
- $x \in U$ belongs with certainty neither to the X nor to $U - X$ (i.e., x belongs to the R -boundary region of X) if $[x]_R \cap (U - X) \neq \emptyset$ and $[x]_R \cap X \neq \emptyset$.

Several generalizations of the above approach have been proposed in the literature (for references see, e.g., the bibliography in [28–30]).

In particular, in some of these approaches, set inclusion to a degree is used instead of the exact inclusion.

Different aspects of vagueness in the rough set framework are discussed (see, e.g., paper by Marcus in [42, 70, 71]).

Our knowledge about the approximated concepts is often partial and uncertain [5]. For example, concept approximation should be constructed from examples and counterexamples of objects for the concepts [76]. Hence, concept approximations constructed from a given sample of objects are extended, using inductive reasoning, on objects not yet observed. The rough set approach for dealing with concept approximation under such partial knowledge is presented, e.g., in [102]. Moreover, the concept approximations should be constructed under dynamically changing environments [70]. This leads to a more complex situation where the boundary regions are not crisp sets, which is consistent with the postulate of the higher order vagueness considered by philosophers (see, e.g., [69]). It is worthwhile to mention that a rough set approach to the approximation of compound concepts has been developed and at this time no traditional method is able to directly approximate compound concepts [103, 104]. The approach is based on hierarchical learning and ontology approximation (see, e.g., paper by Bazan *et al.* [46], by Nguyen *et al.* in [21], and by Skowron *et al.* in [11, 50]). Approximation of concepts in distributed environments is discussed in a paper by Skowron in [20]. A survey of algorithmic methods for concept approximation based on rough sets and Boolean reasoning is presented, e.g., in papers by Bazan *et al.* [15] and in [105].

13.5.1 Generalizations of Approximation Spaces

Several generalizations of the classical rough set approach based on approximation spaces defined as pairs of the form (U, R) , where R is the equivalence relation (called indiscernibility relation) on the set U , have been reported in the literature. Let us mention two of them.

A generalized approximation space¹⁵ can be defined by a tuple $AS = (U, I, \nu)$, where I is the *uncertainty function* defined on U with values in the powerset $\text{Pow}(U)$ of U ($I(x)$ is the *neighbourhood* of x) and ν is the *inclusion function* defined on the Cartesian product $\text{Pow}(U) \times \text{Pow}(U)$ with values in the interval $[0, 1]$ measuring the degree of inclusion of sets. The lower AS_* and upper AS^* approximation operations can be defined in AS by

$$AS_*(X) = \{x \in U : \nu(I(x), X) = 1\}, \quad (25)$$

$$AS^*(X) = \{x \in U : \nu(I(x), X) > 0\}. \quad (26)$$

In the standard case, $I(x)$ is equal to the equivalence class $B(x)$ of the indiscernibility relation $I(B)$; in case of tolerance (similarity) relation $\tau \subseteq U \times U$, we take $I(x) = \{y \in U : x \tau y\}$; i.e., $I(x)$ is equal to the tolerance class of τ defined by x . The standard inclusion relation ν_{SRI} is defined for $X, Y \subseteq U$ by

$$\nu_{SRI}(X, Y) = \begin{cases} \frac{\text{card}(X \cap Y)}{\text{card}(X)}, & \text{if } X \neq \emptyset, \\ 1, & \text{otherwise.} \end{cases} \quad (27)$$

For applications it is important to have some constructive definitions of I and ν .

One can consider another way to define $I(x)$. Usually together with AS we consider some set F of formulas describing sets of objects in the universe U of AS defined by semantics $\|\cdot\|_{AS}$, i.e., $\|\alpha\|_{AS} \subseteq U$ for any $\alpha \in F$. Now, one can take the set

$$N_F(x) = \{\alpha \in F : x \in \|\alpha\|_{AS}\}, \quad (28)$$

and $I(x) = \{\|\alpha\|_{AS} : \alpha \in N_F(x)\}$. Hence, more general uncertainty functions having values in $\text{Pow}(\text{Pow}(U))$ can be defined and in the consequence different definitions of approximations are considered. For example, one can consider the following definitions of approximation operations in AS :

$$AS_\circ(X) = \{x \in U : \nu(Y, X) = 1 \text{ for some } Y \in I(x)\}, \quad (29)$$

$$AS^\circ(X) = \{x \in U : \nu(Y, X) > 0 \text{ for any } Y \in I(x)\}. \quad (30)$$

There are also different forms of rough inclusion functions. Let us consider two examples.

In the first example of a rough inclusion function, a threshold $t \in (0, 0.5)$ is used to relax the degree of inclusion of sets. The rough inclusion function ν_t is defined by

$$\nu_t(X, Y) = \begin{cases} 1, & \text{if } \nu_{SRI}(X, Y) \geq 1 - t, \\ \frac{\nu_{SRI}(X, Y) - t}{1 - 2t}, & \text{if } t \leq \nu_{SRI}(X, Y) < 1 - t, \\ 0, & \text{if } \nu_{SRI}(X, Y) \leq t. \end{cases} \quad (31)$$

This is an interesting ‘rough-fuzzy’ example because we put the standard rough membership function as an argument into the formula often used for fuzzy membership functions.

One can obtain approximations considered in the variable-precision rough set approach (VPRSM) [82] by substituting in (25)–(26) the rough inclusion function ν_t defined by (31) instead of ν , assuming that Y is a decision class and $N(x) = B(x)$ for any object x , where B is a given set of attributes.

Another example of application of the standard inclusion was developed by using probabilistic decision functions. For more detail the reader is referred to papers by Ślęzak *et al.* in [3, 15] and paper [87].

The rough inclusion relation can also be used for function approximation and relation approximation [102]. In the case of function approximation the inclusion function ν^* for subsets $X, Y \subseteq U \times U$, where

¹⁵ Some other generalizations of approximation spaces are also considered in the literature (for references see, e.g., the bibliography in [28–30]).

$X, Y \subseteq \mathcal{R}$ and \mathcal{R} is the set of reals, is defined by

$$\nu^*(X, Y) = \begin{cases} \frac{\text{card}(\pi_1(X \cap Y))}{\text{card}(\pi_1(X))}, & \text{if } \pi_1(X) \neq \emptyset, \\ 1, & \text{if } \pi_1(X) = \emptyset, \end{cases} \quad (32)$$

where π_1 is the projection operation on the first coordinate. Assume now that X is a cube and Y is the graph $G(f)$ of the function $f : \mathcal{R} \rightarrow \mathcal{R}$. Then, e.g., X is in the lower approximation of f if the projection on the first coordinate of the intersection $X \cap G(f)$ is equal to the projection of X on the first coordinate. This means that the part of the graph $G(f)$ is ‘well’ included in the box X ; i.e., for all arguments that belong to the box projection on the first coordinate, the value of f is included in the box X projection on the second coordinate.

The approach based on inclusion functions has been generalized to the *rough mereological approach* (see, e.g., papers by Polkowski *et al.* in [11, 15, 16, 68] and also Section 13.6.2). The inclusion relation $x \mu_r y$ with the intended meaning x is a part of y to a degree at least r has been taken as the basic notion of the rough mereology being a generalization of the Leśniewski mereology [66]. Research on rough mereology has shown importance of another notion, namely, *closeness* of compound objects (e.g., concepts). This can be defined by $xcl_{r,r'}y$ if and only if $x \mu_r y$ and $y \mu_{r'}x$.

Rough mereology offers a methodology for synthesis and analysis of objects in a distributed environment of intelligent agents, in particular, for synthesis of objects satisfying a given specification to a satisfactory degree or for control in such a complex environment. Moreover, rough mereology has been used for developing the foundations of the *information granule calculi*, aiming at formalization of the computing with words paradigm, recently formulated by Lotfi Zadeh [60]. More complex information granules are defined recursively, using already-defined information granules and their measures of inclusion and closeness. Information granules can have complex structures like classifiers or approximation spaces. Computations on information granules are performed to discover relevant information granules, e.g., patterns or approximation spaces for compound concept approximations.

Usually there are considered families of approximation spaces labeled by some parameters. By tuning such parameters according to chosen criteria (e.g., minimal description length) one can search for the optimal approximation space for concept description (see, e.g., paper by Bazan *et al.* in [15]).

13.5.2 Concept Approximation

In this section we consider the problem of approximation of concepts over a universe U^∞ (concepts that are subsets of U^∞). We assume that the concepts are perceived only through some subsets of U^∞ , called samples. This is a typical situation in the machine learning, pattern recognition, or data mining approaches [64, 76, 106]. We explain the rough set approach to induction of concept approximations using the generalized approximation spaces of the form $AS = (U, I, \nu)$ defined in Section 13.5.1.

Let $U \subseteq U^\infty$ be a finite sample. By Π_U we denote a perception function from $P(U^\infty)$ into $P(U)$ defined by $\Pi_U(C) = C \cap U$ for any concept $C \subseteq U^\infty$. Let $AS = (U, I, \nu)$ be an approximation space over the sample U .

The problem we consider is how to extend the approximations of $\Pi_U(C)$ defined by AS to approximation of C over U^∞ . We show that the problem can be described as searching for an extension $AS_C = (U^\infty, I_C, \nu_C)$ of the approximation space AS , relevant for approximation of C . This requires to show how to extend the inclusion function ν from subsets of U to subsets of U^∞ that are relevant for the approximation of C . Observe that for the approximation of C it is enough to induce the necessary values of the inclusion function ν_C without knowing the exact value of $I_C(x) \subseteq U^\infty$ for $x \in U^\infty$.

Let AS be a given approximation space for $\Pi_U(C)$ and let us consider a language L in which the neighborhood $I(x) \subseteq U$ is expressible by a formula $pat(x)$, for any $x \in U$. It means that $I(x) = \|pat(x)\|_U \subseteq U$, where $\|pat(x)\|_U$ denotes the meaning of $pat(x)$ restricted to the sample U . In case of rule-based classifiers patterns of the form $pat(x)$ are defined by feature value vectors.

We assume that for any new object $x \in U^\infty \setminus U$, we can obtain (e.g., as a result of sensor measurement) a pattern $\text{pat}(x) \in L$ with semantics $\|\text{pat}(x)\|_{U^\infty} \subseteq U^\infty$. However, the relationships between information granules over U^∞ like sets $\|\text{pat}(x)\|_{U^\infty}$ and $\|\text{pat}(y)\|_{U^\infty}$, for different $x, y \in U^\infty$, are, in general, known only if they can be expressed by relationships between the restrictions of these sets to the sample U , i.e., between sets $\Pi_U(\|\text{pat}(x)\|_{U^\infty})$ and $\Pi_U(\|\text{pat}(y)\|_{U^\infty})$.

The set of patterns $\{\text{pat}(x) : x \in U\}$ is usually not relevant for approximation of the concept $C \subseteq U^\infty$. Such patterns are too specific or not enough general, and can directly be applied only to a very limited number of new objects. However, by using some generalization strategies, one can search, in a family of patterns definable from $\{\text{pat}(x) : x \in U\}$ in L , for such new patterns that are relevant for approximation of concepts over U^∞ . Let us consider a subset $\text{PATTERNS}(AS, L, C) \subseteq L$ chosen as a set of pattern candidates for relevant approximation of a given concept C . For example, in case of rule-based classifier one can search for such candidate patterns among sets definable by subsequences of feature value vectors corresponding to objects from the sample U . The set $\text{PATTERNS}(AS, L, C)$ can be selected by using some quality measures checked on meanings (semantics) of its elements restricted to the sample U (like the number of examples from the concept $\Pi_U(C)$ and its complement that support a given pattern). Then, on the basis of properties of sets definable by these patterns over U , we induce approximate values of the inclusion function v_C on subsets of U^∞ definable by any of such pattern and the concept C .

Next, we induce the value of v_C on pairs (X, Y) where $X \subseteq U^\infty$ is definable by a pattern from $\{\text{pat}(x) : x \in U^\infty\}$ and $Y \subseteq U^\infty$ is definable by a pattern from $\text{PATTERNS}(AS, L, C)$.

Finally, for any object $x \in U^\infty \setminus U$, we induce the approximation of the degree $v_C(\|\text{pat}(x)\|_{U^\infty}, C)$, applying a conflict resolution strategy Conflict_res (a voting strategy, in case of rule-based classifiers) to two families of degrees:

$$\{v_C(\|\text{pat}(x)\|_{U^\infty}, \|\text{pat}\|_{U^\infty}) : \text{pat} \in \text{PATTERNS}(AS, L, C)\}, \quad (33)$$

$$\{v_C(\|\text{pat}\|_{U^\infty}, C) : \text{pat} \in \text{PATTERNS}(AS, L, C)\}. \quad (34)$$

Values of the inclusion function for the remaining subsets of U^∞ can be chosen in any way – they do not have any impact on the approximations of C . Moreover, observe that for the approximation of C we do not need to know the exact values of uncertainty function I_C – it is enough to induce the values of the inclusion function v_C . Observe that the defined extension v_C of v to some subsets of U^∞ makes it possible to define an approximation of the concept C in a new approximation space AS_C .

Observe that one can also follow principles of Bayesian reasoning and use degrees of v_C to approximate C (see, e.g., [107, 108]).

In this way, the rough set approach to induction of concept approximations can be explained as a process of inducing a relevant approximation space.

13.5.3 Higher Order Vagueness

In [69], it is stressed that vague concepts should have non-crisp boundaries. In the definition presented in [30], the notion of boundary region is defined as a crisp set $BN_B(X)$. However, let us observe that this definition is relative to the subjective knowledge expressed by attributes from B . Different sources of information may use different sets of attributes for concept approximation. Hence, the boundary region can change when we consider these different views. Another aspect is discussed in [70], where it is assumed that information about concepts is incomplete; e.g., the concepts are given only on samples (see, e.g., [64, 76, 86]). From [70] it follows that vague concepts cannot be approximated with satisfactory quality by static constructs such as induced membership inclusion functions, approximations, or models derived, e.g., from a sample. Understanding of vague concepts can only be realized in a process in which the induced models are adaptively matching the concepts in a dynamically changing environment. This conclusion seems to have important consequences for further development of rough set theory in combination with fuzzy sets and other soft computing paradigms for adaptive approximate reasoning.

13.6 Rough Sets and Granular Computing

In our approach to GC, we use the general optimization criterion based on the minimal length principle. In searching for (sub-)optimal solutions it is necessary to construct many compound granules using some specific operations such as generalization, specification, or fusion. Granules are labeled by parameters. By tuning these parameters we optimize the granules relative to their description size and the quality of data description, i.e., two basic components on which the optimization measures are defined.

From this general description of tasks in RGC, it follows that together with specification of elementary granules and operation on them it is necessary to define measures of granule quality (e.g., measures of their inclusion, covering, or closeness) and tools for measuring the size of granules. Very important are also optimization strategies searching for (parameterized) granules.

We discuss the searching process for relevant (for concept approximation) neighborhoods in approximation spaces based on modeling relevant relational and syntactical structures built from partial information about objects and concepts.

We would also like to emphasize that the importance in RGC of risk measures defined on granules is emphasized. The values of such measures are indicating how properties of granules are changing when some of their parameters were changed.

We also present an example showing how utility functions defined on granules can be used in RGC. In general, utility functions are helping to relax the exact constraints by making it possible to work with constraints which should be satisfied to a degree expressed by utility functions.

First, we discuss issues related to ontology approximation and the rough mereological approach.

13.6.1 Ontological Framework for Approximation

In a number of papers (see, e.g., [93]), the problem of ontology approximation has been discussed together with possible applications to approximation of compound concepts or to knowledge transfer (see, e.g., paper by T.T. Nguyen in [50], by Skowron in [93], and in [109]).

In the ontology [110] (vague) concepts and local dependencies between them are specified. Global dependencies can be derived from local dependencies. Such derivations can be used as hints in searching for relevant compound patterns (information granules) in approximation of more compound concepts from the ontology. The ontology approximation problem is one of the fundamental problems related to approximate reasoning in distributed environments. One should construct (in a given language that is different from the ontology specification language) not only approximations of concepts from ontology but also vague dependencies specified in the ontology. It is worthwhile to mention that an ontology approximation should be induced on the basis of incomplete information about concepts and dependencies specified in the ontology. Information granule calculi based on rough sets have been proposed as tools making it possible to solve this problem. Vague dependencies have vague concepts in premisses and conclusions. The approach to approximation of vague dependencies based only on degrees of closeness of concepts from dependencies and their approximations (classifiers) is not satisfactory for approximate reasoning. Hence, more advanced approach should be developed. Approximation of any vague dependency is a method which allows for any object to compute the arguments ‘for’ and ‘against’ its membership to the dependency conclusion on the basis of the analogous arguments relative to the dependency premisses. Any argument is a compound information granule (compound pattern). Arguments are fused by local schemes (production rules) discovered from data. Further fusions are possible through composition of local schemes, called approximate reasoning schemes (AR schemes) (see, e.g., [11]). To estimate the degree to which (at least) an object belongs to concepts from ontology the arguments ‘for’ and ‘against’ those concepts are collected and next a conflict resolution strategy is applied to them to predict the degree.

13.6.2 Mereology and Rough Mereology

This section introduces some basic concepts of rough mereology (see, e.g., papers by Polkowski *et al.* in [15, 31, 68]).

Exact and rough concepts can be characterized by a new notion of an element, alien to naive set theory in which this theory has been coded until now. For an information system $\mathcal{A} = (U, A)$, and a set B of attributes, the mereological element el_B^A is defined by letting

$$xel_B^A X \text{ if and only if } B(x) \subseteq X. \quad (35)$$

Then, a concept X is B -exact if and only if either $xel_B^A X$ or $xel_B^A U \setminus X$ for each $x \in U$, and the concept X is B -rough if and only if for some $x \in U$ neither $xel_B^A X$ nor $xel_B^A U \setminus X$.

Thus, the characterization of the dichotomy exact–rough cannot be done by means of the element notion of naive set theory, but requires the notion of containment (\subseteq), i.e., a notion of mereological element.

The Leśniewski mereology (theory of parts) is based on the notion of a part [66]. The relation π of part on the collection U of objects satisfies the following:

$$1. \text{ If } x\pi y \text{ then not } y\pi x. \quad (36)$$

$$2. \text{ If } x\pi y \text{ and } y\pi z \text{ then } x\pi z. \quad (37)$$

The notion of mereological element el_π is introduced as

$$xel_\pi y \text{ if and only if } x\pi y \text{ or } x = y. \quad (38)$$

In particular, the relation of proper inclusion \subset is a part relation π on any non-empty collection of sets, with the element relation $el_\pi = \subseteq$.

Formulas expressing, e.g., rough membership, quality of decision rule, and quality of approximations can be traced back to a common root, i.e., $v(X, Y)$ defined by equation (27). The value $v(X, Y)$ defines the degree of *partial containment* of X into Y and naturally refers to the Leśniewski mereology. An abstract formulation of this idea in paper by Polkowski *et al.* in [68] connects the mereological notion of element el_π with the partial inclusion by introducing a *rough inclusion* as a relation $v \subseteq U \times U \times [0, 1]$ on a collection of pairs of objects in U endowed with part π relation, and such that

$$1. v(x, y, 1) \text{ if and only if } xel_\pi y; \quad (39)$$

$$2. \text{ if } v(x, y, 1) \text{ then (if } v(z, x, r) \text{ then } v(z, y, r)); \quad (40)$$

$$3. \text{ if } v(z, x, r) \text{ and } s < r \text{ then } v(z, x, s). \quad (41)$$

Implementation of this idea in information systems can be based on *Archimedean t-norms* (see paper by Polkowski *et al.* in [68]); each such norm T is represented as $T(r, s) = g(f(r) + f(s))$, with f, g pseudoinverses to each other continuous and decreasing on $[0, 1]$. Letting for (U, A) and $x, y \in U$

$$DIS(x, y) = \{a \in A : a(x) \neq a(y)\}, \quad (42)$$

and

$$v(x, y, r) \text{ if and only if } g\left(\frac{\text{card}(DIS(x, y))}{\text{card}(A)}\right) \geq r, \quad (43)$$

v defines a rough inclusion that satisfies additionally the transitivity rule

$$\frac{v(x, y, r), v(y, z, s)}{v(x, z, T(r, s))}. \quad (44)$$

Simple examples here are as follows: the Menger rough inclusion in the case $f(r) = -\ln r$, $g(s) = e^{-s}$ yields $v(x, y, r)$ if and only if $e^{-\frac{\text{card}(DIS(x, y))}{\text{card}(A)}} \geq r$ and it satisfies the transitivity rule:

$$\frac{v(x, y, r), v(y, z, s)}{v(x, y, r \cdot s)}; \quad (45)$$

i.e., the t -norm T is the Menger (product) t -norm $r \cdot s$, and, the Łukasiewicz rough inclusion with $f(x) = 1 - x = g(x)$ yielding $v(x, y, r)$ if and only if $1 - \frac{\text{card}(\text{DIS}(x, y))}{\text{card}(A)} \geq r$ with the transitivity rule:

$$\frac{v(x, y, r), v(y, z, s)}{v(x, z, \max\{0, r + s - 1\})}, \quad (46)$$

i.e., with the Łukasiewicz t -norm.

Rough inclusions (see paper by Polkowski *et al.* in [68]) can be used in *granulation of knowledge* [60]. Granules of knowledge are constructed as aggregates of indiscernibility classes close enough with respect to a chosen measure of closeness. In a nutshell, a granule $g_r(x)$ about x of radius r can be defined as the aggregate of all y with $v(y, x, r)$. The aggregating mechanism can be based on the class operator of mereology (cf. rough mereology in paper by Polkowski *et al.* in [68]) or on set-theoretic operations of union.

Rough mereology combines rough inclusions with methods of mereology. It employs the operator of mereological class that makes collections of objects into objects. The class operator Cls satisfies the requirements, with any non-empty collection M of objects made into the object $\text{Cls}(M)$

$$\text{if } x \in M, \text{ then } x \in \text{el}_\pi \text{Cls}(M); \quad (47)$$

$$\text{if } x \in \text{el}_\pi \text{Cls}(M), \text{ then there exist } y, z \text{ such that } y \in \text{el}_\pi x, y \in \text{el}_\pi z, z \in M. \quad (48)$$

In case of the part relation \subset on a collection of sets, the class $\text{Cls}(M)$ of a non-empty collection M is the union $\bigcup M$.

Granulation by means of the class operator Cls consists in forming the granule $g_r(x)$ as the class $\text{Cls}(y : v(y, x, r))$. One obtains a granule family with regular properties (see [50]).

13.6.3 Quality of Approximation Space

A key task in GC is the information granulation process, which is responsible in the formation of information aggregates (patterns) from a set of available data. A methodological and algorithmic issue is the formation of transparent (understandable) information granules, meaning that they should provide a clear and understandable description of patterns held in data. Such fundamental property can be formalized by a set of constraints that must be satisfied during the information granulation process. Usefulness of these constraints is measured by quality of approximation space:

$$\text{Quality}_1 : \text{Set_AS} \times P(U) \rightarrow [0, 1], \quad (49)$$

where U is a non-empty set of objects and Set_AS is a set of possible approximation spaces with the universe U .

Example 1. If $\text{AS}^*(X) \neq \emptyset$ for $\text{AS} \in \text{Set_AS}$ and $X \subseteq U$ then

$$\text{Quality}_1(\text{AS}, X) = v_{\text{SRI}}(\text{AS}^*(X), \text{AS}_*(X)) = \frac{\text{card}(\text{AS}_*(X))}{\text{card}(\text{AS}^*(X))}. \quad (50)$$

The value $1 - \text{Quality}_1(\text{AS}, X)$ expresses the degree of completeness of our knowledge about X , given the approximation space AS .

Example 2. In applications we usually use another quality measures based on the minimal length principle [81, 83] where the description length of approximation is also included. Let us denote by $\text{description}(\text{AS}, X)$ the description length of approximation of X in AS . The description length may be measured, e.g., by the sum of description lengths of algorithms testing membership for neighborhoods used in construction of the lower approximation, the upper approximation, and the boundary region of

the set X . Then the quality $\text{Quality}_2(\text{AS}, X)$ can be defined by

$$\text{Quality}_2(\text{AS}, X) = g(\text{Quality}_1(\text{AS}, X), \text{description}(\text{AS}, X)), \quad (51)$$

where g is a relevant function used for fusion of values $\text{Quality}_1(\text{AS}, X)$ and $\text{description}(\text{AS}, X)$.

One can consider different optimization problems relative to a given class Set_AS of approximation spaces. For example, for given $X \subseteq U$ and a threshold $t \in [0, 1]$ one can search for an approximation space AS , satisfying the constraint $\text{Quality}(\text{AS}, X) \geq t$. Another example can be related to searching for an approximation space satisfying additionally the constraint $\text{Cost}(\text{AS}) < c$, where $\text{Cost}(\text{AS})$ denotes the cost of approximation space AS (e.g., measured by the number of attributes used to define neighborhoods in AS) and c is a given threshold.

In the process of searching for (sub-)optimal approximation spaces, different strategies are used. Let us consider one illustrative example. Let $DT = (U, A, d)$ be a decision system (a given sample of data), where U is a set of objects, A a set of attributes, and d a decision. We assume that for any object x is accessible only a partial information equal to the A -signature of x (object signature, for short), i.e., $\text{Inf}_A(x) = \{(a, a(x)) : a \in A\}$ and analogously for any concept there is only given a partial information about this concept by a sample of objects, e.g., in the form of decision table. One can use object signatures as new objects in a new relational structure \mathcal{R} . In this relational structure \mathcal{R} are also modeled some relations between object signatures, e.g., defined by the similarities of these object signatures. Discovery of relevant relations on object signatures is an important step in the searching process for relevant approximation spaces. In the next step, we select a language \mathcal{L} of formulas expressing properties over the defined relational structure \mathcal{R} and we search for relevant formulas in \mathcal{L} . The semantics of formulas (e.g., with one free variable) from \mathcal{L} are subsets of object signatures. Observe that each object signature defines a neighborhood of objects from a given sample (e.g., decision table DT) and another set on the whole universe of objects being an extension of U . In this way, each formula from \mathcal{L} defines a family of sets of objects over the sample and also another family of sets over the universe of all objects. Such families can be used to define new neighborhoods of a new approximation space, e.g., by taking unions of the described above families. In the searching process for relevant neighborhoods, we use information encoded in the given sample. More relevant neighborhoods are making it possible to define relevant approximation spaces (from the point of view of the optimization criterion). It is worth to mention that often this searching process is even more compound. For example, one can discover several relational structures (not only one, e.g., \mathcal{R} as it was presented before) and formulas over such structures defining different families of neighborhoods from the original approximation space and next fuse them for obtaining one family of neighborhoods or one neighborhood in a new approximation space. Such kind of modeling is typical for hierarchical modeling (see paper by Bazan *et al.* [46]), e.g., when we search for relevant approximation space for objects composed from parts for which some relevant approximation spaces have been already found.

Let us consider some illustrative examples of granule modeling (see Figure 13.5). Any object $x \in U$, in a given information system $IS_1 = (U, A)$, is perceived by means of its signature $\text{Inf}_A(x) = \{(a, a(x)) : a \in A\}$. On the first level, we consider objects with signatures represented by the information system $IS_1 = (U, A)$. Objects with the same signature are indiscernible. On the next level of modeling we consider as objects some relational structures over signatures of objects from the first level. For example, for any signature u one can consider as a relational structure a neighborhood defined by a similarity relation τ between signatures of objects from the first level (see Figure 13.5). Attributes of objects on the second level describe properties of relational structures. Hence, indiscernibility classes defined by such attributes are sets of relational structures – in our example sets of neighborhoods. We can continue this process of hierarchical modeling by considering as objects on the third-level signatures of objects from the second level. In our example, the third level of modeling represents modeling of clusters of neighborhoods defined by the similarity relation τ . Observe that it is possible to link objects from a higher level with objects from a lower level. In our example, any object from the second level is a neighborhood or τ . Any element u' of this neighborhood defines on the first level an elementary granule (indiscernibility class) $\{x \in U : \text{Inf}_A(x) = u'\}$. Hence, any neighborhood $\tau(u)$ defines on the first level a family of elementary

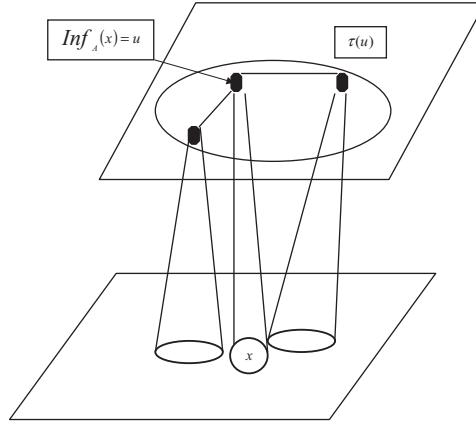


Figure 13.5 Modeling of granules

granules corresponding to signatures from the neighborhood. Now, one can consider as a quality measure for the similarity τ a function assigning to τ a degree to which the union of the elementary granules mentioned above is included into a given concept.

In the second example, we assume that the information system on the first level has a bit more general structure. Namely, on any attribute value set V_a there is defined a relational structure \mathcal{R}_a and a language \mathcal{L}_a of formulas for expressing properties over V_a . For example, one can consider an attribute *time* with values in the set \mathcal{N} of natural numbers, i.e., $V_a \subseteq \mathcal{N}$. The value $time(x)$ is interpreted as a time at which the object x was perceived. The relational structure \mathcal{R}_{time} is defined by (V_a, S) , where S is the successor relation in \mathcal{N} , i.e., xSy if and only if $y = x + 1$. Then relational structures on the second layer can correspond to windows of a given length T , i.e., structures of the form $(\{u_1, \dots, u_T\}, S)$, where for some x_1, \dots, x_T we have $u_i = Inf_A(x_i)$ and $time(x_{i+1}) = time(x_i) + 1$ for $i = 1, \dots, T$. Hence, the attributes on the second layer of modeling correspond to properties of windows, while attributes on the third level could correspond to clusters of windows. Again in looking for relevant clusters we should consider links of the higher levels with lower levels. Another possibility will be to consider some relational structures on the attribute values sets on the second layer. This makes it possible to model relations between windows such as the relation *overlap* or the relation *earlier than*. Then, attributes on this level could describe properties of sequences of windows. Such attributes can correspond to some models of processes. Yet another possibility is to use additionally some spatial relations (e.g., nearness) between the successive elements of windows.

For structural objects is often used a decomposition method for modeling relational structures on the second level. The signature of objects is decomposed into parts and some relations between parts are considered, which are defined over relational structures with the universe $\times_{a \in A} V_a$.

In the third example, we assume that the information system on the second level is constructed by constrained sum of information systems from the first level (see, e.g., paper by Skowron *et al.* in [11]). For example, any object of $+_R(IS_1, IS_2)$ consists of pairs (x_1, x_2) of objects from IS_1 and IS_2 , satisfying some constraints described by $R \subseteq U_1 \times U_2$, i.e., $U = R \cap (U_1 \times U_2)$. The attributes of $+_R(IS_1, IS_2)$ consist of the attributes of IS_1 and IS_2 , except that if there are any attributes in common, then we make their distinct copies to avoid confusion. The relational structures on the second level are constructed using signatures of parts and relations between them described by constraints.

The above examples are typical for GC where for a given task it is necessary to search for granules in a given granular system which are satisfying some optimization criteria. The discussed methods are used in spatiotemporal reasoning (see, e.g., [111]), in behavioral pattern identification, and planning (see, e.g., papers by Bazan *et al.* in [46, 53]).

There are some other basic concepts which should be considered in GC. One of them is related to risk. In the following section we present some remarks about risk in construction of granules.

13.6.4 Risk and Utility Functions in Construction of Granules

There is a large literature on relationships between decision making and risk. In this section, we discuss some problems related to risk in GC. An example of risk analysis (based on rough sets) for medical data the reader can find in the paper by Bazan *et al.* in [39].

First, we recall the definition of *granule system*. Any such system GS consists of a set of granules G . Moreover, a family of relations with the intended meaning *to be a part to a degree* between granules is distinguished. The degree structure is described by a relation *to be an exact part*. More formally, a granule system is any tuple

$$GS = (G, H, <, \{v_p\}_{p \in H}, size), \quad (52)$$

where G is a non-empty set of granules. H is a non-empty set of granule inclusion degrees with a binary relation $<$ (usually a strict partial order), which defines on H a structure used to compare the degrees. $v_p \subseteq G \times G$ is a binary relation *to be a part to a degree at least p* between granules from G , called *rough inclusion*. Let $size : G \rightarrow R_+$ be the granule size function, where R_+ is the set of non-negative reals.¹⁶

In construction of granule systems it is necessary to give a constructive definition of all their components. In particular, one should specify how more compound granules are defined from already-defined granules or given elementary granules. Usually, the set of granules is defined as the least set generated from distinguished elementary granules (e.g., defined by indiscernibility classes) by some operations on the granules. These operations are making it possible to fuse elementary granules for obtaining new granules relevant for the task to be solved. In the literature many different operations on granules are reported (see, e.g., paper by Skowron *et al.* in [11]) from those defined by Boolean combination of descriptors to compound classifiers or networks of classifiers.

Let us consider, a task of searching in the set of granules of a granule system GS for a granule g , satisfying a given constraint to a satisfactory degree, e.g., $v_{tr}(g, g_0)$, where $v : G \times G \rightarrow [0, 1]$ is the inclusion function, $v_{tr}(g, g_0)$ means that $v(g, g_0) \geq tr$, g_0 is a given granule, and tr is a given threshold. Let g^* be a solution; i.e., g^* satisfies the condition

$$v(g^*, g_0) > tr. \quad (53)$$

Risk analysis is a well-established notion in decision theory [112]. We would like to illustrate the importance of risk analysis in GC.

A typical risk analysis task in GC can be described as follows. For a granule g^* is constructed a granule $N(g^*)$, i.e., representing a cluster of granules defined by g^* received by changing some parameters of g^* such as attribute values used in the g^* description. We would like to estimate how this change influences the condition (53).

First, let us assume that $v(g^*, g_0) = v_{SRI}(\|g^*\|, \|g_0\|)$, where $\|\cdot\|$ denotes the semantic of granule, i.e., a function $\|\cdot\| : G \rightarrow P(U)$ for a given universe of objects U and v_{SRI} is the standard rough inclusion function. Then, one can take

$$\delta^* = \arg \min_{\delta \in [0, tr]} (v(N(g^*), g_0) \geq tr - \delta).$$

The value δ^* can be treated as a *risk degree* of changing the inclusion degree in g_0 when the granule g^* is substituted by $N(g^*)$.

One can consider a hierarchy of granules over g^* defined by an ascending sequence $N_1(g^*), \dots, N_k(g^*)$, i.e., $\|N_1(g^*)\| \leq \dots \leq \|N_k(g^*)\|$ and corresponding risk degrees $\delta_1^* \leq \dots \leq \delta_k^*$. For example, if δ_1^* is sufficiently small then g^* is called *robust* with respect to deviations caused by taking $N_1(g^*)$ instead of g^* . However, when i is increasing then taking $N_i(g^*)$ instead of g^* gradually increases the risk degree. The above example illustrates the importance of risk analysis in GC. Information maps introduced in [113] can be used for risk analysis.

¹⁶ Properties which such a function should satisfy will be discussed elsewhere.

Let us now move to the concept of *utility function* over granules. The concept of utility function has been intensively studied in decision theory or game theory [114, 115]. We would like to present an illustrative example showing that such functions are important for granule systems.

We assume two granule systems GS and GS_0 with granule sets G and G_0 are given. We consider two properties of granules in this systems, i.e., $P \subseteq G$ and $P_0 \subseteq G_0$. Moreover we assume that checking the membership for P is much simpler than that for P_0 (e.g., because granules from G_0 are much simpler than granules from G). This means that there are given algorithms \mathcal{A} and \mathcal{A}_0 for checking the membership in P and P_0 , respectively, and the complexity of \mathcal{A}_0 is much lower than the complexity of the algorithm \mathcal{A} . Under the above assumptions it is useful to search for a *utility function* $Utility : G \rightarrow G_0$, reducing the membership problem for P to the membership problem for P_0 , i.e., a function with the following property: $g \in P$ if and only if $Utility(g) \in P_0$. Construction of the utility function satisfying the above condition may be not feasible. However, it often becomes feasible when we relax the binary membership relation \in to the membership at least to a given degree (see, e.g., [102]). This example illustrates the important property of utility functions. Usually, G_0 is a set of scalar values or it is assumed that some preference relation over G_0 is given.

Finally, we would like to add that in GC it is necessary to develop methods searching for approximation of risk degrees and utility function from data and domain knowledge analogously to approximation of complex concepts (see, e.g., paper by Bazan *et al.* in [46]).

For systems searching for adaptive approximation of complex concepts, it is necessary to develop strategies based on the minimal length principle in GC, risk measures in GC, and utility functions in GC. This will also require developing methods for approximation of risk measures and utility functions.

13.7 Rough Sets and Logic

The father of contemporary logic is a German mathematician Gottlob Frege (1848–1925). He thought that mathematics should not be based on the notion of set but on the notions of logic. He created the first axiomatized logical system but it was not understood by the logicians of those days.

During the first three decades of the twentieth-century, there was a rapid development in logic bolstered to a great extent by Polish logicians, especially Alfred Tarski (1901–1983).

Development of computers and their applications stimulated logical research and widened its scope in areas such as proof theory, linear logic, intuitionistic logic, and higher order logic.

When we speak about logic, we generally mean *deductive logic*. It gives us tools designed for deriving true propositions from other true propositions. Deductive reasoning always leads to true conclusions. The theory of deduction has well-established, generally accepted, theoretical foundations. Deductive reasoning is the main tool used in mathematical reasoning and found no application beyond it.

Rough set theory has contributed to some extent to various kinds of deductive reasoning. Particularly, various kinds of logics based on the rough set approach have been investigated, rough set methodology contributed essentially to modal logics, many-valued logic, intuitionistic logic, and others (see, e.g., references in [28]).

A summary of this research can be found in [14] and interested reader is advised to consult this volume.

In natural sciences (e.g., in physics) *inductive reasoning* is of primary importance. The characteristic feature of such reasoning is that it does not begin from axioms (expressing general knowledge about the reality) like in deductive logic, but some partial knowledge (examples) about the universe of interest are the starting point of this type of reasoning, which are generalized next and they constitute the knowledge about wider reality than the initial one. In contrast to deductive reasoning, inductive reasoning does not lead to true conclusions but only to probable (possible) ones. Also in contrast to the logic of deduction, the logic of induction does not have uniform, generally accepted, theoretical foundations as yet, although many important and interesting results have been obtained, e.g., concerning statistical and computational learning and others.

Verification of validity of hypotheses in the logic of induction is based on experiment rather than the formal reasoning of the logic of deduction. Physics is the best illustration of this fact.

The research on inductive logic has a few centuries' long history and outstanding English philosopher John Stuart Mill (1806–1873) is considered its father [116].

The creation of computers and their innovative applications essentially contributed to the rapid growth of interest in inductive reasoning. This domain develops very dynamically thanks to computer science. Machine learning, knowledge discovery, reasoning from data, expert systems, and others are examples of new directions in inductive reasoning. It seems that rough set theory is very well suited as a theoretical basis for inductive reasoning. Basic concepts of this theory fit very well to represent and analyze knowledge acquired from examples, which can be next used as starting point for generalization. Besides, in fact rough set theory has been successfully applied in many domains to find patterns in data (data mining) and acquire knowledge from examples (learning from examples). Thus, rough set theory seems to be another candidate as a mathematical foundation of inductive reasoning (see, e.g., papers by Bazan *et al.* in [21, 46] and [102]).

The most interesting from computer science point of view is *commonsense* reasoning. We use this kind of reasoning in our everyday life, and examples of such kind of reasoning we face in newspapers, radio, TV, etc., in political and economic debates and discussions.

The starting point to such reasoning is the knowledge possessed by the specific group of people (*common knowledge*) concerning some subject and intuitive methods of deriving conclusions from it. We do not have here possibilities of resolving the dispute by means of methods given by deductive logic (reasoning) or by inductive logic (experiment). So the best known method of solving the dilemma voting, negotiations, or even war.

These methods do not reveal the truth or falsity of the thesis under consideration at all. Of course, such methods are not acceptable in mathematics or physics. Nobody is going to solve by voting, negotiations, or declare a war the truth of Fermat's theorem or Newton's laws.

Reasoning of this kind is the least studied from the theoretical point of view and its structure is not sufficiently understood, in spite of many interesting theoretical research in this domain [117]. The meaning of commonsense reasoning, considering its scope and significance for some domains, is fundamental and rough set theory can also play an important role in it but more fundamental research must be done to this end (see, e.g., paper by Skowron *et al.* [50]).

In particular, the rough truth introduced in [118] and studied seems to be important for investigating commonsense reasoning in the rough set framework [117].

Let us consider a simple example. In the considered decision table we assume that $U = \text{Birds}$ is a set of birds that are described by some condition attributes from a set A . The decision attribute is a binary attribute Flies with possible values *yes* if the given bird flies and *no* otherwise. Then, we define the set of abnormal birds by $Ab_A(\text{Birds}) = A^*(\{x \in \text{Birds} : \text{Flies}(x) = \text{no}\})$. Hence, we have $Ab_A(\text{Birds}) = \text{Birds} - A^*(\{x \in \text{Birds} : \text{Flies}(x) = \text{yes}\})$ and $\text{Birds} - Ab_A(\text{Birds}) = A^*(\{x \in \text{Birds} : \text{Flies}(x) = \text{yes}\})$. It means that for normal birds it is consistent, with knowledge represented by A , to assume that they can fly; i.e., it is possible that they can fly. One can optimize $Ab_A(\text{Birds})$ using A to obtain minimal boundary region in the approximation of $\{x \in \text{Birds} : \text{Flies}(x) = \text{no}\}$.

It is worthwhile to mention that in [117] has been presented an approach combining the rough sets with non-monotonic reasoning. There are distinguished some basic concepts that can be approximated on the basis of sensor measurements and more complex concepts that are approximated using so-called transducers defined by first-order theories constructed over approximated concepts. Another approach to commonsense reasoning has been developed in a number of papers (see, e.g., papers by Bazan *et al.* in [21, 46] and Skowron *et al.* in [11, 50]). The approach is based on an ontological framework for approximation. In this approach approximations are constructed for concepts and dependencies between the concepts represented in a given ontology expressed, e.g., in natural language. Still another approach combining rough sets with logic programming is discussed in the paper by Vitória in [23].

To recapitulate, the characteristics of the three above-mentioned kinds of reasoning are given below:

1. Deductive:

- reasoning method: axioms and rules of inference;
- applications: mathematics;

- theoretical foundations: complete theory;
- conclusions: true conclusions from true premisses;
- hypotheses verification: formal proof.

2. Inductive:

- reasoning method: generalization from examples;
- applications: natural sciences (physics);
- theoretical foundation: lack of generally accepted theory;
- conclusions: not true but probable (possible);
- hypotheses verification: experiment.

3. Common sense:

- reasoning method based on commonsense knowledge with intuitive rules of inference expressed in natural language;
- applications: everyday life, humanities;
- theoretical foundation: lack of generally accepted theory;
- conclusions obtained by mixture of deductive and inductive reasoning based on concepts expressed in natural language, e.g., with application of different inductive strategies for conflict resolution (such as voting, negotiations, cooperation, and war) based on human behavioral patterns;
- hypotheses verification: human behavior.

Finally we would like to refer to [119] where some research trends within the Rasiowa–Pawlak school concerning the application of logic to AI have been selected and discussed. The methods investigated in this school have undergone a remarkable evolution during the past couple decades. This evolution, however, indicates certain directions for the future studies. For a better understanding of the evolutionary scope of the research directions conducted within the Rasiowa–Pawlak school, readers are encouraged to consult Figure 13.6.

13.8 Exemplary Research Directions and Applications

In this chapter we have discussed some basic issues and methods related to rough sets. For more detail the reader is referred to the literature cited at the beginning of this chapter (see also <http://rsds.wsiz.rzeszow.pl>).

We are now observing a growing research interest in the foundations of rough sets, including the various logical, algebraic, and philosophical aspects of rough sets and complexity issues (see, e.g., references in [28]).

Some relationships have already been established between rough sets and other approaches as well as a wide range of hybrid systems have been developed (see, e.g., references in [28]).

As a result, rough sets are linked with decision system modeling and analysis of complex systems, fuzzy sets, neural networks, evolutionary computing, data mining and knowledge discovery, pattern recognition, machine learning, data mining, approximate reasoning, and multicriteria decision making. In particular, rough sets are used in probabilistic reasoning, granular computing (including information granule calculi based on rough mereology), intelligent control, intelligent agent modeling, identification of autonomous systems, and process specification.

A wide range of applications of methods based on rough set theory alone or in combination with other approaches have been discovered in the following areas: acoustics, bioinformatics, business and finance, chemistry, computer engineering (e.g., data compression, digital image processing, digital signal processing, parallel and distributed computer systems, sensor fusion, and fractal engineering), decision analysis and systems, economics, electrical engineering (e.g., control, signal analysis, and power systems), environmental studies, digital image processing, informatics, medicine, molecular biology, musicology, neurology, robotics, social science, software engineering, spatial visualization, Web engineering, and Web

Evolution of AI models of computing in the Rasiowa-Pawlak School



Inspirations outside of mathematics

Figure 13.6 Evolution of logical approaches to AI in the Rasiowa–Pawlak school

mining. Below we list some suggestions for readings in selected application domains (for references see, e.g., [28]):

- machine learning, pattern recognition, data mining, and knowledge discovery;
- bioinformatics;
- multicriteria decision making;
- medicine;
- signal processing and image processing;
- hierarchical learning and ontology approximation; and
- other domains.

13.9 Rough Sets: A Challenge

There are many real-life problems that are still hard to solve using the existing methodologies and technologies. Among such problems are, e.g., classification of medical images, control of autonomous systems like unmanned aerial vehicles or robots, or problems related to monitoring or rescue tasks in multiagent systems. All these problems are closely related to intelligent systems that are more and more widely applied in different real-life projects.

One of the main challenges in developing intelligent systems is discovering methods for approximate reasoning from measurements to perception, i.e., deriving from concepts resulting from sensor measurements concepts enunciated in natural language that express perception.

Nowadays, new emerging computing paradigms are investigated, attempting to make progress in solving problems related to this challenge. Further progress depends on a successful cooperation of specialists from different scientific disciplines such as mathematics, computer science, artificial intelligence, biology, physics, chemistry, bioinformatics, medicine, neuroscience, linguistics, psychology, and sociology. In particular, different aspects of reasoning from measurements to perception are investigated in psychology [120], neuroscience [121], layered learning [122], mathematics of learning [121], machine learning, pattern recognition [76], and data mining [64], and also by researchers working on recently emerged computing paradigms such as CWP [60], GC [11], rough sets, rough-mereology, and rough-neural computing [11].

One of the main problems investigated in machine learning, pattern recognition [76], and data mining [64] is concept approximation. It is necessary to induce approximations of concepts (models of concepts) from available experimental data. The data models developed so far in such areas like statistical learning, machine learning, and pattern recognition are not satisfactory for approximation of compound concepts resulting in the perception process. Researchers from the different areas have recognized the necessity to work on new methods for concept approximation (see, e.g., [103, 104]). The main reason is that these compound concepts are, in a sense, too far from measurements, which makes the searching for relevant (for their approximation) features infeasible in a huge space. There are several research directions aiming at overcoming this difficulty. One of them is based on the interdisciplinary research where the results concerning perception in psychology or neuroscience are used to help deal with compound concepts (see, e.g., [76]). There is a great effort in neuroscience toward understanding the hierarchical structures of neural networks in living organisms [121]. Also, mathematicians are recognizing problems of learning as the main problem of the current century [121]. The problems discussed so far are also closely related to complex system modeling. In such systems again the problem of concept approximation and reasoning about perceptions using concept approximations is one of the challenges nowadays. One should take into account that modeling complex phenomena entails the use of local models (captured by local agents, if one would like to use the multiagent terminology (see paper by Skowron in [20]) that next should be fused. This process involves the negotiations between agents to resolve contradictions and conflicts in local modeling. This kind of modeling will become more and more important in solving complex real-life problems, which we are unable to model using traditional analytical approaches. The latter approaches lead to exact models. However, the necessary assumptions used to develop them are causing the resulting solutions to be too far from reality to be accepted. New methods or even a new science should be developed for such modeling [123].

One of the possible solutions in searching for methods for compound concept approximations is the layered learning idea [122]. Inducing concept approximation should be developed hierarchically starting from concepts close to sensor measurements to compound target concepts related to perception. This general idea can be realized using additional domain knowledge represented in natural language. For example, one can use principles of behavior on the roads, expressed in natural language, trying to estimate, from recordings (made, e.g., by camera and other sensors) of situations on the road, if the current situation on the road is safe or not. To solve such a problem one should develop methods for concept approximations together with methods aiming at approximation of reasoning schemes (over such concepts) expressed in natural language. Foundations of such an approach are based on rough set theory [13] and its extension rough mereology (see, e.g., papers by Polkowski *et al.* in [11, 15, 16, 68]), both discovered in Poland.

Objects we are dealing with are information granules. Such granules are obtained as the result of information granulation [60].

CWP ‘derives from the fact that it opens the door to computation and reasoning with information which is perception- rather than measurement-based. Perceptions play a key role in human cognition, and underlie the remarkable human capability to perform a wide variety of physical and mental tasks without any measurements and any computations. Everyday examples of such tasks are driving a car in city traffic, playing tennis and summarizing a story’ [60].

The rough mereological approach (see, e.g., papers by Polkowski *et al.* in [11, 15, 16, 68]) is based on calculi of information granules for constructing compound concept approximations. Constructions of information granules should be robust with respect to their input information granule deviations. In this way also a granulation of information granule constructions is considered. As the result we obtain the so-called AR schemes (AR networks) (see, e.g., papers by Skowron *et al.* in [11, 15, 16, 68]). AR schemes can be interpreted as complex patterns [64]. Searching methods for such patterns relevant for a given target concept have been developed [11]. Methods for deriving relevant AR schemes are of high computational complexity. The complexity can be substantially reduced by using domain knowledge. In such a case AR schemes are derived along reasoning schemes in natural language that are retrieved from domain knowledge. Developing methods for deriving such AR schemes is one of the main goals of our projects.

The outlined research directions create foundations toward understanding the nature of reasoning from measurements to perception. These foundations are crucial for constructing intelligent systems for many real-life projects.

In [124] we discuss the WGC as a basic methodology for Perception-based computing (PBC). By wisdom, we understand an adaptive ability to make judgments correctly to a satisfactory degree (in particular, correct decisions) having in mind real-life constraints. We propose in [124] RGC as the basis for WGC.

13.10 Conclusions

In this chapter we have presented basic concepts of rough set theory. We have also listed some research directions and exemplary applications based on the rough set approach to information granulation and granular computing.

A variety of methods for decision rule generation, reducts computation, and continuous variable discretization are very important issues not discussed here. We have only mentioned the methodology based on discernibility and Boolean reasoning for efficient computation of different entities including reducts and decision rules. For more details the reader is referred to [28, 88]. Several extensions of the rough set approach have been proposed in the literature [29]. In particular, it has been shown that the rough set approach can be used for synthesis and analysis of concept approximations in the distributed environment of intelligent agents. The relationships of rough set theory to many other theories has been extensively investigated. In particular, its relationships to fuzzy set theory, theory of evidence, Boolean reasoning methods, statistical methods, and decision theory have been clarified and seem now to be thoroughly understood. There are reports on many hybrid methods obtained by combining the rough set approach with other approaches such as fuzzy sets, neural networks, genetic algorithms, principal component analysis, and singular value decomposition. Many important research topics in rough set theory such as various logics related to rough sets and many advanced algebraic properties of rough sets were mentioned in the chapter only. The reader can find details in the books, articles, and journals cited in this chapter.

The rough set concept has led to its various generalizations. Some of them have been discussed in the article. Among extensions not discussed in this chapter is the rough set approach to multicriteria decision making (for references see, e.g., the papers by Słowiński *et al.* in [20] and in this handbook as well as the bibliography in [28–30]).

Recently, it has been shown that the rough set approach can be used for synthesis and analysis of concept approximations in the distributed environment of intelligent agents. We outlined the rough mereological

approach and its applications in calculi of information granules for synthesis of information granules satisfying a given specification to a satisfactory degree. We have also presented an approach to conflict analysis based on rough sets. This approach is based on properties of (in)discernibility. Finally, we have discussed a challenge for research on rough sets related to approximate reasoning from measurements to perception.

Acknowledgments

The research of Andrzej Skowron has been supported by the grant N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland and by the grant Innovative Economy Operational Programme 2007–2013 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland. The research by James Peters has been supported by the Engineering Research Council of Canada (NSERC) grant 185986.

References

- [1] K. Cios, W. Pedrycz, and R. Swiniarski. *Data Mining Methods for Knowledge Discovery*. Kluwer, Norwell, MA, 1998.
- [2] S. Demri and E. Orłowska (eds). *Incomplete Information: Structure, Inference, Complexity. Monographs in Theoretical Computer Science*. Springer-Verlag, Heidelberg, 2002.
- [3] B. Dunin-Kęplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds). *Monitoring, Security, and Rescue Tasks in Multiagent Systems (MSRAS'2004). Advances in Soft Computing*, Springer, Heidelberg, 2005.
- [4] I. Düntsch and G. Gediga. *Rough Set Data Analysis: A Road to Non-Invasive Knowledge Discovery*. Methodos Publishers, Bangor, UK, 2000.
- [5] J.W. Grzymała-Busse. *Managing Uncertainty in Expert Systems*. Kluwer, Norwell, MA, 1990.
- [6] M. Inuiguchi, S. Hirano, and S. Tsumoto (eds). *Rough Set Theory and Granular Computing. Studies in Fuzziness and Soft Computing*, Vol. 125. Springer-Verlag, Heidelberg, 2003.
- [7] B. Kostek. *Soft Computing in Acoustics, Applications of Neural Networks, Fuzzy Logic and Rough Sets to Physical Acoustics. Studies in Fuzziness and Soft Computing*, Vol. 31. Physica-Verlag, Heidelberg, 1999.
- [8] B. Kostek. *Perception-Based Data Processing in Acoustics: Applications to Music Information Retrieval and Psychophysiology of Hearing. Studies in Computational Intelligence*, Vol. 3. Springer, Heidelberg, 2005.
- [9] T.Y. Lin, Y.Y. Yao, and L.A. Zadeh (eds). *Rough Sets, Granular Computing and Data Mining. Studies in Fuzziness and Soft Computing*. Physica-Verlag, Heidelberg, 2001.
- [10] E. Orłowska (ed.). *Incomplete Information: Rough Set Analysis, Studies in Fuzziness and Soft Computing*, Vol. 13. Springer-Verlag/Physica-Verlag, Heidelberg, 1997.
- [11] S.K. Pal, L. Polkowski, and A. Skowron (eds). *Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies*. Springer-Verlag, Heidelberg, 2004.
- [12] S.K. Pal and A. Skowron (eds). *Rough Fuzzy Hybridization: A New Trend in Decision-Making*. Springer-Verlag, Singapore, 1999.
- [13] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory, Knowledge Engineering and Problem Solving*, Vol. 9. Kluwer, Dordrecht, The Netherlands, 1992.
- [14] L. Polkowski. *Rough Sets: Mathematical Foundations. Advances in Soft Computing*. Physica-Verlag, Heidelberg, 2002.
- [15] L. Polkowski, T.Y. Lin, and S. Tsumoto (eds). *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems. Studies in Fuzziness and Soft Computing*, Vol. 56. Springer-Verlag/Physica-Verlag, Heidelberg, 2000.
- [16] L. Polkowski and A. Skowron (eds). *Rough Sets in Knowledge Discovery 1: Methodology and Applications. Studies in Fuzziness and Soft Computing*, Vol. 18. Physica-Verlag, Heidelberg, 1998.
- [17] L. Polkowski and A. Skowron (eds). *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems. Studies in Fuzziness and Soft Computing*, Vol. 19. Physica-Verlag, Heidelberg, 1998.
- [18] A. Skowron (ed.). *Proceedings of the 5th Symposium on Computation Theory*, Zaborów, Poland, 1984, Lecture Notes in Computer Science, Vol. 208. Springer-Verlag, Berlin, 1985.
- [19] R. Słowiński (ed). *Intelligent Decision Support – Handbook of Applications and Advances of the Rough Sets Theory, System Theory, Knowledge Engineering and Problem Solving*, Vol. 11. Kluwer, Dordrecht, The Netherlands, 1992.
- [20] N. Zhong and J. Liu (eds). *Intelligent Technologies for Information Analysis*. Springer, Heidelberg, 2004.

- [21] J.F. Peters and A. Skowron (eds). *Transactions on Rough Sets I: Journal Subline, Lecture Notes in Computer Science*, Vol. 3100. Springer, Heidelberg, 2004.
- [22] J.F. Peters and A. Skowron (eds). *Transactions on Rough Sets III: Journal Subline, Lecture Notes in Computer Science*, Vol. 3400. Springer, Heidelberg, 2005.
- [23] J.F. Peters and A. Skowron (eds). *Transactions on Rough Sets IV: Journal Subline, Lecture Notes in Computer Science*, Vol. 3700. Springer, Heidelberg, 2005.
- [24] J.F. Peters, A. Skowron, D. Dubois, J.W. Grzymała-Busse, M. Inuiguchi, and L. Polkowski (eds). *Transactions on Rough Sets II: Journal Subline, Lecture Notes in Computer Science*, Vol. 3135. Springer, Heidelberg, 2004.
- [25] J.F. Peters and A. Skowron (eds). *Transactions on Rough Sets V: Journal Subline, Lecture Notes in Computer Science*, Vol. 4100. Springer, Heidelberg, 2006.
- [26] J.F. Peters, A. Skowron, I. Düntsch, J.W. Grzymała-Busse, E. Orłowska, and L. Polkowski (eds). *Transactions on Rough Sets VI: Journal Subline, Lecture Notes in Computer Science*, Vol. 4374. Springer, Heidelberg, 2007.
- [27] J.F. Peters, A. Skowron, V.M. Marek, E. Orłowska, R. Słowiński, and W. Ziarko (eds). *Transactions on Rough Sets VII: Journal Subline, Lecture Notes in Computer Science*, Vol. 4400. Springer, Heidelberg, 2007.
- [28] Z. Pawlak and A. Skowron. Rough sets and boolean reasoning. *Inf. Sci.* 177(1) (2007) 41–73.
- [29] Z. Pawlak and A. Skowron. Rough sets: Some extensions. *Inf. Sci.* 177(1) (2007) 28–40.
- [30] Z. Pawlak and A. Skowron. Rudiments of rough sets. *Inf. Sci.* 177(1) (2007) 3–27.
- [31] N. Cercone, A. Skowron, and N. Zhong (eds). Special issue: Rough sets, fuzzy sets, data mining, and granular soft computing. *Comput. Intell. Int. J.* 17(3). (2001).
- [32] T.Y. Lin (ed.). Special issue: Rough set theory. *J. Intell. Autom. Soft Comput.* 2(2) (1996).
- [33] J. Peters and A. Skowron (eds). Special issue: A rough set approach to reasoning about data. *Int. J. Intell. Syst.* 16(1) (2001).
- [34] S.K. Pal, W. Pedrycz, A. Skowron, and R. Swiniarski (eds). Special volume: Rough-neuro computing. *Neurocomputing* (36) (2001).
- [35] A. Skowron and S.K. Pal (eds). Special volume. Rough sets, pattern recognition and data mining. *Pattern Recognit. Lett.* 24(6) (2003). Special Volume.
- [36] R. Słowiński and J. Stefanowski (eds). Special issue: Proceedings of the First International Workshop on Rough Sets: State of the Art and Perspectives, Kiekrz, Poznań, Poland, September 2–4 (1992). *Found. Comput. Decis. Sci.* 18(3–4). (1993).
- [37] W. Ziarko (ed). Special issue: Rough sets and knowledge discovery. *Comput. Intell. Int. J.* 11(2) (1995).
- [38] W. Ziarko (ed). Special issue: Rough sets. *Fundam. Inf.* 27(2–3) (1996).
- [39] J.J. Alpigini, J.F. Peters, A. Skowron, and N. Zhong (eds). *Third International Conference on Rough Sets and Current Trends in Computing (RSCTC'2002)*, Malvern, PA, October 14–16, 2002, Lecture Notes in Artificial Intelligence, Vol. 2475. Springer-Verlag, Heidelberg, 2002.
- [40] S. Hirano, M. Inuiguchi, and S. Tsumoto (eds). *Proceedings of International Workshop on Rough Set Theory and Granular Computing (RSTGC'2001)*, Matsue, Shimane, Japan, May 20–22, 2001, Bulletin of the International Rough Set Society, Vol. 5(1–2). International Rough Set Society, Matsue, Shimane, 2001.
- [41] T.Y. Lin and A.M. Wildberger (eds). *Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery*. Simulation Councils, San Diego, CA, 1995.
- [42] L. Polkowski and A. Skowron (eds). *First International Conference on Rough Sets and Soft Computing RSCTC'1998, Lecture Notes in Artificial Intelligence*, Vol. 1424. Springer-Verlag, Warsaw, Poland, 1998.
- [43] A. Skowron, S. Ohsuga, and N. Zhong (eds). *Proceedings of the 7th International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing (RSFDGrC'99)*, Yamaguchi, November 9–11, 1999, Lecture Notes in Artificial Intelligence, Vol. 1711. Springer-Verlag, Heidelberg, 1999.
- [44] A. Skowron and M. Szczuka (eds). *Proceedings of the Workshop on Rough Sets in Knowledge Discovery and Soft Computing at ETAPS 2003*, April 12–13, 2003, Electronic Notes in Computer Science, Vol. 82(4). Elsevier, Amsterdam, The Netherlands, 2003. www.elsevier.nl/locate/entcs/volume82.html, accessed 2008.
- [45] D. Ślęzak, G. Wang, M. Szczuka, I. Düntsch, and Y. Yao (eds). *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2005)*, Regina, Canada, August 31–September 3, 2005, Part I, Lecture Notes in Artificial Intelligence, Vol. 3641. Springer-Verlag, Heidelberg, 2005.
- [46] D. Ślęzak, J.T. Yao, J.F. Peters, W. Ziarko, and X. Hu (eds). *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2005)*, Regina, Canada, August 31–September 3, 2005, Part II, Lecture Notes in Artificial Intelligence, Vol. 3642. Springer-Verlag, Heidelberg, 2005.
- [47] T. Terano, T. Nishida, A. Namatame, S. Tsumoto, Y. Ohsawa, and T. Washio (eds). *New Frontiers in Artificial Intelligence, Joint JSAI'2001 Workshop Post-Proceedings, Lecture Notes in Artificial Intelligence*, Vol. 2253. Springer-Verlag, Heidelberg, 2001.

- [48] S. Tsumoto, S. Kobayashi, T. Yokomori, H. Tanaka, and A. Nakamura (eds). *Proceedings of the The Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery*, November 6–8, 1996, University of Tokyo, Japan.
- [49] S. Tsumoto, R. Słowiński, J. Komorowski, and J. Grzymała-Busse (eds). *Proceedings of the 4th International Conference on Rough Sets and Current Trends in Computing (RSCTC'2004)*, Uppsala, Sweden, June 1–5, 2004, Lecture Notes in Artificial Intelligence, Vol. 3066. Springer-Verlag, Heidelberg, 2004.
- [50] G. Wang, Q. Liu, Y. Yao, and A. Skowron (eds). *Proceedings of the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2003)*, Chongqing, China, May 26–29, 2003, Lecture Notes in Artificial Intelligence, Vol. 2639. Springer-Verlag, Heidelberg, 2003.
- [51] W. Ziarko (ed.). *Rough Sets, Fuzzy Sets and Knowledge Discovery: Proceedings of the Second International Workshop on Rough Sets and Knowledge Discovery (RSKD'93)*, Banff, Alberta, Canada, October 12–15 (1993). Workshops in Computing, Springer-Verlag & British Computer Society, London/ Berlin, 1994.
- [52] W. Ziarko and Y. Yao (eds). *Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing (RSCTC'2000)*, Banff, Canada, October 16–19, 2000, Lecture Notes in Artificial Intelligence, Vol. 2005. Springer-Verlag, Heidelberg, 2001.
- [53] S. Greco, Y. Hata, S. Hirano, M. Inuiguchi, S. Miyamoto, H.S. Nguyen, and R. Słowiński (eds), *Proceedings of the Fifth International Conference on Rough Sets and Current Trends in Computing (RSCTC' 2006)*, Kobe, Japan, November 6–8, 2006, Lecture Notes in Artificial Intelligence, Vol. 4259. Springer-Verlag, Heidelberg, 2006.
- [54] J. Yao, P. Lingras, W.-Z. Wu, M. Szczuka, N. Cercone, and D. Ślęzak (eds). *Proceedings of the Second International Conference on Rough Sets and Knowledge Technology (RSKT 2007), Joint Rough Set Symposium (JRS 2007)*, Toronto May 14–16, 2007, Lecture Notes in Computer Science, Vol. 4481. Springer, Heidelberg, 2007.
- [55] A. An, J. Stefanowski, S. Ramanna, C.J. Butz, W. Pedrycz, and G. Wang (eds). *Proceedings of the Eleventh International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC 2007)*, Toronto, Canada, May 14–16, 2007, Lecture Notes in Computer Science, Vol. 4482. Springer, Heidelberg, 2007.
- [56] M. Kryszkiewicz, J.F. Peters, H. Rybiński, and A. Skowron (eds.). *Proceedings of the International Conference on Rough Sets and Intelligent Systems Paradigms (RSEISP 2007)*, Warsaw, Poland, June 28–30, 2007, Lecture Notes in Computer Science, Vol. 4585. Springer, Heidelberg, 2007.
- [57] Z. Pawlak. *Classification of Objects by Means of Attributes*. Reports, Vol. 429. Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland, 1981.
- [58] Z. Pawlak. *Rough Relations*. Reports, Vol. 435. Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland, 1981.
- [59] Z. Pawlak. Rough sets. *Int. J. Comput. Inf. Sci.* 11 (1982) 341–356.
- [60] L.A. Zadeh. A new direction in AI: Toward a computational theory of perceptions. *AI Mag.* 22(1) (2001) 73–84.
- [61] A. Skowron. Toward intelligent systems: Calculi of information granules. *Bull. Int. Rough Set Soc.* 5(1–2) (2001) 9–30.
- [62] A. Skowron and J. Stepaniuk. Information granules: Towards foundations of granular computing. *Int. J. Intell. Syst.* 16(1) (2001) 57–86.
- [63] R. Duda, P. Hart, and R. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2002.
- [64] W. Kloesgen and J. Zytkow (eds). *Handbook of Knowledge Discovery and Data Mining*. Oxford University Press, Oxford, 2002.
- [65] J.D. Ullman. *Principles of Database and Knowledge-Base Systems*, 1. Computer Science Press, MD, 1988.
- [66] S. Leśniewski. Grunzüge eines neuen Systems der Grundlagen der Mathematik. *Fundam. Math.* 14 (1929) 1–81.
- [67] E. Zermelo. Beweiss, dass jede menge wohlgeordnet werden kann. *Math. Ann.* 59 (1904) 514–516.
- [68] T.Y. Lin (ed.). Special issue: Rough sets. *Int. J. Approx. Reason.* 15(4) (1996).
- [69] R. Keefe. *Theories of Vagueness*. Cambridge Studies in Philosophy, Cambridge, UK, 2000.
- [70] A. Skowron. Rough sets and vague concepts. *Fundam. Inf.* 64(1–4) (2005) 417–431.
- [71] S. Read. *Thinking about Logic: An Introduction to the Philosophy of Logic*. Oxford University Press, Oxford/New York, 1994.
- [72] G. Frege. *Grundgesetzen der Arithmetik*, 2. Verlag von Hermann Pohle, Jena, 1903.
- [73] G.W. Leibniz. Discourse on metaphysics. In: G.W. Leibniz (ed), *Philosophical Texts*. Oxford University Press, New York, 1686, pp. 35–68.
- [74] J.F. Peters. Classification of objects by means of features. In: D. Fogel, G. Greenwood, and T. Cholewo (eds.), *Proceedings of the 2007 IEEE Symposium Series on Foundations of Computational Intelligence (IEEE SSCI 2007)*. IEEE, Honolulu, Hawaii, 2007, pp. 1–8.
- [75] Z. Pawlak. Information systems – theoretical foundations. *Inf. Syst.* 6 (1981) 205–218.

- [76] J.H. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, Heidelberg, 2001.
- [77] E. Orłowska and Z. Pawlak. *Expressive Power of Knowledge Representation System*. Reports, Vol. 432. Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland, 1981.
- [78] E. Konrad, E. Orłowska, and Z. Pawlak. *Knowledge Representation Systems*. Report 433. Institute for Computer Science, Polish Academy of Sciences, 1981.
- [79] J. Wróblewski. Theoretical foundations of order-based genetic algorithms. *Fundam. Inf.* 28 (1996) 423–430.
- [80] G. Gediga and I. Düntsch. Rough approximation quality revisited. *Artif. Intell.* 132 (2001) 219–234.
- [81] D. Śleżak. Approximate entropy reducts. *Fundam. Inf.* 53 (2002) 365–387.
- [82] W. Ziarko. Variable precision rough set model. *J. Comput. Syst. Sci.* 46 (1993) 39–59.
- [83] J. Rissanen. Modeling by shortes data description. *Automatica* 14 (1978) 465–471.
- [84] R. Słowiński and J. Stefanowski (eds). *Proceedings of the First International Workshop on Rough Sets: State of the Art and Perspectives*, Kiekrz, Poznań, Poland, September 2–4, 1992, 1999.
- [85] J. Lukasiewicz. Die logischen Grundlagen der Wahrscheinlichkeitsrechnung, 1913. In: L. Borkowski (ed), *Jan Lukasiewicz – Selected Works*. North-Holland, Amsterdam, London/ Polish Scientific Publishers, Warsaw, 1970, pp. 16–63.
- [86] T.M. Mitchel. *Machine Learning*. McGraw-Hill Series in Computer Science, Boston, MA, 1999.
- [87] D. Śleżak. Normalized decision functions and measures for inconsistent decision tables analysis. *Fundam. Inf.* 44 (2000) 291–319.
- [88] H.S. Nguyen. Approximate boolean reasoning: Foundations and applications in data mining. *Trans. Rough Sets V(LNCS 4100)* (2006) 334–506.
- [89] A. Skowron. Extracting laws from decision tables. *Comput. Intell. Int. J.* 11 (1995) 371–388.
- [90] D. Śleżak. Approximate reducts in decision tables. In: *Sixth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'1996*, Granada, Spain, 1996, Vol. III, pp. 1159–1164.
- [91] F. Brown. *Boolean Reasoning*. Kluwer, Dordrecht, 1990.
- [92] S.H. Nguyen and H.S. Nguyen. Some efficient algorithms for rough set methods. In: *Sixth International Conference on Information Processing and Management of Uncertainty on Knowledge Based Systems IPMU'1996*, Granada, Spain, 1996, Vol. III, pp. 1451–1456.
- [93] S.K. Pal, S. Bandopadhyay, and S. Biswas (eds). *Proceedings of the First International Conference on Pattern Recognition and Machine Intelligence (PRMI 2005)*, December 18–22, 2005, Indian Statistical Institute, Kolkata, Lecture Notes in Computer Science, Vol. 3776. Springer, Heidelberg, 2005.
- [94] A. Skowron and C. Rauszer. The discernibility matrices and functions in information systems. In: R. Słowiński (ed), *Intelligent Decision Support – Handbook of Applications and Advances of the Rough Sets Theory, Knowledge Engineering and Problem Solving*, Vol.11. Kluwer, Dordrecht, The Netherlands, 1992, pp. 331–362.
- [95] Z. Pawlak and A. Skowron. Rough membership functions. In: R. Yager, M. Fedrizzi, and J. Kacprzyk (eds), *Advances in the Dempster-Shafer Theory of Evidence*. John Wiley & Sons, New York, 1994, pp. 251–271.
- [96] L.A. Zadeh. Fuzzy sets. *Inf. Control* 8 (1965) 338–353.
- [97] D. Dubois and H. Prade. Foreword. In: *Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory, Knowledge Engineering and Problem Solving*, Vol. 9. Kluwer, Dordrecht, 1992.
- [98] S.K. Pal and P. Mitra. *Pattern Recognition Algorithms for Data Mining*. CRC Press, Boca Raton, FL, 2004.
- [99] Z. Pawlak. An inquiry into anatomy of conflicts. *J. Inf. Sci.* 109 (1998) 65–68.
- [100] R. Deja and A. Skowron. On some conflict models and conflict resolution. *Rom. J. Inf. Sci. Technol.* 5(1–2) (2002) 69–82.
- [101] R. Kowalski. A logic-based approach to conflict resolution. Report. Department of Computing, Imperial College, 2003, pp.1–28. <http://www.doc.ic.ac.uk/rak/papers/conflictresolution.pdf> accessed 2008.
- [102] A. Skowron, J. Stepaniuk, J. Peters, and R. Swiniarski. Calculi of approximation spaces. *Fundam. Inf.* 72(1–3) (2006) 363–378.
- [103] L. Breiman. Statistical modeling: The two cultures. *Stat. Sci.* 16(3) (2001) 199–231.
- [104] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.
- [105] A. Skowron. Rough sets in KDD – plenary talk. In: Z. Shi, B. Faltings, and M. Musen (eds), *16th World Computer Congress (IFIP'2000): Proceedings of Conference on Intelligent Information Processing (IIP'2000)*. Publishing House of Electronic Industry, Beijing, 2000, pp. 1–14.
- [106] T.Y. Lin and N. Cercone (eds). *Rough Sets and Data Mining – Analysis of Imperfect Data*. Kluwer, Boston, 1997.
- [107] Z. Pawlak. Decision rules, Bayes' rule and rough sets. In: A. Skowron, S. Ohsuga, and N. Zhong (eds), *Proceedings of the 7th International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft*

- Computing (RSFDGrC'99)*, Yamaguchi, November 9–11, 1999, Lecture Notes in Artificial Intelligence, Vol. 1711. Springer-Verlag, Heidelberg, 1999, pp.1–9.
- [108] D. Ślęzak and W. Ziarko. The investigation of the Bayesian rough set model. *Int. J. Approx. Reason.* 40 (2005) 81–91.
- [109] A. Skowron. Perception logic in intelligent systems (keynote talk). In: S. Blair, U. Chakraborty, S.-H. Chen, *et al.* (ed), *Proceedings of the 8th Joint Conference on Information Sciences (JCIS 2005)*, July 21–26, 2005, Salt Lake City, UT. X-CD Technologies: A Conference & Management Company, Ontario, 2005, pp. 1–5.
- [110] S. Staab and R. Studer (eds). *Handbook on Ontologies. International Handbooks on Information Systems*. Springer, Heidelberg, 2004.
- [111] A. Skowron and P. Synak. Complex patterns. *Fundam. Inf.* 60(1–4) (2004) 351–366.
- [112] D.M. Byrd and C.R. Cathern. Introduction to Risk Analysis: A Systematic Approach to Science-Based Decision Making. ABS Group, Rockville, MD, 2000.
- [113] A. Skowron and P. Synak. Reasoning in information maps. *Fundam. Inf.* 59(2–3) (2004) 241–259.
- [114] J. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1947.
- [115] P.C. Fishburn. *Utility Theory for Decision Making*. Robert E. Krieger, Huntington, NY, 1970.
- [116] J.S. Mill. *Ratiocinative and Inductive, Being a Connected View of the Principles of Evidence, and the Methods of Scientific Investigation*. Parker, Son, and Bourn, West Strand, London, 1862.
- [117] P. Doherty, W. Łukaszewicz, A. Skowron, and A. Szafas. *Knowledge Engineering: A Rough Set Approach, Studies in Fuzziness and Soft Computing*, Vol. 202. Springer, Heidelberg, 2006.
- [118] Z. Pawlak. Rough logic. *Bull. Pol. Acad. Sci. Tech. Sci.* 35(5–6) (1987) 253–258.
- [119] A. Jankowski and A. Skowron. Logic for artificial intelligence: A Rasiowa–Pawlak school perspective. In: A. Ehrenfeucht, W. Marek, and M. Srebrny (eds), *Andrzej Mostowski and Foundational Studies*. IOS Press, Amsterdam, 2007. To appear.
- [120] L.W. Barsalou. Perceptual symbol systems. *Behav. Brain Sci.* 22 (1999) 577–660.
- [121] T. Poggio and S. Smale. The mathematics of learning: Dealing with data. *Not. AMS* 50(5) (2003) 537–544.
- [122] P. Stone. *Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer*. The MIT Press, Cambridge, MA, 2000.
- [123] M. Gell-Mann. *The Quark and the Jaguar – Adventures in the Simple and the Complex*. Brown, London, 1994.
- [124] A. Jankowski and A. Skowron. Wisdom granular computing (WGC). In: W. Pedrycz, A. Skowron, and V. Kreinovich (eds.), *Handbook of Granular Computing*. Wiley & Sons, New York, 2007.

14

Wisdom Granular Computing

Andrzej Jankowski and Andrzej Skowron

Ask not what mathematics can do for biology; ask rather what biology can do for mathematics!

– Stanislaw Ulam (Adventures of a mathematician, Scribner, New York, 1976)

14.1 Introduction

Starting from the very beginning of written history, humans have speculated about the nature of mind, thought, and language. The first synthesis milestone in this domain has been done by Aristotle. In particular, he has initiated research in the area of ontology and formalization of philosophers' speculation by means of syllogistic logic. Classical and medieval grammarians explored more subtle features of language that Aristotle shortchanged. In the thirteenth-century Ramon Llull was the first to build *machines* that used logical means to produce knowledge by some kind of calculation. For today's computers one of the most important concept is Boolean algebra. This concept is based on analysis of properties of conjunction, disjunction, negation, and also relations to set concepts such as identity, set inclusion, and the empty set. It is worth to mention that investigation of concept's structure has been initiated by Gottfried Wilhelm Leibniz. In particular, he underlined that all our concepts are composed out of a very small number of basic concepts, which form the alphabet of human thoughts, and complex concepts are obtained from these basic concepts by a uniform and symmetrical combination, analogously to arithmetical operations such as multiplication or addition. In unpublished writings by Gottfried Wilhelm Leibniz (see the book [1] by Bertrand Russell) he developed logic to a level which was reached only 150 years later. Especially, we would like to recall the following sentences of Gottfried Wilhelm Leibniz [2, 3]:

If controversies were to arise, there would be no more need of disputation between two philosophers than between two accountants. For it would suffice to take their pencils in their hands, and say to each other: Let us calculate.

...

Languages are the best mirror of the human mind, and that a precise analysis of the signification of words would tell us more than anything else about the operations of the understanding.

Hence, Gottfried Wilhelm Leibniz should be considered a precursor of modern granular computing (GC) understood as a calculus of human thoughts. Through centuries since then, mathematicians have been developing tools to deal with this issue. Unfortunately, the developed tools in crisp mathematics, in particular, in classical mathematical logic do not yet allow for understanding natural language used by

human to express thoughts and reasoning about them, an understanding which will allow us to construct truly intelligent systems.

One of the reason is that humans, capable of solving efficiently many real-life problems, are able to express their thoughts by means of vague, uncertain, imprecise concepts and reason about such concepts. Lotfi Zadeh (see, e.g., [4]) proposed to base the calculus of thoughts using fuzzy logic to move from computing with numbers to computing with words, and further from manipulations of measurements to manipulations of perceptions. This idea has been developed by Lotfi Zadeh himself in a number of papers (see, e.g., [5, 6]) and by other researchers, using also rough set methods (see, e.g., [7]).

In [8–10] Lotfi Zadeh proposed the term ‘information granule’: *An information granule is a clump of objects of some sort, drawn together on the basis of indistinguishability, similarity, or functionality. In this definition, being general enough to comprise a large number of special cases, the stress is laid on the reasons for clustering objects into clumps, and three such motives are suggested: indistinguishability, similarity, and functionality.*

There are several papers on rough set theory in which an attempt has been made to develop methods for calculi of information granules (see, e.g., [7]).

In [11] wisdom technology (WisTech) is discussed as one of the main paradigms for development of new applications in intelligent systems. In this chapter, it is emphasized that in developing more advanced applications we are moving from data granules to information granules, then from information granules to knowledge granules, and finally from knowledge granules to wisdom granules.

We discuss all these granules with a special emphasis on wisdom granules. Calculi of such advanced granules are very important for making progress in the development of intelligent systems.

Solving complex problems, e.g., by multiagent systems (MAS) requires new approximate reasoning methods based on new computing paradigms. One such recently emerging computing paradigm is rough-granular computing (RGC). Computations in RGC are performed on information granules representing often vague, partially specified, and compound concepts delivered by agents engaged in tasks such as knowledge representation, communication with other agents, and reasoning.

The research on the foundations on WGC and, in particular, on RGC is based on a continuation of approaches to computational models of approximate reasoning developed by Rasiowa (see [12]), Pawlak (see [13]), and their students. In some sense, it is a succession of ideas initiated by Leibniz and Boole and is currently continued in a variety of forms. Of course, the Rasiowa–Pawlak school is also some kind of continuation of the Polish School of Mathematics and Logics, which led to the development of the modern understanding of the basic computational aspects of logic, epistemology, ontology, foundations of mathematics, and natural deduction. The two fundamental tools of the Rasiowa–Pawlak school are (i) *Computation models of logical concept (especially such concepts as deduction or algebraic many-valued models for classical, modal, and constructive mathematics)* – based on the method of treating the sets of logically equivalent statements (or formulas) as abstract algebras known as Lindebaum–Tarski algebras; (ii) *Computation models of vague concept* – originally Łukasiewicz has proposed to treat uncertainty (or vague concepts) as concepts of many-valued logic. The rough set concept, due to Pawlak [13], developed in the Rasiowa–Pawlak school, is based on classical two-valued logic. The rough set approach has been developed to deal with uncertainty and vagueness. The approach makes it possible to reason precisely about approximations of vague concepts. These approximations are temporary, subjective, and are adaptively changing with changes in environments [14–16].

This chapter is organized as follows. First, we outline WisTech in Section 14.2. In Section 14.3, we discuss a general concept of calculi of granules used in optimization processes in which the goals are achieved by performing computations on granules. Also a general definition of granule is discussed. Different kinds of granules are presented in Section 14.4. We start from data granules; next are presented information granules which are used to construct knowledge granules. The most advanced granules are wisdom granules on which adaptive judgment (AJ) is performed in solving real-life problems. Calculi of wisdom granules are discussed in Section 14.4.4. Examples of problems in developing such calculi are discussed. In particular, algorithmic methods for developing efficient calculi of wisdom granules for solving real-life problems are one of the main challenges of GC.

14.2 Wisdom Technology

In this section, we give a short introduction to WisTech. For more details on WisTech the reader is referred to [11].

There are many indications that we are currently witnessing the onset of an era of radical changes, depending on the further advancement of technology to acquire, represent, store, process, discover, communicate, and learn wisdom. In this chapter, we call this technology *wisdom technology* (or WisTech, for short). The term *wisdom* commonly means ‘rightly judging’ [17]. This common notion can be refined. By *wisdom*, we understand an adaptive ability to make judgments correctly to a satisfactory degree (in particular, correct decisions) having in mind real-life constraints.

One of the basic objectives of this chapter is to indicate the role of GC for the design and implementation of WisTech computation models. An important aspect of WisTech is that the complexity and uncertainty of real-life constraints mean that in practice we must reconcile ourselves to the fact that our judgments are based on non-crisp concepts and also do not take into account all the knowledge accumulated and available to us. This is why consequences of our judgments are usually imperfect. But as a consolation, we also learn to improve the quality of our judgments via observation and analysis of our experience during interaction with the environment. Satisfactory decision-making levels can be achieved as a result of improved judgments.

The intuitive nature of wisdom understood in this way can be expressed metaphorically as shown in (1):

$$Wisdom = KSN + AJ + IP, \quad (1)$$

where *KSN*, *AJ*, and *IP* denote *knowledge sources network*, *adaptive judgment*, and *interactive processes*, respectively. The combination of the technologies represented in (1) offers an intuitive starting point for a variety of approaches to designing and implementing computational models for WisTech. The equation (1) is called the *wisdom equation*. There are many ways to build WisTech computational models.

The issues discussed in this chapter are relevant for the current research directions (see, e.g., [18–25] and the literature cited in these articles).

Our approach to Wistech is based on RGC.

14.3 Calculi of Granules: General Comments

GC is required to achieve feasibility and efficiency in solving hard real-life problems, which are not solvable by traditional methods. Such problems are related to distributed systems of units (agents) interacting with each other and with dynamically changing environments. The information available to the agents about objects and concepts is usually only partial and concepts are often vague. Moreover, this information is dynamically changing, which requires that the developed methods should be adaptive to these changes.

Computations in GC are performed on granules. The aim of the computations is to reach a given goal. Usually, this is an optimization process [26].

GC can deal with problems specified by means of vague concepts. The solution constructed for them should satisfy a given specification to a satisfactory degree. Certainly, this creates problems but also, as Lotfi Zadeh observed, e.g., in the foreword to [7], *there is a tolerance for imprecision which can be exploited to achieve tractability, robustness, and low solution cost*. Moreover, for many problems *the expressive power of words is higher than the expressive power of numbers and/or the available information is not precise enough to justify the use of numbers*.

Computations in GC are performed under partial and uncertain information about objects and concepts which are often vague. This requires special modeling techniques for building models of concepts. These techniques allow us to induce relational and syntactical structures used to represent objects and to express their properties. The relevant structures should be discovered to assure the satisfactory approximation of the considered concepts. Concept approximations are constructed over semantical and syntactical structures.

Calculi of granules create the core for GC. They consist of atomic granules and operations making it possible to assemble more compound granules from already-constructed ones. The challenge is how, for given problems or a class of problems, such calculi can be developed. Granular calculi for WisTech are compound and they require further study. We would like to add here some comments on granule construction. New granules are often defined as a result of interaction between already-defined granules or their fusion. Numerous examples of operations on information granules can be found in [7, 11, 14, 15, 27–35]. These granules can be basic, such as indiscernibility classes, degrees of inclusion, or more advanced, such as classifiers, networks of classifiers, agents, or their coalitions.

In the literature on modeling of complex processes, one can find some extreme opinions about advantages of some granules (e.g., representing data models) over others. For example, some authors suggest that there exists a superiority of analog modeling over the discrete modeling (e.g., [36]). However, the choice of models depends on the application. Moreover, often it is convenient to use both modeling for different components of complex processes. It is hence necessary to discover fusion operations of such models in order to construct new relevant granules representing more complex processes obtained from such components. For some applications, it can be useful to consider granules with syntactical structure defined by differential equations (e.g., with some initial conditions) and their, e.g., continuous solutions representing the semantical structure. In modeling of complex processes, one should also be able to fuse such analog granules (models) with granules representing discrete models of processes or vague concepts, i.e., granules defined by description in natural language. Strategies for learning of such fusion operations are important for applications.

It will be worth to mention here a general scheme for the granule construction. This scheme consists of the following main steps:

- A new class \mathcal{M} of semantical structures is defined on the basis of the existing semantical structures of objects and attribute values sets (see, e.g., [28]).
- A language \mathcal{L} for description of properties of relational structures from \mathcal{M} is discovered (selected).
- Relevant properties (features, attributes) are selected from the language \mathcal{L} . They are used, e.g., as granules (patterns) in approximation of concepts.

All these steps require the discovery of new relevant semantical and syntactical structures and they are the main obstacles in approximation of complex concepts as it is known in machine learning, pattern recognition, and data mining. The discovery can be facilitated by domain knowledge (see, e.g., [29, 31–33, 37]). In searching for relevant structures, there is a necessity of continuous interaction between the two worlds: semantical and syntactical. Moreover, for adaptive systems, it is necessary to adaptively change semantical and syntactical structures in response to changes in the environments interacting with the system.

Observe that between semantical and syntactical structures there exists a kind of interaction or game analogous to reported in the research on cognition in MAS (see, e.g., [30, 38, 39]), where it is emphasized the role of a brain duality in reasoning by agents. In our discussion, we use two terms *intension* and *extension* of a formula (expression) in the usual logical sense. In reasoning about real world two kinds of reasoning are used. The first one is based on symbolic reasoning and the second one on scene reasoning (reasoning using semantical structures). These structures are changing by adopting information from sensors interacting with the environments. On the basis of examples of perceived objects some (perceptual) syntactical structures (e.g., formulas) defining on the set of examples their intensions are induced and next these syntactical structures are used, e.g., for matching objects using their extensions (i.e., meaning on an extension of the set of example). There is a feedback between these two structures. To illustrate this, let us consider one example. On the one hand, an agent can look for extensions that are general as much as possible, but on the other hand the generalization obtained in this way may not be relevant (e.g., the pattern defined by the extension may not be relevant for approximation of the considered concept). Then, a feedback could be sent to the syntactical structures calling for a reconstruction, e.g., of language in which properties are expressed aiming to discover more relevant descriptions of objects. This reconstruction should give new descriptions of objects, leading to relevant extensions. When relevant adaptation of syntactical structures becomes infeasible then a feedback can be sent to semantical structures with

requirement for reconstruction, i.e., a change in semantical perception of objects. In [30] is shown that this game between syntactical and semantical structures has deep roots in algebra, mathematical logic, and category theory, e.g., in Galois theory and in topos theory. In the described process, a lot of different kinds of granules are involved. Methods for modeling such granules and strategies for computations on such granules should be developed.

Intuitions illustrated here are basic for the implementation of GC in MAS, where one can imagine an *intelligent* agent equipped with two hemispheres (*of a brain*) – the left is used for describing things in a symbolic language and for symbolic reasoning, while the right hemisphere deals with imagining by the agent acceptable models satisfying certain features and with reasoning on possibilities of switching from one model to another (scene reasoning) [30]. Such an agent communicates with the world through sensors attached to both hemispheres of brain.

In particular, computations in GC should make it possible to reason from measurements by sensors to higher level concepts representing perception (see, e.g., [7]).

14.4 From Data Granules to Wisdom Granules

In [28] several kinds of granules and operations on granules have been discussed. In this section, we characterize granules from another perspective. We discuss granules corresponding to the diagram presented in [11] (see Figure 14.1) and discussed in the context of the wisdom equation (see (1)).

In this figure, there are four distinguished kinds of granules: data granules, information granules, knowledge granules, and wisdom granules. They correspond to technologies: database technology, information technology, knowledge management technology, and WisTech, respectively. In [11] the main features of these technologies were discussed. Here, we discuss granules corresponding to these technologies.

14.4.1 Data Granules

In Figure 14.1 the term ‘data’ is understood as a stream of symbols without any interpretation of their meaning.

More formally, one can define data granules assuming that there is a given relational structure over a given set V_a of values of a given attribute a [13, 40] or over the Cartesian product of such sets for a given set of attributes B , i.e., $\prod_{a \in B} V_a$. Then by taking a language of formulas of signature defined by this

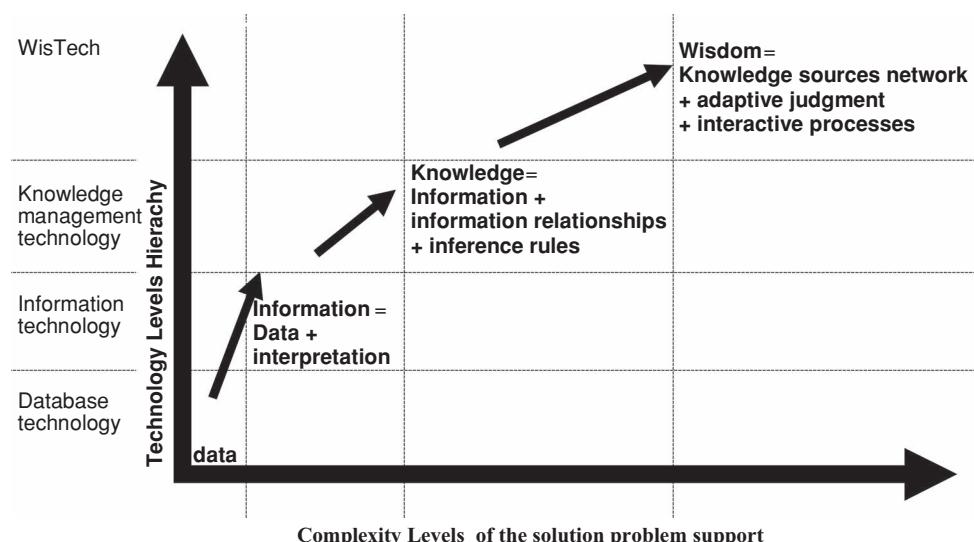


Figure 14.1 Wisdom equation context

relational structure, one can define subsets of $\prod_{a \in B} V_a$ equal to the semantic meaning of formulas with some free variable in the considered structure. These sets are data granules. For example, one can consider a linear order on V_a and define as data granules intervals. Having relational structures over V_a and V_b one can extend them on $V_a \times V_b$ by adding some constrain relations, e.g., in $V_a \times V_b$, representing, for example, the closeness of values from V_a and V_b if these sets are subsets of real numbers. Then formulas over the extended structure define new data granules.

14.4.2 Information Granules

Information granules are related to modeled objects. A typical example of information granules in rough sets are indiscernibility classes or similarity classes of objects defined by information systems or decision tables [13, 40]. Usually, these granules are obtained as follows. For example, if $\alpha(x)$ is a formula defining a data granule with values for x in the set of attribute value vectors then an information granule can be defined assuming that an object y (from the considered set of objects, e.g., a set of patients) is satisfying a formula $\alpha^*(y)$ if and only if the B -signature of this object $Inf_B(y)$ satisfies α [13, 40]. In this way, subsets of objects are defined, which are interpreted as information granules. Certainly, one can also consider families of information granules as an information granule. A typical example is a set of indiscernibility classes considered as an information granule.

14.4.3 Knowledge Granules

New knowledge granules are constructed by means of some operations from already-established granules. At the beginning some atomic (primitive) granules should be distinguished. In constructing knowledge granules inductive reasoning is also used, e.g., in constructing granules corresponding to classifiers (see, e.g., [28]).

One can consider different relationships between information granules. Quality measures for association rules [41, 42] in a given information system can be treated as a standard example of relations between information granules. Another example is given by relationships defined by given two families of indiscernibility classes creating a partition of the universe of objects. The relationships between such granules can be defined using positive regions [13, 40] or entropy [42–44].

One can consider different kinds of inference rules related to knowledge. These rules can be treated as schemes making it possible to derive properties of some knowledge granules from properties of other knowledge granules used for their construction. Dependencies between approximated concepts are examples of such rules. Another example is given by approximate reasoning schemes (AR schemes) (see, e.g., [27]) representing constructions of compound patterns from more elementary ones together with information about how degrees of inclusion of these elementary patterns in some input concepts are propagating to the degree of inclusion of the compound pattern in the target concept. Other examples of rules of inference on knowledge granules are considered in reasoning about knowledge [45].

It is worthwhile to mention that in GC there is no opposition between computational and dynamical approaches [36, 46]. For example, one can consider granules corresponding to dynamical models of some processes with the syntactical structure represented by differentials equation with initial conditions and the semantical structure representing solutions of equations. Moreover, one can consider interaction between such granules and other computational granules, e.g., representing discrete models of other processes. Results of such interactions are compositions of models of processes represented by interacting granules.

14.4.4 Wisdom Granules and Calculi of Wisdom Granules Based on RGC

Let us recall from [11] some general features of WisTech. From the perspective of the metaphor expressed in the wisdom equation (1), WisTech can be perceived as the integration of three technologies (corresponding to three components in the wisdom equation (1)). The first component is related to knowledge source networks:

Knowledge sources network (KSN) – By knowledge we traditionally understand every organized set of information along with the inference rules; in this context one can easily imagine the following examples illustrating the concept of KSN:

- Representations of states of reality perceived by our senses (or observed by the “receptors” of another observer) are integrated as a whole in our minds in a network of sources of knowledge and then stored in some part of our additional memory.
- A network of knowledge levels represented by agents in some MAS and the level of knowledge about the environment registered by means of receptors.

The third component *IP* of wisdom equation (1) is about *interactive processes* where interaction is understood as a sequence of stimuli and reactions over time; examples are

- the dialogue of two people,
- a sequence of actions and reactions between an unmanned aircraft and the environment in which the flight takes place, or
- a sequence of movements during some multiplayer game.

Far more difficult, conceptually, seems to be the second component of wisdom equation (1), i.e., adaptive judgment. Intuitions behind the AJ can be expressed as follows:

Adaptive judgment – understood here as arriving at decisions resulting from the evaluation of patterns observed in sample objects. This form of judgment is made possible by mechanisms in a metalanguage (meta-reasoning) which, on the basis of available knowledge sources’ selection and on the basis of understanding interactive processes’ history and their current status, enable us to perform the following activities under real-life constraints:

- identification and judgment of importance (for future judgment) of sample phenomena, available for observation, in the surrounding environment;
- planning current priorities for actions to be taken (in particular, on the basis of understanding interactive processes’ history and their current status) toward making optimal judgments;
- selection of fragments of ordered knowledge (hierarchies of information and judgment strategies) satisfactory for making a decision at the planned time (a decision here is understood as a commencing interaction with the environment or as selecting the future course to make judgments);
- prediction of important consequences of the planned interaction of processes;
- adaptive learning and, in particular, reaching conclusions deduced from patterns observed in sample objects leading to adaptive improvement in the AJ process.

One of the main barriers hindering an acceleration in the development of WisTech applications lies in developing satisfactory computation models implementing the functioning of ‘adaptive judgment.’ This difficulty primarily consists in overcoming the complexity of the process of integrating the local assimilation and processing of changing non-crisp and incomplete concepts necessary to make correct judgments. In other words, we are only able to model tested phenomena using local (subjective) models and interactions between them. In practical applications, usually, we are not able to give global models of analyzed phenomena (see, e.g., [47–52]). However, we can only approximate global models by integrating the various incomplete perspectives of problem perception. One of the potential computation models for ‘adaptive judgment’ might be the *rough-granular approach*.

Granule calculi consist of some atomic granules and operations on granules used for new granule generation. Wisdom granules have compound structure and are generated by sophisticated operations or strategies. Both wisdom granules and operations should be discovered and adaptively learned from data and domain knowledge.

Let us consider some examples of wisdom granules. Their different parts are responsible for different tasks such as interaction with other granules and environment, adaptive judgment, communication with knowledge networks, or optimization of different granule parameters.

Let us start from examples of granules resulting by interaction between granules. The simplest example is a decision table representing the result of interaction between condition and decision attributes or two agents observing the same objects. The next example concerns the interaction of a given object and a decision rule. The result of such an interaction is a matching degree of this object to the rule. Continuing this example one can consider the results of interaction of rule with decision class, object with classifier, the interaction of measurements by sensors with ontology of concepts, or behavioral patterns. One can also consider agents as compound granules and their interactions producing new granules. They can be related to actions, plans, agent cooperation or competition, coalition formation, etc. Interactions in MAS are intensively studied and are one of the main topics in MAS.

In adaptive judgment we need wisdom granules representing degrees of satisfiability of concepts represented in the granule, granules for expressing the propagation of satisfiability degrees from sensor measurements to higher level concepts representing perception, granules used for representation of reasoning about changes in adaptive learning of action selection or plan execution. For generating such granules advanced strategies should be developed. Such strategies are parts of wisdom granules. Adaptive judgment requires strategies for adaptive learning of concepts represented in wisdom granules. Hence, in particular, granules representing the discovered, by such strategies, semantical and syntactical structures, as well as granules representing discovered new patterns and concepts constructed out of the patterns, are needed.

Advanced strategies are required to extract from compound granules their relevant parts. In particular, advanced strategies are needed to extract, for a given granule representing the goal and for given granules representing knowledge networks, some granules representing relevant fragments of such knowledge networks. Analogous strategies should be developed for relational and syntactical structures considered in the context of a given goal for given sensor measurements.

14.5 Rough-Granular Computing

In this section we outline the RGC.

Developing methods for approximation of compound concepts expressing the result of perception belongs to the main challenges of perception-based computing. The perceived concepts are expressed in natural language. We discuss the rough-granular approach to approximation of such concepts from sensory data and domain knowledge. This additional knowledge, represented by ontology of concepts, is used to make it feasible the search for features (condition attributes) relevant for the approximation of concepts on different levels of the concept hierarchy defined by a given ontology. We report several experiments of the proposed methodology for approximation of compound concepts from sensory data and domain knowledge. The approach is illustrated by examples relative to interactions of agents, ontology approximation, adaptive hierarchical learning of compound concepts and skills, behavioral pattern identification, planning, conflict analysis and negotiations, and perception-based reasoning. The presented results seem to justify the following claim of Lotfi A. Zadeh:

In coming years, granular computing is likely to play an increasingly important role in scientific theories – especially in human-centric theories in which human judgement, perception and emotions are of pivotal importance.

The question how ontologies of concepts can be discovered from sensory data remains as one of the greatest challenges for many interdisciplinary projects on learning of concepts.

The concept approximation problem is the basic problem investigated in machine learning, pattern recognition, and data mining [42]. It is necessary to induce approximations of concepts (models of concepts) consistent (or almost consistent) with some constraints. In the most typical case, the constraints are defined by a training sample. For more compound concepts, we consider constraints defined by domain ontology consisting of vague concepts and dependencies between them. Information about the classified objects and concepts is partial. In the most general case, the adaptive approximation of concepts is

performed under interaction with dynamically changing environment. In all these cases, searching for suboptimal models relative to the minimal length principle (MLP) is performed. Notice that in adaptive concept approximation one of the components of the model should be the adaptation strategy. Components involved in construction of concept approximation which are tuned in searching for suboptimal models relative to MLP are called information granules. In RGC, information granule calculi are used for construction of components of classifiers and classifiers themselves (see, e.g., [16]) satisfying given constraints. An important mechanism in RGC is related to generalization schemes, making it possible to construct more compound patterns from less compound patterns. Generalization degrees of schemes are tuned using, e.g., some evolutionary strategies.

Rough set theory due to Zdzisław Pawlak [13, 34, 40, 43, 44] is a mathematical approach to imperfect knowledge. The problem of imperfect knowledge has been tackled for a long time by philosophers, logicians, and mathematicians. Recently, it also became a crucial issue for computer scientists, particularly in the area of artificial intelligence. There are many approaches to the problem of how to understand and manipulate imperfect knowledge. The most successful one is, no doubt, the fuzzy set theory proposed by Lotfi A. Zadeh [53]. Rough set theory presents still another attempt to solve this problem. It is based on an assumption that objects and concepts are perceived by partial information about them. Due to this some objects can be indiscernible. From this fact it follows that some sets cannot be exactly described by available information about objects; they are rough, not crisp. Any rough set is characterized by its (lower and upper) approximations. The difference between the upper and lower approximation of a given set is called its boundary. Rough set theory expresses vagueness by employing a boundary region of a set. If the boundary region of a set is empty, it means that the set is crisp; otherwise the set is rough (inexact). A non-empty boundary region of a set indicates that our knowledge about the set is not sufficient to define the set precisely. One can recognize that rough set theory is, in a sense, a formalization of the idea presented by Gotlob Frege [54].

One of the consequences of perceiving objects using only available information about them is that for some objects one cannot decide if they belong to a given set or not. However, one can estimate the degree to which objects belong to sets. This is another crucial observation in building the foundations for approximate reasoning. In dealing with imperfect knowledge one can only characterize satisfiability of relations between objects to a degree, not precisely. Among relations on objects the rough inclusion relation, which describes to what degree objects are parts of other objects, plays a special role. A rough mereological approach (see, e.g., [7, 27, 55]) is an extension of the Leśniewski mereology [56] and is based on the relation *to be a part to a degree*. It will be interesting to note here that Jan Łukasiewicz was the first who started to investigate the inclusion to a degree of concepts in his discussion on relationships between probability and logical calculi [57].

The very successful technique for rough set methods was Boolean reasoning [58]. The idea of Boolean reasoning is based on construction for a given problem P a corresponding Boolean function f_P with the following property: the solutions for the problem P can be decoded from prime implicants of the Boolean function f_P . It is worth to mention that to solve real-life problems it is necessary to deal with Boolean functions having a large number of variables.

A successful methodology based on the discernibility of objects and Boolean reasoning has been developed in rough set theory for computing of many key constructs like reducts and their approximations, decision rules, association rules, discretization of real-value attributes, symbolic value grouping, searching for new features defined by oblique hyperplanes or higher order surfaces, pattern extraction from data, as well as conflict resolution or negotiation [44, 59, 60]. Most of the problems involving the computation of these entities are NP-complete or NP-hard. However, it was possible to develop efficient heuristics yielding suboptimal solutions for these problems. The results of experiments on many data sets are very promising. They show very good quality of solutions generated by the heuristics in comparison with other methods reported in literature (e.g., with respect to the classification quality of unseen objects). Moreover, they are very time efficient. It is important to note that the methodology makes it possible to construct heuristics having a very important approximation property. Namely, expressions generated by heuristics (i.e., implicants) close to prime implicants define approximate solutions for the problem (see, e.g., [61]).

The rough set approach offers tools for approximate reasoning in MAS. The typical example is the approximation by one agent of concepts of another agent. The approximation of a concept is based on a decision table representing information about objects perceived by both agents.

The strategies for data models inducing developed so far are often not satisfactory for approximation of compound concepts that occur in the perception process. Researchers from the different areas have recognized the need for developing of new methods for concept approximation (see, e.g., [62, 63]). The main reason for this is that these compound concepts are, in a sense, too far from measurements, which makes the searching for relevant features infeasible in a very huge space. There are several research directions aiming at overcoming this difficulty. One of them is based on the interdisciplinary research where the knowledge pertaining to perception in psychology or neuroscience is used to help deal with compound concepts (see, e.g., [20, 64, 65]). There is a great effort in neuroscience toward understanding the hierarchical structures of neural networks in living organisms [64, 66, 67]. Also mathematicians are recognizing problems of learning as the main problem of the current century [67]. These problems are closely related to complex system modeling as well. In such systems again the problem of concept approximation and its role in reasoning about perceptions is one of the challenges nowadays. One should take into account that modeling complex phenomena entails the use of local models (captured by local agents, if one would like to use the multiagent terminology [47, 50, 51]) that should be fused afterward. This process involves negotiations between agents [47, 50, 51] to resolve contradictions and conflicts in local modeling. This kind of modeling is becoming more and more important in dealing with complex real-life phenomena which we are unable to model using traditional analytical approaches. The latter approaches lead to exact models. However, the necessary assumptions used to develop them result in solutions that are too far from reality to be accepted. New methods or even a new science should therefore be developed for such modeling [68].

One of the possible approaches in developing methods for compound concept approximations can be based on the layered (hierarchical) learning [69, 70]. Inducing concept approximation should be developed hierarchically starting from concepts that can be directly approximated using sensor measurements toward compound target concepts related to perception. This general idea can be realized using additional domain knowledge represented in natural language. For example, one can use some rules of behavior on the roads, expressed in natural language, to assess from recordings (made, e.g., by camera and other sensors) of actual traffic situations if a particular situation is safe or not (see, e.g., [29, 31–34]). The hierarchical learning has also been used for identification of risk patterns in medical data and extended for therapy planning (see, e.g. [37, 71]). Another application of hierarchical learning for sunspot classification is reported in [72]. To deal with such problems one should develop methods for concept approximations together with methods aiming at approximation of reasoning schemes (over such concepts) expressed in natural language. The foundations of such an approach, creating a core of perception logic, are based on rough set theory [13, 34, 40, 43, 44] and its extension rough mereology [7, 27, 55]. The (approximate) Boolean reasoning methods can be scaled to the case of compound concept approximation.

In the following section, we discuss more examples.

14.6 Some Solutions Based on WGC and Challenges for WGC

The prediction of behavioral patterns of a compound object evaluated over time is usually based on some historical knowledge representation used to store information about changes in relevant features or parameters. This information is usually represented as a data set and has to be collected during long-term observation of a complex dynamic system. For example, in case of road traffic, we associate the object-vehicle parameters with the readouts of different measuring devices or technical equipment placed inside the vehicle or in the outside environment (e.g., alongside the road, in a helicopter observing the situation on the road, or in a traffic patrol vehicle). Many monitoring devices serve as informative sensors, such as global positioning system, laser scanners, thermometers, range finders, digital cameras, radar, and image and sound converters (see, e.g. [73]). Hence, many vehicle features serve as models of physical sensors. Here are some exemplary sensors: location, speed, current acceleration or deceleration, visibility, and humidity (slipperiness) of the road. By analogy to this example, many features of compound objects are often dubbed sensors. Some rough set tools been developed (see,

e.g., [32, 33]) for perception modeling that make it possible to recognize behavioral patterns of objects and their parts changing over time. More complex behavior of compound objects or groups of compound objects can be presented in the form of *behavioral graphs*. Any behavioral graph can be interpreted as a *behavioral pattern* and can be used as a complex classifier for recognition of complex behaviours. The complete approach to the perception of behavioral patterns, based on behavioral graphs and the dynamic elimination of behavioral patterns, is presented in [32, 33]. The tools for dynamic elimination of behavioral patterns are used for switching off in the *system attention* procedures searching for identification of some behavioral patterns. The developed rough set tools for perception modeling are used to model networks of classifiers. Such networks make it possible to recognize behavioral patterns of objects changing over time. They are constructed using an ontology of concepts provided by experts that engage in approximate reasoning on concepts embedded in such an ontology. Experiments on data from a vehicular traffic simulator [74] show that the developed methods are useful in the identification of behavioral patterns.

The following example concerns human–computer interfaces that allow for interactions with experts in dialogs with them to transfer to the system their knowledge about structurally compound objects. For pattern recognition systems [75], e.g., for optical character recognition systems it will be helpful to transfer to the system a certain knowledge about the expert view on borderline cases. The central issue in such pattern recognition systems is the construction of classifiers within vast and poorly understood search spaces, which is a very difficult task. Nonetheless, this process can be greatly enhanced with knowledge about the investigated objects provided by a human expert. It was developed a framework for the transfer of such knowledge from the expert and for incorporating it into the learning process of a recognition system using methods based on rough mereology (see, e.g., [76]). It is also demonstrated how this knowledge acquisition can be conducted in an interactive manner, with a large data set of handwritten digits as an example.

The next two examples are related to approximation of compound concepts in reinforcement learning and planning. These examples are important for approximate reasoning in distributed environments.

In reinforcement learning [16, 77–83], the main task is to learn the approximation of the function $Q(s, a)$, where s and a denote a global state of the system and an action performed by an agent ag , respectively, and the real value of $Q(s, a)$ describes the reward for executing the action a in the state s . In approximation of the function $Q(s, a)$ probabilistic models are used. However, for compound real-life problems it may be hard to build such models for such a compound concept as $Q(s, a)$ [63]. We propose another approach to the approximation of $Q(s, a)$ based on ontology approximation. The approach is based on the assumption that in a dialog with experts an additional knowledge can be acquired, making it possible to create a ranking of values $Q(s, a)$ for different actions a in a given state s . In the explanation given by expert about possible values of $Q(s, a)$, concepts from a special ontology are used. Then, using this ontology one can follow hierarchical learning methods to learn the approximations of concepts from ontology. Such concepts can have a temporal character too. This means that the ranking of actions may depend not only on the actual action and the state but also on actions performed in the past and changes caused by these actions.

In [37, 71] a computer tool based on rough sets for supporting automated planning of the medical treatment (see, e.g., [84, 85]) is discussed. In this approach, a given patient is treated as an investigated complex dynamical system, while diseases of this patient (respiratory distress syndrome, patent ductus arteriosus, sepsis, ureaplasma, and respiratory failure) are treated as compound objects changing and interacting over time. As a measure of planning success (or failure) in experiments, is used a special hierarchical classifier that can predict the similarity between two plans as a number between 0.0 and 1.0. This classifier has been constructed on the basis of the special ontology specified by human experts and data sets. It is important to mention that besides the ontology, experts provided the exemplary data (values of attributes) for the purpose of concepts approximation from the ontology. The methods of construction of such classifiers are based on AR schemes and were described, e.g., in [29, 31–33]. This method was applied for approximation of similarity between plans generated in automated planning and plans proposed by human experts during the realistic clinical treatment.

One can observe an analogous problem in learning discernibility between compound objects. For example, let us consider a state of a complex system in two close moments of time: $x(t + \Delta t)$ and $x(t)$.

The *difference* between these two states cannot be described using numbers, but *words* [4]; i.e., it can be expressed in terms of concepts from a special ontology of (vague) concepts (e.g., acquired from a domain expert). These concepts should be again approximated using a relevant language and then in terms of the induced approximations the *difference* between $x(t + \Delta t)$ and $x(t)$ can be characterized and next used in decision making.

One of the WGC (and in particular, RGC) challenges is to develop approximate reasoning techniques for reasoning about dynamics of distributed systems of judges, i.e., agents judging rightly [17]. These techniques should be based on systems of evolving local perception logics (i.e., logics of agents or teams of agents) rather than on a global logic [35, 86]. The approximate reasoning about global behavior of judge's system is infeasible without methods for approximation of compound vague concepts and approximate reasoning about them. One can observe here an analogy to phenomena related to the emergent patterns in complex adaptive systems [52]. Let us observe that judges can be organized into a hierarchical structure; i.e., one judge can represent a coalition of judges in interaction with other agents existing in the environment [48, 87, 88]. Such judges representing coalitions play an important role in hierarchical reasoning about behavior of judge populations. Strategies for coalition formation and cooperation [48, 49, 87] are of critical importance in designing systems of judges with dynamics adhering to the given specification to a satisfactory degree. Developing strategies for discovery of information granules representing relevant coalitions and cooperation protocols is a challenge for RGC.

The problems discussed above can be treated as problems of searching for information granules satisfying vague requirements (constraints, specification). The strategies for construction of information granules should be adaptive. It means that the adaptive strategies should make it possible to construct information granules satisfying constraints under dynamically changing environment. This requires reconstruction or tuning of already-constructed information granules which are used as components of data models or the whole models, e.g., classifiers. In the adaptive process, the construction of information granules generalizing some previously constructed ones plays a special role. The mechanism for relevant generalization is crucial. One can imagine for this task many different strategies, e.g., based on adaptive feedback control for tuning the generalization. Cooperation with specialists from different areas such as neuroscience (see, e.g., [64] for visual objects recognition), psychology (see, e.g., [67] for discovery of mechanisms for hierarchical perception), biology (see, e.g., [89] for cooperation based on swarm intelligence), or social science (see, e.g., [48] for modeling of agents behavior) can help discover such adaptive strategies for extracting suboptimal (relative to the MLP) data models satisfying vague constraints. This research may also help us develop strategies for the discovery of ontologies relevant for compound concept approximation.

Another challenge for WGC concerns developing methods for the discovery of structures of complex processes from data (see, e.g., [90–92]), in particular, from temporal data (see, e.g., [29, 32, 33, 93–95]). One of the important problems in discovering structures of complex processes from data is to correctly identify the type of the relevant process, e.g., continuous type of processes modeled by differential equations with some parameters [36], discrete models represented by behavioral graphs (see, e.g., [29, 32, 33]), or models of concurrent processes, e.g. Petri nets (see, e.g., [91, 92]). When a type is selected, usually some optimization of parameters is performed to obtain a relevant structure model. For example, structures of behavioral graphs (see, e.g., [29, 32, 33, 37]) are obtained by composition of temporal patterns discovered from data. So far, in the experiments this process was facilitated by domain experts. However, we plan to develop heuristics searching for such behavioral graphs directly from data without such expert support. The behavioral graphs are granules related to a given level of hierarchy of granules. Properties of such granules representing behavioral patterns are used to define indiscernibility classes or similarity classes of behavioral graphs (processes). From these granules and relations between them (induced from relational structures on values sets of attributes defined by behavioral patterns [28]), new behavioral graphs are modeled. In this way, new granules on a higher level of granule hierarchy are obtained. They may represent, e.g., behavioral graphs of more compound groups of objects [96]. Patterns (properties) defined by these more compound behavioral graphs (on the basis of their components and interaction between them) are important in recognition or prediction of properties of complex processes. In particular, learning interactions which lead to the relevant behavioral patterns for compound processes is a challenge.

Let us consider one more challenge for WGC. We assume that there is given a distributed system of locally interacting agents (parts). These interactions lead to the global pattern represented by means of an *emergent patterns* (see, e.g., [52, 97]). One can model such a system as a special kind of game in which each agent can only interact with agents from its local neighborhood. The strategy of each agent in the game is defined by its rules of interaction with neighbors. It is well known that the emergent patterns are very hard to predict. There are many complex real-life systems in which such patterns are observed, e.g., in ecological systems, immune systems, economies, global climate, ant colonies [98, 99]. The challenge we consider is related to developing strategies for learning local interactions among agents leading to a given emergent pattern. The evolutionary strategies are here the candidates (see, e.g., [97–99]). We would like to emphasize the role of WGC in learning such strategies. One possible approach can be based on the assumption that the learning process should be organized in such a way that gradually learned interactions lead to granules of agents represented by coalitions of agents, rather than by particular agents. The behavior of each coalition is determined by some specific interactions of its agents which lead to interaction of the coalition as a whole with other agents or coalitions. Learning a hierarchy of coalitions is hard but can be made feasible by using domain knowledge, analogously to hierarchical learning (see, e.g., [32, 33, 37]). Domain knowledge can help discover languages for expressing behavioral patterns (properties) of coalitions on each level of coalition hierarchy and heuristics for identifying such behavioral patterns from sensor measurements. Without such ‘hints’ based on domain knowledge the learning seems to be infeasible for real-life problems. Using this approach based on hierarchy of coalitions, the evolutionary strategies should only make it possible to learn interactions on each level of hierarchy between existing on this level coalitions leading to discovery on the next-level relevant new coalitions and their properties. Let us observe that the discovered coalition by relevant granulation of lower level coalitions will usually satisfy the specification for this higher coalition level only to a degree. The discovered coalitions, i.e., new granules, can be treated as approximate solutions on a given level of hierarchy. It is worthwhile to mention another possibility related to learning relevant hierarchical coalitions for gradually changing tasks for the whole system, from simple to more compound tasks.

We summarize the above discussion as follows. One may try to describe emergent patterns in terms of interactions between some high-level granules – coalitions. We assume that the construction of relevant compound coalitions can be supported by domain ontology. Then, coalitions are discovered gradually using the hierarchical structure defined by the domain ontology. On each level of hierarchy a relevant language should be discovered. In this language, it should be possible to much easierly express properties of coalitions (e.g., using interactions between teams of agents) than using the language accessible on the preceding level where only more detailed descriptions are accessible (e.g., interactions between single agents). Hence, the same coalition can be described using different languages from two successive levels. However, the language from a higher level is more relevant for further modeling of more compound coalitions. Discovery of a relevant language for each level of hierarchy from languages on the proceeding levels may be feasible. However, discovery of a relevant language for modeling of compound coalitions on the highest hierarchy level directly from the very basic level (where, e.g., only interactions between single agents are directly expressible) may be unfeasible.

Finally, let us observe that the following four principles of adaptive information processing in decentralized systems presented in [97] are also central for WGC and RGC in searching for relevant granules by using evolutionary techniques:

1. Global information is encoded as statistics and dynamics of patterns over the system’s components.
2. Randomness and probabilities are essential.
3. The system carries out a fine-grained, parallel search of possibilities.
4. The system exhibits a continual interplay of bottom-up and top-down processes.

14.7 Conclusion

We presented WGC as a basic methodology in WisTech. Different kinds of granules, from data granules to the most compound wisdom granules, were discussed. Several important features of granules were

distinguished, such as size, diversity, structure with parts, type, ability to interact with other granules, and adaptiveness. The important role of RGC for solving problems related to WisTech was discussed. Conclusions from the current projects for WGC and RGC were reported and some challenges for WGC and RGC were included.

In the current projects, we are developing RGC methods on which WisTech can be based. The developed methods are used to construct wisdom engines. By wisdom engine we understand a system which implements the concept of wisdom. We plan to design specific systems for some tasks such as (1) intelligent document manager; (2) job market search; (3) brand monitoring; (4) decision support for global management systems (e.g., world forex, stock market, and world tourist); (5) intelligent assistant (e.g., physician, and lawyer); (6) discovery of processes from data (e.g., gene expression networks); and (7) rescue system (for more details see [11, 51]).

Acknowledgments

The research was supported by the grant N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland and by the grant Innovative Economy Operational Programme 2007–2013 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland . Moreover, the research of Andrzej Jankowski was supported by Institute of Decision Process Support.

References

- [1] B. Russell. *History of Western Philosophy*. Allen & Unwin, Great Britain, London, 1946.
- [2] G.W. Leibniz, *Dissertatio de Arte Combinatoria*, Leipzig, 1666.
- [3] G.W. Leibniz. *New Essays on Human Understanding* (1705). Translated and edited by Peter Remnant and Jonathan Bennett. Cambridge UK, Cambridge, 1982.
- [4] L.A. Zadeh. From computing with numbers to computing with words – From manipulation of measurements to manipulation of perceptions. *IEEE Trans. Circuits Syst.* 45 (1999) 105–119.
- [5] L.A. Zadeh. A new direction in AI: toward a computational theory of perceptions. *AI Mag.* 22(1) (2001) 73–84.
- [6] L.A. Zadeh. Toward a generalized theory of uncertainty (GTU) – an outline. *Inf. Sci.* 171 (2005) 1–40.
- [7] S.K. Pal, L. Polkowski, and A. Skowron (eds). *Rough-Neural Computing: Techniques for Computing with Words, Cognitive Technologies*. Springer, Heidelberg, 2004.
- [8] L.A. Zadeh. Outline of a new approach to the analysis of complex system and decision processes. *IEEE Trans. Syst. Man Cybern.* 3 (1973) 28–44.
- [9] L.A. Zadeh. Fuzzy sets and information granularity. In: M. Gupta, R. Ragade, and R. Yager (eds), *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, 1979, pp. 3–18.
- [10] L.A. Zadeh. Toward a theory of fuzzy information granulation and its certainty in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.
- [11] A. Jankowski and A. Skowron. A wistech paradigm for intelligent systems. In: *Transactions on Rough Sets VI: Journal Subline*. LNCS 4374. Springer, Heidelberg, 2006, pp. 94–132.
- [12] H. Rasiowa. *Algebraic Models of Logics*. Warsaw University, Warsaw, 2001.
- [13] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory, Knowledge Engineering and Problem Solving*. Kluwer Academic Publishers, Dordrecht, The Netherlands, Vol. 9, 1991.
- [14] J. Bazan, A. Skowron, and R. Swiniarski. Rough sets and vague concept approximation: from sample approximation to adaptive learning. In: *Transactions on Rough Sets V: Journal Subline*. LNCS 4100, Springer, Heidelberg, 2006, pp. 39–62.
- [15] A. Skowron. Rough sets and vague concepts. *Fundam. Inf.* 64(1–4) (2005) 417–431.
- [16] A. Skowron, J. Stepaniuk, J.F. Peters, and R. Swiniarski. Calculi of approximation spaces. *Fundam. Inf.* 72(1–3) (2006) 363–378.
- [17] S. Johnson. *Dictionary of the English Language in Which the Words Are Deduced from Their Originals, and Illustrated in Their Different Significations by Examples from the Best Writers*, 2 Vols. F.C. and J. Rivington, London, 1816.
- [18] N.L. Cassimatis, E.T. Mueller, and P.H. Winston. Achieving human-level intelligence through integrated systems and research. *AI Mag.* 27 (2006) 12–14.

- [19] N.L. Cassimatis. A cognitive substrate for achieving human-level intelligence. *AI Mag.* 27 (2006) 45–56.
- [20] K.D. Forbus and T.R. Hinrichs. Companion cognitive systems: a step toward human-level AI. *AI Mag.* 27 (2006) 83–95.
- [21] R. Granger. Engines of the brain: the computational instruction set of human cognition. *AI Mag.* 27(2) (2006) 15–31.
- [22] R.M. Jones and R.E. Wray. Comparative analysis of frameworks for knowledge-intensive intelligent agents. *AI Mag.* 27(2) (2006) 57–70.
- [23] C. Schlenoff, J. Albus, E. Messina, A.J. Barbera, R. Madhavan, and S. Balakirsky. Using 4d/rcs to address ai knowledge integration. *AI Mag.* 27 (2006) 71–81.
- [24] W. Swartout, J. Gratch, R.W. Hill, E. Hovy, S. Marsella, J. Rickel, and D. Traum. Towards virtual humans. *AI Mag.* 27 (2006) 96–108.
- [25] P. Langley. Cognitive architectures and general intelligent systems. *AI Mag.* 27 (2006) 33–44.
- [26] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer Academic Publishers, Dordrecht, 2003.
- [27] A. Skowron and J. Stepaniuk. Information granules and rough-neural computing. In: S.K. Pal, L. Polkowski, and A. Skowron (eds). *Rough-Neural Computing: Techniques for Computing with Words, Cognitive Technologies*. Springer, Heidelberg, 2004, pp. 43–84.
- [28] A. Skowron and J. Stepaniuk. Rough sets and granular computing: toward rough-granular computing. In: *Handbook of Granular Computing*. Wiley, UK, 2008.
- [29] J.G. Bazan and A. Skowron. Classifiers based on approximate reasoning schemes. In: B. Dunin-Kęplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds), *Monitoring, Security, and Rescue Tasks in Multiagent Systems (MSRAS'2004). Advances in Soft Computing*. Springer, Heidelberg, 2005, pp. 191–202.
- [30] A. Jankowski and A. Skowron. Logic for artificial intelligence: the Rasiowa - Pawlak school perspective. In: A. Ehrenfeucht, V. Marek, and M. Srebrny (eds), Andrzej Mostowski and Foundational Studies. IOS Press, Amsterdam, 2008, to appear.
- [31] H.S. Nguyen, J. Bazan, A. Skowron, and S.H. Nguyen. Layered learning for concept synthesis. In: *Transactions on Rough Sets I: Journal Subline*. LNCS 3100, Springer, Heidelberg, 2006, pp. 187–208.
- [32] J.G. Bazan, J.F. Peters, and A. Skowron. Behavioral pattern identification through rough set modelling. In: D. Ślęzak, J.T. Yao, J.F. Peters, W. Ziarko, and X. Hu (eds), *Proceedings of the 10th International Conference on Rough Sets, fuzzy sets, Data mining, and Granular Computing (RSFDGrC'2005)*, Regena, Canada, August 31– September 3, 2005, part II, LNAI 3642, Springer, Heidelberg, 2005, pp. 688–697.
- [33] D. Ślęzak, J.T. Yao, J.F. Peters, W. Ziarko, and X. Hu (eds), *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2005)*, Regina, Canada, August 31–September 3, 2005, Part II, LNAI 3642. Springer, Heidelberg, 2005.
- [34] P. Doherty, W. Łukaszewicz, A. Skowron, and A. Szalas. *Knowledge Representation Techniques: A Rough Set Approach, Studies in Fuzziness and Soft Computing*. Springer, Heidelberg, Germany, Vol 202, 2006.
- [35] A. Skowron. Rough sets in perception-based computing (keynote talk). In: S.K. Pal, S. Bandyopadhyay, and S. Biswas (eds), *First International Conference on Pattern Recognition and Machine Intelligence (PReMI'05)*, December 18–22, 2005, Indian Statistical Institute, Kolkata, LNCS 3776, Springer, Heidelberg, 2005, pp. 21–29.
- [36] V.G. Ivancevic and T.T. Ivancevic. *Geometrical Dynamics of Complex Systems. A Unified Modelling Approach to Physics, Control, Biomechanics, Neurodynamics and Psycho-Socio-Economical Dynamics*. Springer, Dordrecht, 2006.
- [37] J. Bazan, P. Kruczak, S. Bazan-Socha, A. Skowron, and J.J. Pietrzyk. Automatic planning of treatment of infants with respiratory failure through rough set modeling. In: S. Greco, Y. Hato, S. Hirano, M. Inuiguchi, S. Miyamoto, H.S. Nguyen, and R. Slowiński (eds), *Proceedings of the 5th International Conference on Rough Sets and Current Trends in Computing (RSCTC 2006)*, Kobe, Japan, November 6–8, 2006, LNAI 4259, Springer, Heidelberg, 2006, pp. 418–427.
- [38] R. Sun. *Cognition and Multi-Agent Interaction From Cognitive Modeling to Social Simulation*. Cambridge University Press, New York, 2006.
- [39] R. Sun. *Duality of the Mind: A Bottom-up Approach Toward Cognition*. Lawrence Erlbaum Associates, Mahwah, NJ, 2000.
- [40] Z. Pawlak and A. Skowron. Rudiments of rough sets. *Inf. Sci.* 177(1) (2007) 3–27.
- [41] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In: P. Buneman and S. Jajodia (eds), *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Washington, DC, May 26–28, 1993, ACM Press, New York, 1993, pp. 207–216.
- [42] J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, Heidelberg, 2001.

- [43] Z. Pawlak and A. Skowron. Rough sets: some extensions. *Inf. Sci.* 177(1) (2007) 28–40.
- [44] Z. Pawlak and A. Skowron. Rough sets and boolean reasoning. *Inf. Sci.* 177(1) (2007) 41–73.
- [45] J.Y. Halpern, R. Fagin, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge, MA, 1995.
- [46] M. Mitchell. A Complex-systems perspective on the “Computation vs. Dynamics” debate in cognitive science. In: M.A. Gernsbacher and S.J. Derry (eds), *Proceedings of the 20th Annual Conference of the Cognitive Science Society (COGSCI 1998)*, University of Wisconsin-Madison, Madison, August 1–4, 1998, pp. 710–715.
- [47] K. Sycara. Multiagent systems. *AI Mag.* 19(2) (1998) 79–92.
- [48] J. Liu. *Autonomous Agents and Multi-Agent Systems: Explorations in Learning, Self-Organization and Adaptive Computation*. World Scientific Publishing, Singapore, 2001.
- [49] J. Liu, X. Jin, and K.C. Tsui. *Autonomy Oriented Computing: From Problem Solving to Complex Systems Modeling*. Kluwer/Springer, Heidelberg, 2005.
- [50] M. Luck, P. McBurney, and C. Preist. *Agent Technology. Enabling Next Generation Computing: A Roadmap for Agent Based Computing*, AgentLink, 2003.
- [51] B. Dunin-Keplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds). *Monitoring, Security, and Rescue Tasks in Multiagent Systems (MSRAS'2004), Advances in Soft Computing*. Springer, Heidelberg, 2005.
- [52] A. Desai. Adaptive complex enterprises. *Commun. ACM* 48 (2005) 32–35.
- [53] L.A. Zadeh. Fuzzy sets. *Inf. Control* 8 (1965) 338–353.
- [54] G. Frege. Grundgesetzen der Arithmetik. Verlag von Hermann Pohle, Jena, Vol. 2, 1903.
- [55] L. Polkowski and A. Skowron. Rough mereology: a new paradigm for approximate reasoning. *Int. J. Approx. Reason.* 15(4) (1996) 333–365.
- [56] S. Leśniewski. Grungzüge eines neuen Systems der Grundlagen der Mathematik. *Fundam. Math.* 14 (1929) 1–81.
- [57] J. Łukasiewicz. Die logischen Grundlagen der Wahrscheinlichkeitsrechnung, Kraków 1913. In: L. Borkowski (ed.), *Jan Łukasiewicz – Selected Works*. North-Holland/Polish Scientific Publishers, Amsterdam, London, Warsaw, 1970, pp. 16–63.
- [58] F. Brown. *Boolean Reasoning*. Kluwer Academic Publishers, Dordrecht, 1990.
- [59] A. Skowron. Rough sets in KDD (plenary talk). In: Z. Shi, B. Faltings, and M. Musen (eds), *16-th World Computer Congress (IFIP'2000): Proceedings of Conference on Intelligent Information Processing (IIP'2000)*, Publishing House of Electronic Industry, Beijing, 2000, pp. 1–14.
- [60] H.S. Nguyen. Approximate boolean reasoning: foundations and applications in data mining. In: *Transactions on Rough Sets V: Journal Subline*. LNCS 4100, Springer, Heidelberg, 2006, pp. 344–523.
- [61] Rough Set Exploration System (RSES), logic.mimuw.edu.pl/~rses, accessed January 22, 2008.
- [62] L. Breiman. Statistical modeling: the two cultures. *Stat. Sci.* 16(3) (2001) 199–231.
- [63] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
- [64] R. Miikkulainen, J.A. Bednar, Y. Choe, and J. Sirosh. *Computational Maps in the Visual Cortex*. Springer, Heidelberg, 2005.
- [65] K.D. Forbus and T.R. Hinrichs. Engines of the brain: the computational instruction set of human cognition. *AI Mag.* 27 (2006) 15–31.
- [66] M. Fahle and T. Poggio. *Perceptual Learning*. MIT Press, Cambridge, MA, 2002.
- [67] T. Poggio and S. Smale. The mathematics of learning: dealing with data. *Not. AMS* 50(5) (2003) 537–544.
- [68] M. Gell-Mann. *The Quark and the Jaguar – Adventures in the Simple and the Complex*. Brown and Co., London, 1994.
- [69] P. Stone. *Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer*. MIT Press, Cambridge, MA, 2000.
- [70] S. Behnke. *Hierarchical Neural Networks for Image Interpretation*, LNCS 2766. Springer, Heidelberg, 2003.
- [71] J. Bazan, P. Kruczek, S. Bazan-Socha, A. Skowron, and J.J. Pietrzyk. Risk pattern identification in the treatment of infants with respiratory failure through rough set modeling. In: *Proceedings of IPMU'2006*, Paris, France, July 2–7, 2006, Éditions E.D.K., Paris, 2006, pp. 2650–2657.
- [72] S.H. Nguyen, T.T. Nguyen, and H.S. Nguyen. Rough set approach to sunspot classification. In: D. Ślezak, J.T. Yao, J.F. Peters, W. Ziarko, and X. Hu (eds), *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2005)*, Regina, Canada, August 31–September 3, 2005, Part II, LNAI 3642. Springer, Heidelberg, 2005, pp. 263–272.
- [73] C. Urmon, J. Anhalt, M. Clark, T. Galatali, J.P. Gonzalez, J. Gowdy, A. Gutierrez, S. Harbaugh, M. Johnson-Roberson, H. Kato, P.L. Koon, K. Peterson, B.K. Smith, S. Spiker, E. Tryzelaar, and W.R.L. Whittaker. *High Speed Navigation of Unrehearsed Terrain: Red Team Technology for Grand Challenge 2004*. Technical Report CMU-RI-TR-04-37. Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, June 2004.
- [74] J. Bazan. The road simulator, logic.mimuw.edu.pl/~bazan/simulator, accessed January 22, 2008.

- [75] R. Duda, P. Hart, and R. Stork. *Pattern Classification*. Wiley, New York, 2002.
- [76] T.T. Nguyen and A. Skowron. Rough set approach to domain knowledge approximation. In: G. Wang, Q. Liu, Y. Yao, and A. Skowron (eds), *Proceedings of the 9-th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2003)*, Chongqing, China, October 19–22, 2003, LNCS 2639, Springer, Heidelberg, 2003, pp. 221–228.
- [77] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
- [78] T.G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Artif. Intell.* 13(5) (2000) 227–303.
- [79] A. McGovern. *Autonomous Discovery of Temporal Abstractions from Interaction with an Environment*. Ph.D. Thesis. University of Massachusetts, Amherst, 2002.
- [80] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: a survey. *J. Artif. Intell. Res.* 4 (1996) 227–303.
- [81] J.F. Peters. Approximation spaces for hierarchical intelligent behavioural system models. In: B.D.-Kęplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds), *Monitoring, Security and Rescue Techniques in Multiagent Systems, Advances in Soft Computing*. Physica-Verlag, Heidelberg, 2004, pp. 13–30.
- [82] J.F. Peters. Rough ethology: towards a biologically-inspired study of collective behaviour in intelligent systems with approximation spaces. In: *Transactions on Rough Sets III: LNCS Journal Subline*. LNCS 3400, Springer, Heidelberg, 2005, PP. 153–174.
- [83] J.F. Peters and C. Henry. Reinforcement learning with approximation spaces. *Fundam. Inf.* 71(2–3) (2006) 323–349.
- [84] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Elsevier, Morgan Kaufmann, CA, 2004.
- [85] W. Van Wezel, R. Jorna, and A. Meystel. *Planning in Intelligent Systems: Aspects, Motivations, and Methods*. Wiley, Hoboken, NJ, 2006.
- [86] A. Skowron. Perception logic in intelligent systems. In: S. Blair, U. Chakraborty, S.-H. Chen, et al. (eds), *Proceedings of the 8th Joint Conference on Information Sciences (JCIS 2005)*, July 21–26, 2005, Salt Lake City, UT, X-CD Technologies: A Conference & Management Company, ISBN 0-9707890-3-3, Toronto, Ontario, Canada, 2005, pp. 1–5.
- [87] R.M. Axelrod. *The Complexity of Cooperation*. Princeton University Press, Princeton, NJ, 1997.
- [88] S. Kraus. *Strategic Negotiations in Multiagent Environments*. MIT Press, MA, 2001.
- [89] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence. From Natural to Artificial Systems*. Oxford University Press, Oxford, 1999.
- [90] Z. Pawlak. Concurrent versus sequential the rough sets perspective. *Bull. EATCS* 48 (1992) 178–190.
- [91] A. Skowron and Z. Suraj. Rough sets and concurrency. *Bull. Pol. Acad. Sci.* 41(3) (1993) 237–254.
- [92] Z. Suraj. Rough set methods for the synthesis and analysis of concurrent processes. In: L. Polkowski, S. Tsumoto, and T.Y. Lin (eds), *Rough Set Methods and Applications, Studies in Fuzziness and Soft Computing 56*. Physica-Verlag, Heidelberg, 2000, pp. 379–488.
- [93] K.P. Unnikrishnan, N. Ramakrishnan, P.S. Sastry, and R. Uthurusamy. *4th KDD Workshop on Temporal Data Mining: Network Reconstruction from Dynamic Data Aug 20, 2006, The Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data (KDD 2006)*, August 20–23, 2006, Philadelphia, <http://people.cs.vt.edu/ramakris/kddtm06/cfp.html>.
- [94] W. Bridewell, P. Langley, S. Racunas, and S. Borrett, Learning process models with missing data. In: J. Fürnkranz, T. Scheffer, and M. Spiliopoulou (eds), *Proceedings of the Seventeenth European Conference on Machine Learning (ECML 2006)*, Berlin, Germany, September 18–22, 2006, LNCS 4212, Springer, Berlin, 2006, pp. 557–565.
- [95] P. Langley, J.N. Sánchez, L. Todorovski, and S. Dzeroski. Inducing process models from continuous data. In: C. Sammut and A.G. Hoffmann (eds), *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002)*, University of New South Wales, Sydney, Australia, July 8–12, Morgan Kaufmann, San Francisco, 2002, pp. 347–354.
- [96] A. Skowron and P. Synak. Complex patterns. *Fundam. Inf.* 60(1–4) (2004) 351–366.
- [97] M. Mitchell. Complex systems: network thinking. *Artif. Intell.* 170(18) (2006) 1194–1212.
- [98] M. Mitchell and M. Newman. Complex systems theory and evolution. In: M. Pagel (ed), *Encyclopedia of Evolution*. Oxford University Press, New York, 2002.
- [99] L.A. Segel and I.R. Cohen (eds). *Design Principles for the Immune System and Other Distributed Autonomous Systems*. Oxford University Press, New York, 2001.

15

Granular Computing for Reasoning about Ordered Data: The Dominance-Based Rough Set Approach

Salvatore Greco, Benedetto Matarazzo, and Roman Slowiński

15.1 Introduction

Rough set theory [1, 2] relies on the idea that some knowledge (data, information) is available about objects of a universe of discourse U . Thus, a subset of U is defined using the available knowledge about the objects and not on the base of information about membership or non-membership of the objects to the subset. For example, knowledge about patients suffering from a certain disease may contain information about body temperature, blood pressure, etc. All patients described by the same information are *indiscernible* in view of the available knowledge and form groups of similar objects. These groups are called *elementary sets* and can be considered as elementary granules of the available knowledge about patients. Elementary sets can be combined into compound concepts. For example, elementary sets of patients can be used to represent a set of patients suffering from a certain disease. Any union of elementary sets is called *crisp set*, while other sets are referred to as *rough sets*. Each rough set has boundary-line objects, i.e., objects which, in view of the available knowledge, cannot be classified with certainty as members of the set or of its complement. Therefore, in the rough set approach, any set is associated with a pair of crisp sets, called the lower and the upper approximation. Intuitively, in view of the available information, the *lower approximation* consists of all objects which certainly belong to the set, and the *upper approximation* contains all objects which possibly belong to the set. The difference between the upper and the lower approximation constitutes the *boundary* region of the rough set. Analogously, for a partition of universe U into classes, one may consider rough approximation of the partition. It appeared to be particularly useful for analysis of classification problems, being the most common decision problem.

The rough set approach operates on an *information table* composed of a set U of objects described by a set Q of attributes. If in the set Q disjoint sets (C and D) of *condition* and *decision* attributes are distinguished, then the information table is called *decision table*. It is often assumed, without loss of generality, that set D is a singleton $\{d\}$, and thus decision attribute d makes a partition of set U into decision classes. Data collected in such a decision table correspond to a multiple-attribute classification problem.

The classical indiscernibility-based rough set approach (IRSA) is naturally adapted to analysis of this type of decision problem, because the set of objects can be identified with examples of classification and it is possible to extract all the essential knowledge contained in the decision table using indiscernibility or similarity relations. However, as pointed out by the authors (see e.g. [3–6]), IRSA cannot extract all the essential knowledge contained in the decision table if a background knowledge about *monotonic relationships* between evaluation of objects on condition attributes and their assignment to decision classes has to be taken into account. Such a background knowledge is typical for data describing various phenomena, as well as for data describing multiple-criteria decision problems (see e.g. [7]); e.g., ‘the larger the mass and the smaller the distance, the larger the gravity’, ‘the more a tomato is red, the more it is ripe’, or ‘the better the school marks of a pupil, the better his overall classification’. The monotonic relationships typical for multiple-criteria decision problems follow from preferential ordering of value sets of attributes (scales of criteria), as well as preferential ordering of decision classes.

In order to take into account the *ordinal* properties of the considered attributes and the monotonic relationships between conditions and decisions, a number of methodological changes to the original rough set theory were necessary. The main change was the replacement of the indiscernibility relation with a *dominance relation*, which permits approximation of ordered sets. The dominance relation is a very natural and rational concept within multiple-criteria decision analysis. The dominance-based rough set approach (DRSA) has been proposed and characterized by the authors (see e.g. [3–6, 8, 9]).

Looking at DRSA from granular computing perspective, we can say that DRSA permits to deal with *ordered data* by considering a specific type of information granules defined by means of *dominance-based constraints* having a syntax of the type: ‘ x is at least R ’ or ‘ x is at most R ’, where R is a qualifier from a properly ordered scale. In evaluation space, such granules are *dominance cones*. In this sense, the contribution of DRSA consists in

- extending the paradigm of granular computing to problems involving ordered data,
- specifying a proper syntax and modality of information granules (the dominance-based constraints which should be adjoined to other modalities of information constraints, such as possibilistic, veristic, and probabilistic [10]), and
- defining a methodology dealing properly with this type of information granules and resulting in a theory of computing with words and reasoning about ordered data.

Let us observe that other modalities of information constraints, such as veristic, possibilistic, and probabilistic, have also to deal with ordered values (with qualifiers relative to grades of truth, possibility, and probability). We believe, therefore, that granular computing with ordered data and DRSA as a proper way of reasoning about ordered data are very important in the future development of the whole domain of granular computing.

In the late 1990s, adapting IRSA to the analysis of preference-ordered data became a particularly challenging problem within the field of multiple-criteria decision support. Why might it be so important? The answer is related to the nature of the input preference information available in multiple-criteria decision analysis and of the output of this analysis. As to the input, the rough set approach requires a set of *decision examples*. Such representation is convenient for the acquisition of preference information from decision makers. Very often, in multiple-criteria decision analysis, this information has to be given in terms of preference model parameters, such as importance weights, substitution rates, and various thresholds. Producing such information requires a significant cognitive effort on the part of the decision maker. It is generally acknowledged that people often prefer to make exemplary decisions and cannot always explain them in terms of specific parameters. For this reason, the idea of inferring preference models from exemplary decisions provided by the decision maker is very attractive. Furthermore, the exemplary decisions may be inconsistent because of limited clear discrimination between values of particular criteria and because of hesitation on the part of the decision maker. These *inconsistencies* cannot be considered as a simple error or as noise. They can convey important information that should be taken into account in the construction of the decision maker’s preference model. The rough set approach is intended to deal with inconsistency and this is a major argument to support its application to multiple-criteria decision analysis. The output of the analysis, i.e., the model of preferences in terms of ‘if . . . , then . . . ’ *decision rules*, is very

convenient for decision support, because it is intelligible and speaks the same language as the decision maker. The separation of certain and uncertain knowledge about the decision maker's preferences results from the distinction of different kinds of decision rules, induced from lower approximations of decision classes or from the boundaries, i.e., the difference between upper and lower approximations which is composed of inconsistent examples. Such a preference model is more general than the traditional functional models considered within multiattribute utility theory or the relational models considered, for example, in outranking methods. This conclusion has been acknowledged by a thorough study of axiomatic foundations of the preference models [11–13]. Let us mention that DRSA has also been used as a tool for inducing parameters of other preference models than the decision rules, like the relational outranking models used in multiple-criteria choice problems [14].

DRSA can be applied straightforward to multiple-criteria *classification* (called also *sorting* or *ordinal classification*) problems. The decision table contains in this case the preference information in the form of a finite set of classification examples provided by the decision maker. Note that, while multiple-criteria classification is based on absolute evaluation of objects, multiple-criteria *choice* and *ranking* refer to pairwise comparisons of objects. These pairwise comparisons are in this case the preference information provided by the decision maker. Thus, the decision table is replaced with the pairwise comparison table (PCT) and the decision rules to be discovered from this table characterize a comprehensive preference relation on the set of objects. In consequence, the preference model of the decision maker is a *set of decision rules*. It may be used to *explain* the decision policy of the decision maker and to *recommend* a good choice or preference ranking with respect to new objects.

In the context of all three kinds of multiple-criteria decision problems – classification, choice, and ranking – one can consider imprecise preference information using fuzzy sets. For this purpose, in [3, 15], we proposed *fuzzy set extensions* of DRSA based on fuzzy connectives.

Even if DRSA has been proposed to deal with ordinal properties of data related to preferences in decision problems, the concept of dominance-based rough approximation can be used in a much more general context [16]. This is because the *monotonicity*, which is crucial for DRSA, is also meaningful for problems where preferences are not considered. Monotonicity concerns mutual trends between different variables like distance and gravity in physics or inflation rate and interest rate in economics. Whenever we discover a relationship between different aspects of a phenomenon, this relationship can be represented by a monotonicity with respect to some specific measures of these aspects. So, in general, monotonicity is a property translating in a formal language a primitive intuition of relationship between different concepts of our knowledge. In this perspective, DRSA gives a very general framework for reasoning about data related by monotonicity relationships.

In IRSAs, the idea of monotonicity is not evident, although it is also present there. Because of very coarse representation of considered concepts, monotonicity is taken into account in the sense of presence or absence of particular aspects characterizing the concepts. This is why IRSAs can be considered as a particular case of DRSA.

Monotonicity gains importance when the binary scale, including only 'presence' and 'absence' of an aspect, becomes finer and permits to consider the presence of a property to a certain degree. Due to *graduality*, the idea of monotonicity can be exploited in the whole range of its potential. Graduality is typical for fuzzy set philosophy [17], and thus a joint consideration of rough sets and fuzzy sets is worthwhile. In fact, rough sets and fuzzy sets capture the two basic complementary aspects of monotonicity: rough sets deal with relationships between different concepts, and fuzzy sets deal with expression of different dimensions in which the concepts are considered. For this reason, many approaches have been proposed to combine fuzzy sets with rough sets (see, e.g., [18–21]).

The main preoccupation in almost all the studies combining rough sets with fuzzy sets was related to a fuzzy extension of Pawlak's definition of lower and upper approximations using *fuzzy connectives* [22, 23]. DRSA can also be combined with fuzzy sets along this line, obtaining a rough set model permitting to deal with fuzziness in preference representation [3, 15, 24]. Let us remark, however, that in fact there is no rule for the choice of the 'right' fuzzy connective, so this choice is always arbitrary to some extent. Moreover, there is another drawback for fuzzy extensions of rough sets involving fuzzy connectives: they are based on cardinal properties of membership degrees. In consequence, the result of these extensions is sensitive to order-preserving transformation of membership degrees.

For example, consider the t -conorm of Łukasiewicz as fuzzy connective; it may be used in the definition of both fuzzy lower approximation (to build fuzzy implication) and fuzzy upper approximation (as a fuzzy counterpart of a union). The t -conorm of Łukasiewicz is defined as

$$T^*(\alpha, \beta) = \min\{\alpha + \beta, 1\}, \quad \alpha, \beta \in [0, 1].$$

$T^*(\alpha, \beta)$ can be interpreted as follows. If $\alpha = \mu_X(z)$ represents the membership of z to set X and $\beta = \mu_Y(z)$ represents the membership of z to set Y , then $T^*(\alpha, \beta)$ expresses the membership of z to set $X \cup Y$. Given two fuzzy propositions p and q , putting $v(p) = \alpha$ and $v(q) = \beta$, $T^*(\alpha, \beta)$ can also be interpreted as $v(p \vee q)$, the truth value of the proposition $p \vee q$. Let us consider the following values of arguments:

$$\alpha = 0.5, \quad \beta = 0.3, \quad \gamma = 0.2, \quad \delta = 0.1,$$

and their order-preserving transformation:

$$\alpha' = 0.4, \quad \beta' = 0.3, \quad \gamma' = 0.2, \quad \delta' = 0.05.$$

The values of the t -conorm in the two cases are as follows:

$$T^*(\alpha, \delta) = 0.6, \quad T^*(\beta, \gamma) = 0.5, \quad T^*(\alpha', \delta') = 0.45, \quad T^*(\beta', \gamma') = 0.5.$$

One can see that the order of the results has changed after the order-preserving transformation of the arguments. This means that the Łukasiewicz t -conorm takes into account not only the ordinal properties of the membership degrees, but also their cardinal properties. A natural question arises: is it reasonable to expect from the membership degree a cardinal content instead of ordinal only? Or, in other words, is it realistic to claim that a human is able to say in a meaningful way not only that

- (a) ‘object x belongs to fuzzy set X more likely than object y ’
(or ‘proposition p is more credible than proposition q ’),

but even something like

- (b) ‘object x belongs to fuzzy set X two times more likely than object y ’
(or ‘proposition p is two times more credible than proposition q ’)?

It is safer, of course, to consider information of type (a), because information of type (b) is rather meaningless for a human (see [25]). The above doubt about the cardinal content of the fuzzy membership degree shows the need for methodologies which consider the imprecision in perception typical for fuzzy sets but avoid as much as possible meaningless transformation of information through fuzzy connectives.

The DRSA proposed in [26, 27] for a fuzzy extension of rough sets takes into account the above request. It avoids arbitrary choice of fuzzy connectives and not meaningful operations on membership degrees. It exploits only ordinal character of the membership degrees and proposes a methodology of fuzzy rough approximation that infers the most cautious conclusion from available imprecise information. In particular, any approximation of knowledge about Y using knowledge about X is based on positive or negative relationships between premises and conclusions; i.e.,

- (i) ‘the more x is X , the more it is Y ’ (positive relationship),
- (ii) ‘the more x is X , the less it is Y ’ (negative relationship).

The following simple relationships illustrate (i) and (ii):

- ‘The larger the market share of a company, the greater its profit’
(positive relationship).
- ‘The greater the debt of a company, the smaller its profit’
(negative relationship).

These relationships have the form of *gradual decision rules*. Examples of these decision rules are:

'if a car is speedy with credibility at least 0.8 and it has high fuel consumption with credibility at most 0.7, then it is a good car with credibility at least 0.9',

and

'if a car is speedy with credibility at most 0.5 and it has high fuel consumption with credibility at least 0.8, then it is a good car with credibility at most 0.6'.

Remark that the syntax of gradual decision rules is based on monotonic relationship between degrees of credibility that can also be found in dominance-based decision rules induced from preference-ordered data. This explains why one can build a fuzzy rough approximation using DRSA.

Finally, the fuzzy rough approximation taking into account monotonic relationships can be applied to case-based reasoning [28]. In this perspective, we propose to consider monotonicity of the type 'the more similar is y to x , the more credible is that y belongs to the same set as x '. Application of DRSA in this context leads to decision rules similar to the gradual decision rules:

'the more object z is similar to a referent object x with respect to condition attribute s , the more z is similar to a referent object x with respect to decision attribute t' ,

or, equivalently, but more technically,

$$s(z, x) \geq \alpha \Rightarrow t(z, x) \geq \alpha,$$

where functions s and t measure the credibility of similarity with respect to condition attribute and decision attribute, respectively. When there are multiple condition and decision attributes, functions s and t aggregate similarity with respect to these attributes.

The decision rules we propose do not need the aggregation of the similarity with respect to different attributes into one comprehensive similarity. This is important because it permits to avoid using aggregation operators (weighted average, min, etc.) which are always arbitrary to some extent. Moreover, the decision rules we propose permit to consider different thresholds for degrees of credibility in the premise and in the conclusion.

The chapter is organized as follows. Section 15.2 recalls the main steps of DRSA for multiple-criteria classification, as well as for general classification problems with monotonic relationships between evaluations of objects on condition attributes and their assignment to decision classes. In Section 15.3, a 'probabilistic' version of DRSA, the variable-consistency DRSA (VC-DRSA), is presented. In Section 15.4, we review fuzzy set extensions of DRSA based on fuzzy connectives. Section 15.5 summarizes DRSA for decision under uncertainty, and Section 15.6 presents DRSA for multiple-criteria choice and ranking in case of crisp or fuzzy preferences. Dominance-based rough approximation of a fuzzy set is presented in Section 15.7. In Section 15.8, monotonic rough approximation of a fuzzy set is related to classical rough set, showing that IRSA is a particular case of DRSA. Section 15.9 is devoted to DRSA for case-based reasoning, and Section 15.10 contains conclusions.

15.2 Dominance-Based Rough Set Approach

In this section we present the main concepts of the DRSA (for a more complete presentation see, for example, [3–6]).

Information about objects is represented in the form of an information table. The rows of the table are labeled by objects, whereas columns are labeled by attributes and entries of the table are attribute values. Formally, by an information table we understand the 4-tuple $S = \langle U, Q, V, \phi \rangle$, where U is a finite set of objects, Q is a finite set of attributes, $V = \bigcup_{q \in Q} V_q$ and V_q is the set of values of the attribute q , and $\phi : U \times Q \rightarrow V_q$ is a total function such that $\phi(x, q) \rightarrow V_q$ for every $q \in Q, x \in U$, called an information function [2]. The set Q is, in general, divided into set C of condition attributes and set D of decision attributes.

We are considering condition attributes with sets of values ordered according to decreasing or increasing preference – such attributes are called *criterias*. For criterion $q \in Q$, \succeq_q is a *weak preference* relation on

U such that $x \succeq_q y$ means ‘ x is at least as good as y with respect to criterion q ’. We suppose that \succeq_q is a complete preorder, i.e., a strongly complete and transitive binary relation, defined on U on the basis of evaluations $\phi(\cdot, q)$. We assume, without loss of generality, that the preference is increasing with the value of $\phi(\cdot, q)$ for every criterion $q \in C$, such that for all $x, y \in U$, $x \succeq_q y$ if and only if $\phi(x, q) \geq \phi(y, q)$.

Furthermore, we assume that the set of decision attributes D is a singleton $\{d\}$. Decision attribute d makes a partition of U into a finite number of decision classes, $Cl = \{Cl_t, t = 1, \dots, n\}$, such that each $x \in U$ belongs to one and only one class $Cl_t \in Cl$. We suppose that the classes are preference ordered; i.e., for all $r, s \in \{1, \dots, n\}$, such that $r > s$, the objects from Cl_r are preferred to the objects from Cl_s . More formally, if \succeq is a *comprehensive weak preference relation* on U , i.e., if for all $x, y \in U$, $x \succeq y$ means ‘ x is at least as good as y ’, we suppose that $[x \in Cl_r, y \in Cl_s, r > s] \Rightarrow [x \succeq y \text{ and not } y \succeq x]$. The above assumptions are typical for consideration of a *multiple-criteria classification problem* (also called multiple-criteria sorting or ordinal classification problem).

The sets to be approximated are called *upward union* and *downward union* of classes, respectively:

$$Cl_t^{\geq} = \bigcup_{s \geq t} Cl_s, \quad Cl_t^{\leq} = \bigcup_{s \leq t} Cl_s, \quad t = 1, \dots, n.$$

The statement $x \in Cl_t^{\geq}$ means ‘ x belongs to at least class Cl_t ’, while $x \in Cl_t^{\leq}$ means ‘ x belongs to at most class Cl_t ’. Let us remark that $Cl_1^{\geq} = Cl_n^{\leq} = U$, $Cl_n^{\geq} = Cl_n$, and $Cl_1^{\leq} = Cl_1$. Furthermore, for $t = 2, \dots, n$, we have

$$Cl_{t-1}^{\leq} = U - Cl_t^{\geq} \quad \text{and} \quad Cl_t^{\geq} = U - Cl_{t-1}^{\leq}.$$

The key idea of the rough set approach is representation (approximation) of knowledge generated by decision attributes, by *granules of knowledge* generated by condition attributes.

In DRSA, where condition attributes are criteria and decision classes are preference ordered, the represented knowledge is a collection of *upward* and *downward unions of classes* and the ‘*granules of knowledge*’ are sets of objects defined using a dominance relation.

We say that x *dominates* y with respect to $P \subseteq C$ (shortly, $x P$ -*dominates* y), denoted by $x D_P y$, if for every criterion $q \in P$, $\phi(x, q) \geq \phi(y, q)$. The relation of P -dominance is reflexive and transitive; i.e., it is a partial preorder.

Given a set of criteria $P \subseteq C$ and $x \in U$, the ‘*granules of knowledge*’ used for approximation in DRSA are

- a set of objects dominating x , called *P-dominating set*,
 $D_P^+(x) = \{y \in U : y D_P x\}$, and
- a set of objects dominated by x , called *P-dominated set*,
 $D_P^-(x) = \{y \in U : x D_P y\}$.

Remark that the ‘*granules of knowledge*’ defined above have the form of upward (positive) and downward (negative) *dominance cones* in the evaluation space.

Let us recall that the *dominance principle* (or Pareto principle) requires that an object x dominating object y on all considered criteria (i.e., x having evaluations at least as good as y on all considered criteria) should also dominate y on the decision (i.e., x should be assigned to at least as good decision class as y). This principle is the only objective principle that is widely agreed on in the multiple-criteria comparisons of objects.

Given $P \subseteq C$, the inclusion of an object $x \in U$ to the upward union of classes Cl_t^{\geq} , $t = 2, \dots, n$, is *inconsistent with the dominance principle* if one of the following conditions holds:

- x belongs to class Cl_t or better but it is P -dominated by an object y belonging to a class worse than Cl_t ; i.e., $x \in Cl_t^{\geq}$, but $D_P^+(x) \cap Cl_{t-1}^{\leq} \neq \emptyset$.
- x belongs to a worse class than Cl_t but it P -dominates an object y belonging to class Cl_t or better; i.e., $x \notin Cl_t^{\geq}$, but $D_P^-(x) \cap Cl_t^{\geq} \neq \emptyset$.

If, given a set of criteria $P \subseteq C$, the inclusion of $x \in U$ to Cl_t^{\geq} is inconsistent with the dominance principle, we say that x belongs to Cl_t^{\geq} with some ambiguity, where $t = 2, \dots, n$. Thus, x belongs to

Cl_t^{\geq} without any ambiguity with respect to $P \subseteq C$ if $x \in Cl_t^{\geq}$ and there is no inconsistency with the dominance principle. This means that all objects P -dominating x belong to Cl_t^{\geq} ; i.e., $D_P^+(x) \subseteq Cl_t^{\geq}$.

Furthermore, x possibly belongs to Cl_t^{\geq} with respect to $P \subseteq C$ if one of the following conditions holds:

- According to decision attribute d , x belongs to Cl_t^{\geq} .
- According to decision attribute d , x does not belong to Cl_t^{\geq} , but it is inconsistent in the sense of the dominance principle with an object y belonging to Cl_t^{\geq} .

In terms of ambiguity, x possibly belongs to Cl_t^{\geq} with respect to $P \subseteq C$ if x belongs to Cl_t^{\geq} with or without any ambiguity. Due to the reflexivity of the dominance relation D_P , the above conditions can be summarized as follows: x possibly belongs to class Cl_t or better, with respect to $P \subseteq C$, if among the objects P -dominated by x there is an object y belonging to class Cl_t or better; i.e., $D_P^-(x) \cap Cl_t^{\geq} \neq \emptyset$.

The P -lower approximation of Cl_t^{\geq} , denoted by $\underline{P}(Cl_t^{\geq})$, and the P -upper approximation of Cl_t^{\geq} , denoted by $\overline{P}(Cl_t^{\geq})$, are defined as follows ($t = 1, \dots, n$):

$$\begin{aligned}\underline{P}(Cl_t^{\geq}) &= \{x \in U : D_P^+(x) \subseteq Cl_t^{\geq}\}, \\ \overline{P}(Cl_t^{\geq}) &= \{x \in U : D_P^-(x) \cap Cl_t^{\geq} \neq \emptyset\}.\end{aligned}$$

Analogously, one can define the P -lower approximation and the P -upper approximation of Cl_t^{\leq} as follows ($t = 1, \dots, n$):

$$\begin{aligned}\underline{P}(Cl_t^{\leq}) &= \{x \in U : D_P^-(x) \subseteq Cl_t^{\leq}\}, \\ \overline{P}(Cl_t^{\leq}) &= \{x \in U : D_P^+(x) \cap Cl_t^{\leq} \neq \emptyset\}.\end{aligned}$$

The P -lower and P -upper approximations so defined satisfy the following *inclusion properties* for each $t \in \{1, \dots, n\}$ and for all $P \subseteq C$:

$$\underline{P}(Cl_t^{\geq}) \subseteq Cl_t^{\geq} \subseteq \overline{P}(Cl_t^{\geq}), \quad \underline{P}(Cl_t^{\leq}) \subseteq Cl_t^{\leq} \subseteq \overline{P}(Cl_t^{\leq}).$$

The P -lower and P -upper approximations of Cl_t^{\geq} and Cl_t^{\leq} have an important *complementarity property*, according to which,

$$\begin{aligned}\underline{P}(Cl_t^{\geq}) &= U - \overline{P}(Cl_{t-1}^{\leq}) \quad \text{and} \quad \overline{P}(Cl_t^{\geq}) = U - \underline{P}(Cl_{t-1}^{\leq}), \quad t = 2, \dots, n; \\ \underline{P}(Cl_t^{\leq}) &= U - \overline{P}(Cl_{t+1}^{\geq}) \quad \text{and} \quad \overline{P}(Cl_t^{\leq}) = U - \underline{P}(Cl_{t+1}^{\geq}), \quad t = 1, \dots, n-1.\end{aligned}$$

The P -boundary of Cl_t^{\geq} and Cl_t^{\leq} , denoted by $Bn_P(Cl_t^{\geq})$ and $Bn_P(Cl_t^{\leq})$, respectively, are defined as follows ($t = 1, \dots, n$):

$$Bn_P(Cl_t^{\geq}) = \overline{P}(Cl_t^{\geq}) - \underline{P}(Cl_t^{\geq}), \quad Bn_P(Cl_t^{\leq}) = \overline{P}(Cl_t^{\leq}) - \underline{P}(Cl_t^{\leq}).$$

Due to complementarity property, $Bn_P(Cl_t^{\geq}) = Bn_P(Cl_{t-1}^{\leq})$, for $t = 2, \dots, n$.

The dominance-based rough approximations of upward and downward unions of classes can serve to induce ‘if ..., then ...’ decision rules. It is meaningful to consider the following five types of decision rules:

1. Certain D_{\geq} -decision rules:

‘If $x_{q1} \succeq_{q1} r_{q1}$ and $x_{q2} \succeq_{q2} r_{q2}$ and ... $x_{qp} \succeq_{qp} r_{qp}$, then $x \in Cl_t^{\geq}$ ’,

where, for each $w_q, z_q \in X_q$, ‘ $w_q \succeq_q z_q$ ’ means ‘ w_q is at least as good as z_q ’.

2. Possible D_{\geq} -decision rules:

‘If $x_{q1} \succeq_{q1} r_{q1}$ and $x_{q2} \succeq_{q2} r_{q2}$ and ... $x_{qp} \succeq_{qp} r_{qp}$, then x possibly belongs to Cl_t^{\geq} ’.

3. Certain D_{\leq} -decision rules:

‘If $x_{q1} \preceq_{q1} r_{q1}$ and $x_{q2} \preceq_{q2} r_{q2}$ and ... $x_{qp} \preceq_{qp} r_{qp}$, then $x \in Cl_t^{\leq}$ ’,

where, for each $w_q, z_q \in X_q$, ‘ $w_q \preceq_q z_q$ ’ means ‘ w_q is at most as good as z_q ’.

4. Possible D_{\leq} -decision rules:

'If $x_{q1} \preceq_{q1} r_{q1}$ and $x_{q2} \preceq_{q2} r_{q2}$ and ... $x_{qp} \preceq_{qp} r_{qp}$, then x possibly belongs to Cl_t^{\leq} '.

5. Approximate $D_{\geq\leq}$ -decision rules:

'If $x_{q1} \succeq_{q1} r_{q1}$ and ... $x_{qk} \succeq_{qk} r_{qk}$, and $x_{q(k+1)} \preceq_{q(k+1)} r_{q(k+1)}$ and ... $x_{qp} \preceq_{qp} r_{qp}$, then $x \in Cl_s^{\geq} \cap Cl_t^{\leq}$ ',

where $s < t$.

The rules of type (1) and (3) represent certain knowledge induced from the information table, while the rules of type (2) and (4) represent possible knowledge. Rules of type (5) represent doubtful knowledge.

15.3 Variable-Consistency Dominance-Based Rough Set Approach

The definitions of rough approximations introduced in Section 15.2 are based on a strict application of the dominance principle. However, when defining non-ambiguous objects, it is reasonable to accept a limited proportion of negative examples, particularly for large information tables. Such extended version of DRSA is called VC-DRSA [29].

For any $P \subseteq C$, we say that $x \in U$ belongs to Cl_t^{\geq} without any ambiguity at consistency level $l \in (0, 1]$ if $x \in Cl_t^{\geq}$ and at least $l \times 100\%$ of all objects $y \in U$ dominating x with respect to P also belong to Cl_t^{\geq} ; i.e.,

$$\frac{|D_P^+(x) \cap Cl_t^{\geq}|}{|D_P^+(x)|} \geq l.$$

The level l is called *consistency level* because it controls the degree of consistency with respect to objects qualified as belonging to Cl_t^{\geq} without any ambiguity. In other words, if $l < 1$, then at most $(1 - l) \times 100\%$ of all objects $y \in U$ dominating x with respect to P do not belong to Cl_t^{\geq} and thus contradict the inclusion of x in Cl_t^{\geq} .

Analogously, for any $P \subseteq C$, we say that $x \in U$ belongs to Cl_t^{\leq} without any ambiguity at consistency level $l \in (0, 1]$ if $x \in Cl_t^{\leq}$ and at least $l \times 100\%$ of all the objects $y \in U$ dominated by x with respect to P also belong to Cl_t^{\leq} ; i.e.,

$$\frac{|D_P^-(x) \cap Cl_t^{\leq}|}{|D_P^-(x)|} \geq l.$$

The concept of non-ambiguous objects at some consistency level l leads naturally to the corresponding definition of P -lower approximations of the unions of classes Cl_t^{\geq} and Cl_t^{\leq} , respectively:

$$P^l(Cl_t^{\geq}) = \left\{ x \in Cl_t^{\geq} : \frac{|D_P^+(x) \cap Cl_t^{\geq}|}{|D_P^+(x)|} \geq l \right\}, t = 2, \dots, n;$$

$$\underline{P}^l(Cl_t^{\leq}) = \left\{ x \in Cl_t^{\leq} : \frac{|D_P^-(x) \cap Cl_t^{\leq}|}{|D_P^-(x)|} \geq l \right\}, t = 1, \dots, n-1.$$

Given $P \subseteq C$ and consistency level l , we can define the corresponding P -upper approximations of Cl_t^{\geq} and Cl_t^{\leq} , denoted by $\overline{P}^l(Cl_t^{\geq})$ and $\overline{P}^l(Cl_t^{\leq})$, respectively, by complementation of $P^l(Cl_{t-1}^{\leq})$ and $\underline{P}^l(Cl_{t+1}^{\geq})$ with respect to U :

$$\overline{P}^l(Cl_t^{\geq}) = U - \underline{P}^l(Cl_{t-1}^{\leq}), \quad \overline{P}^l(Cl_t^{\leq}) = U - \underline{P}^l(Cl_{t+1}^{\geq}).$$

$\overline{P}^l(Cl_t^{\geq})$ can be interpreted as a set of all the objects belonging to Cl_t^{\geq} , possibly ambiguous at consistency level l . Analogously, $\overline{P}^l(Cl_t^{\leq})$ can be interpreted as a set of all the objects belonging to Cl_t^{\leq} , possibly

ambiguous at consistency level l . The P -*boundaries* (P -doubtful regions) of Cl_t^{\geq} and Cl_t^{\leq} at consistency level l are defined as

$$\begin{aligned} Bn_P^l(Cl_t^{\geq}) &= \overline{P}'(Cl_t^{\geq}) - \underline{P}'(Cl_t^{\geq}), \quad t = 2, \dots, n; \\ Bn_P^l(Cl_t^{\leq}) &= \overline{P}'(Cl_t^{\leq}) - \underline{P}'(Cl_t^{\leq}), \quad t = 1, \dots, n-1. \end{aligned}$$

The *variable consistency* model of the DRSA provides some degree of flexibility in assigning objects to lower and upper approximations of the unions of decision classes. The following property can be easily proved: for $0 < l' < l \leq 1$,

$$\begin{aligned} \underline{P}'(Cl_t^{\geq}) &\subseteq \underline{P}''(Cl_t^{\geq}) \quad \text{and} \quad \overline{P}'(Cl_t^{\geq}) \supseteq \overline{P}'''(Cl_t^{\geq}), \quad t = 2, \dots, n; \\ \underline{P}'(Cl_t^{\leq}) &\subseteq \underline{P}''(Cl_t^{\leq}) \quad \text{and} \quad \overline{P}'(Cl_t^{\leq}) \supseteq \overline{P}'''(Cl_t^{\leq}), \quad t = 1, \dots, n-1. \end{aligned}$$

The following two basic types of variable-consistency decision rules can be considered:

1. D_{\geq} -decision rules with the following syntax:

'If $\phi(x, q1) \geq r_{q1}$ and $\phi(x, q2) \geq r_{q2}$ and ... $\phi(x, qp) \geq r_{qp}$, then $x \in Cl_t^{\geq}$ ',

with confidence α (i.e., in fraction α of considered cases), where

$$P = \{q1, \dots, qp\} \subseteq C, (r_{q1}, \dots, r_{qp}) \in V_{q1} \times V_{q2} \times \dots \times V_{qp} \text{ and } t = 2, \dots, n.$$

2. D_{\leq} -decision rules with the following syntax:

'If $\phi(x, q1) \leq r_{q1}$ and $\phi(x, q2) \leq r_{q2}$ and ... $\phi(x, qp) \leq r_{qp}$, then $x \in Cl_t^{\leq}$ ',

with confidence α , where

$$P = \{q1, \dots, qp\} \subseteq C, (r_{q1}, \dots, r_{qp}) \in V_{q1} \times V_{q2} \times \dots \times V_{qp} \text{ and } t = 1, \dots, n-1.$$

The *variable-consistency* model is inspired by the *variable-precision* model proposed by Ziarko [30, 31] within the classical IRSA.

15.4 Fuzzy Set Extensions of the Dominance-Based Rough Set Approach

The concept of dominance can be refined by introducing gradedness through the use of fuzzy sets. With this aim we recall definitions of fuzzy connectives [22, 23]. For each proposition p , we consider its truth value $v(p)$ ranging from $v(p) = 0$ (p is definitely false) to $v(p) = 1$ (p is definitely true); for all intermediate values, the greater $v(p)$, the more credible is the truth of p . A negation is a non-increasing function $N : [0, 1] \rightarrow [0, 1]$, such that $N(0) = 1$ and $N(1) = 0$. Given proposition p , $N(v(p))$ states the credibility of the negation of p . A t -norm T and a t -conorm T^* are two functions $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$ and $T^* : [0, 1] \times [0, 1] \rightarrow [0, 1]$, such that given two propositions, p and q , $T(v(p), v(q))$ represents the credibility of the conjunction of p and q , and $T^*(v(p), v(q))$ represents the credibility of the disjunction of p and q . t -norm T and t -conorm T^* must satisfy the following properties:

$$\begin{aligned} T(\alpha, \beta) &= T(\beta, \alpha) \quad \text{and} \quad T^*(\alpha, \beta) = T^*(\beta, \alpha), \quad \text{for all } \alpha, \beta \in [0, 1], \\ T(\alpha, \beta) &\leq T(\gamma, \delta) \quad \text{and} \quad T^*(\alpha, \beta) \leq T^*(\gamma, \delta), \quad \text{for all } \alpha, \beta, \gamma, \delta \in [0, 1], \\ &\quad \text{such that } \alpha \leq \gamma \text{ and } \beta \leq \delta, \\ T(\alpha, T(\beta, \gamma)) &= T(T(\alpha, \beta), \gamma) \text{ and } T^*(\alpha, T^*(\beta, \gamma)) = T^*(T^*(\alpha, \beta), \gamma), \\ &\quad \text{for all } \alpha, \beta, \gamma \in [0, 1], \\ T(1, \alpha) &= \alpha \quad \text{and} \quad T^*(0, \alpha) = \alpha, \quad \text{for all } \alpha \in [0, 1]. \end{aligned}$$

A negation is strict iff it is strictly decreasing and continuous. A negation N is involutive iff, for all $\alpha \in [0, 1]$, $N(N(\alpha)) = \alpha$. A strong negation is an involutive strict negation. If N is a strong negation,

then (T, T^*, N) is a De Morgan triplet iff $N(T^*(\alpha, \beta)) = T(N(\alpha), N(\beta))$. A fuzzy implication is a function $I : [0, 1] \times [0, 1] \rightarrow [0, 1]$ such that, given two propositions p and q , $I(v(p), v(q))$ represents the credibility of the implication of q by p . A fuzzy implication must satisfy the following properties (see [22]):

$$\begin{aligned} I(\alpha, \beta) &\geq I(\gamma, \beta), \text{ for all } \alpha, \beta, \gamma \in [0, 1], \text{ such that } \alpha \leq \gamma, \\ I(\alpha, \beta) &\geq I(\alpha, \gamma), \text{ for all } \alpha, \beta, \gamma \in [0, 1], \text{ such that } \beta \geq \gamma, \\ I(0, \alpha) &= 1, \quad I(\alpha, 1) = 1, \text{ for all } \alpha \in [0, 1], \\ I(1, 0) &= 0. \end{aligned}$$

An implication I_{N, T^*}^\rightarrow is a T^* -implication if there is a t -conorm T^* and a strong negation N such that $I_{N, T^*}^\rightarrow(\alpha, \beta) = T^*(N(\alpha), \beta)$. A fuzzy similarity on the universe U is a fuzzy binary relation (i.e., function $R : U \times U \rightarrow [0, 1]$) reflexive ($R(x, x) = 1$ for all $x \in U$), symmetric ($R(x, y) = R(y, x)$ for all $x, y \in U$), and T -transitive (given t -norm T , $T(R(x, y), R(y, z)) \leq R(x, z)$ for all $x, y, z \in U$).

Let \succeq_q be a fuzzy weak preference relation on U with respect to criterion $q \in C$, i.e., $\succeq_q : U \times U \rightarrow [0, 1]$, such that, for all $x, y \in U$, $\succeq_q(x, y)$ represents the credibility of the proposition ‘ x is at least as good as y with respect to criterion q ’. Suppose that \succeq_q is a fuzzy partial T -preorder; i.e., it is reflexive ($\succeq_q(x, x) = 1$ for each $x \in U$) and T -transitive ($T(\succeq_q(x, y), \succeq_q(y, z)) \leq \succeq_q(x, z)$, for each $x, y, z \in U$) (see [22]). Using the fuzzy weak preference relations \succeq_q , $q \in C$, a fuzzy dominance relation on U (denotation $D_P(x, y)$) can be defined for all $P \subseteq C$ as follows:

$$D_P(x, y) = T_{q \in P} (\succeq_q (x, y)).$$

Given $(x, y) \in U \times U$, $D_P(x, y)$ represents the credibility of the proposition ‘ x is at least as good as y with respect to each criterion q from P ’. Since the fuzzy weak preference relations \succeq_q are supposed to be partial T -preorders, also the fuzzy dominance relation D_P is a partial T -preorder. Furthermore, let $Cl = \{Cl_t, t = 1, \dots, n\}$ be a set of fuzzy classes in U , such that for each $x \in U$, $Cl_t(x)$ represents the membership function of x to Cl_t . We suppose, as before, that the classes of Cl are increasingly ordered; i.e., for all $r, s \in \{1, \dots, n\}$ such that $r > s$, the objects from Cl_r have a better comprehensive evaluation than the objects from Cl_s . On the basis of the membership functions of the fuzzy class Cl_t , $t = 1, \dots, n$, we can define fuzzy membership functions of two other sets:

- 1) the upward union fuzzy set Cl_t^\geq , whose membership function $Cl_t^\geq(x)$ represents the credibility of the proposition ‘ x is at least as good as the objects in Cl_t ’:

$$Cl_t^\geq(x) = \begin{cases} 1 & \text{if } \exists s \in \{1, \dots, n\} : Cl_s(x) > 0 \text{ and } s > t \\ Cl_t(x) & \text{otherwise} \end{cases};$$

- 2) the downward union fuzzy set Cl_t^\leq , whose membership function $Cl_t^\leq(x)$ represents the credibility of the proposition ‘ x is at most as good as the objects in Cl_t ’:

$$Cl_t^\leq(x) = \begin{cases} 1 & \text{if } \exists s \in \{1, \dots, n\} : Cl_s(x) > 0 \text{ and } s < t \\ Cl_t(x) & \text{otherwise} \end{cases}.$$

The P -lower and the P -upper approximations of Cl_t^\geq with respect to $P \subseteq C$ are fuzzy sets in U , whose membership functions, denoted by $\underline{P}[Cl_t^\geq(x)]$ and $\overline{P}[Cl_t^\geq(x)]$, are defined as

$$\begin{aligned} \underline{P}[Cl_t^\geq(x)] &= T_{y \in U} (T^*(N(D_P(y, x)), Cl_t^\geq(y))), \\ \overline{P}[Cl_t^\geq(x)] &= T_{y \in U}^* (T(D_P(x, y), Cl_t^\geq(y))). \end{aligned}$$

$\underline{P}[Cl_t^\geq(x)]$ represents the credibility of the proposition ‘for all $y \in U$, y does not dominate x with respect to criteria from P or y belongs to Cl_t^\geq ’, while $\overline{P}[Cl_t^\geq(x)]$ represents the credibility of the proposition ‘there is at least one $y \in U$ dominated by x with respect to criteria from P which belongs to Cl_t^\geq ’.

The P -lower and P -upper approximations of Cl_t^{\leq} with respect to $P \subseteq C$, denoted by $\underline{P}[Cl_t^{\leq}(x)]$ and $\overline{P}[Cl_t^{\leq}(x)]$, can be defined, analogously, as

$$\underline{P}[Cl_t^{\leq}(x)] = T_{y \in U}(T^*(N(D_P(x, y)), Cl_t^{\leq}(y))),$$

$$\overline{P}[Cl_t^{\leq}(x)] = T_{y \in U}^*(T(D_P(y, x), Cl_t^{\leq}(y))).$$

$\underline{P}[Cl_t^{\leq}(x)]$ represents the credibility of the proposition ‘for all $y \in U$, x does not dominate y with respect to criteria from P or y belongs to Cl_t^{\leq} ’, while $\overline{P}[Cl_t^{\leq}(x)]$ represents the credibility of the proposition ‘there is at least one $y \in U$ dominating x with respect to criteria from P which belongs to Cl_t^{\leq} ’.

Let us remark that, using the definition of the T^* -implication, it is possible to rewrite the definition of $\underline{P}[Cl_t^{\geq}(x)]$, $\overline{P}[Cl_t^{\geq}(x)]$, $\underline{P}[Cl_t^{\leq}(x)]$, and $\overline{P}[Cl_t^{\leq}(x)]$ in the following way:

$$\underline{P}[Cl_t^{\geq}(x)] = T_{y \in U}(I_{T^*, N}^*(D_P(y, x), Cl_t^{\geq}(y))),$$

$$\overline{P}[Cl_t^{\geq}(x)] = T_{y \in U}^*(N(I_{T^*, N}^*(D_P(x, y), N(Cl_t^{\geq}(y)))),$$

$$\underline{P}[Cl_t^{\leq}(x)] = T_{y \in U}(I_{T^*, N}^*(D_P(x, y), Cl_t^{\leq}(y))),$$

$$\overline{P}[Cl_t^{\leq}(x)] = T_{y \in U}^*(N(I_{T^*, N}^*(D_P(y, x), N(Cl_t^{\leq}(y))))).$$

The following results can be proved:

(1) For each $x \in U$ and for each $t \in \{1, \dots, n\}$,

$$\underline{P}[Cl_t^{\geq}(x)] \leq Cl_t^{\geq}(x) \leq \overline{P}[Cl_t^{\geq}(x)], \quad \underline{P}[Cl_t^{\leq}(x)] \leq Cl_t^{\leq}(x) \leq \overline{P}[Cl_t^{\leq}(x)].$$

(2) If (T, T^*, N) constitute a De Morgan triplet, and if $N[Cl_t^{\geq}(x)] = Cl_{t-1}^{\leq}(x)$ for each $x \in U$ and $t = 2, \dots, n$, then

$$\underline{P}[Cl_t^{\geq}(x)] = N(\overline{P}[Cl_{t-1}^{\leq}(x)]), \quad \overline{P}[Cl_t^{\geq}(x)] = N(\underline{P}[Cl_{t-1}^{\leq}(x)]), \quad t = 2, \dots, n$$

$$\underline{P}[Cl_t^{\leq}(x)] = N(\overline{P}[Cl_{t+1}^{\geq}(x)]), \quad \overline{P}[Cl_t^{\leq}(x)] = N(\underline{P}[Cl_{t+1}^{\geq}(x)]), \quad t = 1, \dots, n-1.$$

(3) For all $P \subseteq R \subseteq C$, for all $x \in U$ and for each $t \in \{1, \dots, n\}$,

$$\underline{P}[Cl_t^{\geq}(x)] \leq \underline{R}[Cl_t^{\geq}(x)], \quad \overline{P}[Cl_t^{\geq}(x)] \geq \overline{R}[Cl_t^{\geq}(x)],$$

$$\underline{P}[Cl_t^{\leq}(x)] \leq \underline{R}[Cl_t^{\leq}(x)], \quad \overline{P}[Cl_t^{\leq}(x)] \geq \overline{R}[Cl_t^{\leq}(x)].$$

Results (1)–(3) can be read as fuzzy counterparts of the following results well known within the classical rough set approach: (1) *inclusion property* says that Cl_t^{\geq} and Cl_t^{\leq} include their P -lower approximations and are included in their P -upper approximations; (2) *complementarity property* says that the P -lower (P -upper) approximation of Cl_t^{\geq} is the complement of the P -upper (P -lower) approximation of its complementary set Cl_{t-1}^{\leq} (analogous property holds for Cl_t^{\leq} and Cl_{t+1}^{\geq}); (3) *monotonicity with respect to sets of attributes* says that enlarging the set of criteria, the membership to the lower approximation does not decrease and the membership to the upper approximation does not increase.

Greco *et al.* [24] proposed, moreover, the following fuzzy rough approximations based on dominance, which go in line with the fuzzy rough approximation by Dubois and Prade [18, 19], concerning classical rough sets:

$$\begin{aligned}\underline{P}[Cl_t^{\geq}(x)] &= \inf_{y \in U} (I(D_P(y, x), Cl_t^{\geq}(y))), \\ \overline{P}[Cl_t^{\geq}(x)] &= \sup_{y \in U} (T(D_P(x, y), Cl_t^{\geq}(y))), \\ \underline{P}[Cl_t^{\leq}(x)] &= \inf_{y \in U} (I(D_P(x, y), Cl_t^{\leq}(y))), \\ \overline{P}[Cl_t^{\leq}(x)] &= \sup_{y \in U} (T(D_P(y, x), Cl_t^{\leq}(y))).\end{aligned}$$

Using fuzzy rough approximations based on DRSA, one can induce decision rules having the same syntax as the decision rules obtained from crisp DRSA. In this case, however, each decision rule has a fuzzy credibility.

15.5 DRSA for Decision under Uncertainty

In [32] we opened a new avenue for applications of the rough set concept to analysis of preference-ordered data. We considered the classical problem of decision under uncertainty extending DRSA by using *stochastic dominance*. In a risky context, an act A stochastically dominates an act B if, for all possible levels k of gain or loss, the probability of obtaining an outcome at least as good as k with A is not smaller than with B. In this context, we have an ambiguity if an act A stochastically dominates an act B, but, nevertheless, B has a comprehensive evaluation better than A. On this basis, it is possible to restate all the concepts of DRSA and adapt this approach to preference analysis under risk and uncertainty. We considered the case of traditional additive probability distribution over the set of future states of the world; however, the model is rich enough to handle non-additive probability distributions and even qualitative ordinal distributions. The rough set approach gives a representation of decision maker's (DM's) preferences under uncertainty in terms of '*if... then...*' decision rules induced from rough approximations of sets of exemplary decisions (preference-ordered classification of acts described in terms of outcomes in uncertain states of the world). This extension is interesting with respect to multiple-criteria decision analysis (MCDA) from two different points of view:

1. Each decision under uncertainty can be viewed as a multicriteria decision, where the criteria are the outcomes in different states of the world.
2. DRSA adapted to decision under uncertainty can be applied to deal with multiple-criteria decision under uncertainty, i.e., a decision problem where in each future state of the world the outcomes are expressed in terms of a set of criteria (see [33, 34]).

15.6 Multiple-Criteria Choice and Ranking Problems

DRSA can also be applied to multiple-criteria choice and ranking problems. However, there is a basic difference between classification problems from one side and choice and ranking from the other side. To give a didactic example, consider a set of companies A for evaluation of a risk of failure, taking into account the debt ratio criterion. To assess the risk of failure of company x , we will not compare the debt ratio of x with the debt ratio of all the other companies from A . The comparison will be made with respect to a fixed risk threshold on the debt ratio criterion. Indeed, the debt ratio of x can be the highest of all companies from A and, nevertheless, x can be classified as a low-risk company if its debt ratio is below the fixed risk threshold. Consider, in turn, the situation, in which we must choose the lowest risk company from A or we want to rank the companies from A from the less risky to the most risky one. In this situation, the comparison of the debt ratio of x with a fixed risk threshold is not useful and, instead, a pairwise comparison of the debt ratio of x with the debt ratio of all other companies from A is relevant for the choice or ranking. Thus, in general, while classification is based on absolute evaluation of

objects (e.g., comparison of the debt ratio to the fixed risk threshold), choice and ranking refer to relative evaluation, by means of pairwise comparisons of objects (e.g., comparisons of the debt ratio of pairs of companies).

The necessity of pairwise comparisons of objects in multiple-criteria choice and ranking problems requires some further extensions of DRSA. Simply speaking, in this context we are interested in the approximation of a binary relation corresponding to a comprehensive preference relation, using other binary relations, corresponding to marginal preference relations on particular criteria, for pairs of objects. In the above example, we would approximate the binary relation of the type ‘from the viewpoint of the risk of failure, company x is comprehensively preferred to company y ’ using binary relations on the debt ratio criterion, like ‘the debt ratio of x is *much better* than that of y ’ or ‘the debt ratio of x is *weakly better* than that of y ’, and so on.

Technically, the modification of DRSA necessary to approach the problems of choice and ranking are twofold:

1. *Pairwise comparison table* (PCT) is considered instead of the simple information table [3]: PCT is a decision table whose rows represent pairs of objects for which multiple-criteria evaluations and a comprehensive preference relation are known.
2. *Dominance principle* is considered for *pairwise comparisons* instead of simple objects: If object x is preferred to y at least as strongly as w is preferred to z on all the considered criteria, then x must be comprehensively preferred to y at least as strongly as w is comprehensively preferred to z .

The application of DRSA to the choice or ranking problems proceeds as follows. First, the DM gives some examples of pairwise comparisons with respect to some reference objects, e.g., a complete ranking from the best to the worst of a limited number of objects – well known to the DM. From this set of examples, a preference model in terms of ‘if ... , then ...’ decision rules is induced. These rules are applied to a larger set of objects. A proper exploitation of the results so obtained gives a final recommendation for the decision problem at hand. Below, we present this methodology more formally and in greater detail.

15.6.1 Pairwise Comparison Table as a Preference Information and a Learning Sample

Let A be the set of objects for the decision problem at hand. Let us also consider a set of reference objects $B \subseteq A$ on which the DM is expressing his/her preferences by pairwise comparisons. Let us represent the comprehensive preference by a function $\text{Pref}: A \times A \rightarrow \mathbf{R}$. In general, for each $x, y \in A$,

- If $\text{Pref}(x, y) > 0$, then $\text{Pref}(x, y)$ can be interpreted as a degree to which x is evaluated better than y .
- If $\text{Pref}(x, y) < 0$, then $\text{Pref}(x, y)$ can be interpreted as a degree to which x is evaluated worse than y .
- If $\text{Pref}(x, y) = 0$, then x is evaluated equivalent to y .

The semantic value of preference $\text{Pref}(x, y)$ can be different. We remember two possible interpretations:

- (a) $\text{Pref}(x, y)$ represents a *degree of outranking* of x over y ; i.e., $\text{Pref}(x, y)$ is the credibility of the proposition ‘ x is at least as good as y ’.
- (b) $\text{Pref}(x, y)$ represents a *degree of net preference* of x over y ; i.e., $\text{Pref}(x, y)$ is the strength with which x is preferred to y .

In case (a), $\text{Pref}(x, y)$ measures the strength of arguments in favor of x and against y , while $\text{Pref}(y, x)$ measures the arguments in favor of y and against x . Thus, there is no relation between values of $\text{Pref}(x, y)$ and $\text{Pref}(y, x)$. In case (b), $\text{Pref}(x, y)$ synthesizes arguments in favor of x and against y together with arguments in favor of y and against x . $\text{Pref}(y, x)$ has a symmetric interpretation and the relation $\text{Pref}(x, y) = -\text{Pref}(y, x)$ is expected.

Let us suppose that objects from set A are evaluated by a consistent family of n criteria $g_i : A \rightarrow \mathbf{R}$, $i = 1, \dots, n$, such that, for each object $x \in A$, $g_i(x)$ represents the evaluation of x with respect to criterion g_i . Using the terms of the rough set approach, the family of criteria constitutes the set C of condition attributes. With respect to each criterion $g_i \in C$, one can consider a particular preference function $Pref_i : \mathbf{R} \times \mathbf{R} \rightarrow \mathbf{R}$, such that for each $x, y \in A$, $Pref_i[g_i(x), g_i(y)]$ for criterion g_i has an interpretation analogous to comprehensive preference relation $Pref(x, y)$; i.e.,

- If $Pref_i[g_i(x), g_i(y)] > 0$, then $Pref_i[g_i(x), g_i(y)]$ is a degree to which x is better than y on criterion g_i .
- If $Pref_i[g_i(x), g_i(y)] < 0$, then $Pref_i[g_i(x), g_i(y)]$ is a degree to which x is worse than y on criterion g_i .
- If $Pref_i[g_i(x), g_i(y)] = 0$, then x is equivalent to y on criterion g_i .

Let us suppose that the DM expresses her preferences with respect to pairs (x, y) from $E \subseteq B \times B$, $|E| = m$. These preferences are represented in an $m \times (n + 1)$ pairwise comparison table S_{PCT} . The m rows correspond to the pairs from E . For each $(x, y) \in E$ in the corresponding row, the first n columns include information about preferences $Pref_i[g_i(x), g_i(y)]$ on particular criteria from set C , while the last, $(n + 1)$ th, column represents the comprehensive preference $Pref(x, y)$.

15.6.2 Dominance in PCT

Given subset $P \subseteq C$ ($P \neq \emptyset$) of criteria and pairs of objects $(x, y), (w, z) \in A \times A$, the pair (x, y) is said to P -dominate the pair (w, z) (denotation $(x, y)D_P(w, z)$) if $Pref_i[g_i(x), g_i(y)] \geq Pref_i[g_i(w), g_i(z)]$ for all $g_i \in P$, i.e., if x is preferred to y at least as strongly as w is preferred to z with respect to each criterion $g_i \in P$. Let us remark that the dominance relation D_P is a partial preorder on $A \times A$; as, in general, it involves different grades of preference on particular criteria, it is called *multigraded* dominance relation.

Given $P \subseteq C$ and $(x, y) \in E$, we define

- a set of pairs of objects P -dominating (x, y) , called P -dominating set,

$$D_P^+(x, y) = \{(w, z) \in E : (w, z)D_P(x, y)\};$$
- a set of pairs of objects P -dominated by (x, y) , called P -dominated set,

$$D_P^-(x, y) = \{(w, z) \in E : (x, y)D_P(w, z)\}.$$

The P -dominating sets and the P -dominated sets defined on E for considered pairs of reference objects from E are ‘granules of knowledge’ that can be used to express P -lower and P -upper approximations of set $Pref^{\geq k} = \{(x, y) \in E : Pref(x, y) \geq k\}$, corresponding to comprehensive preference of degree at least k , and set $Pref^{\leq k} = \{(x, y) \in E : Pref(x, y) \leq k\}$, corresponding to comprehensive preference of degree at most k , respectively:

$$\begin{aligned} \underline{P}(Pref^{\geq k}) &= \{(x, y) \in E : D_P^+(x, y) \subseteq Pref^{\geq k}\}, \\ \overline{P}(Pref^{\geq k}) &= \{(x, y) \in E : D_P^-(x, y) \cap Pref^{\geq k} \neq \emptyset\}, \\ \underline{P}(Pref^{\leq k}) &= \{(x, y) \in E : D_P^-(x, y) \subseteq Pref^{\leq k}\}, \\ \overline{P}(Pref^{\leq k}) &= \{(x, y) \in E : D_P^+(x, y) \cap Pref^{\leq k} \neq \emptyset\}. \end{aligned}$$

The set difference between P -lower and P -upper approximations of sets $Pref^{\geq k}$ and $Pref^{\leq k}$ contains all the ambiguous pairs (x, y) :

$$Bn_P(Pref^{\geq k}) = \overline{P}(Pref^{\geq k}) - \underline{P}(Pref^{\geq k}),$$

$$Bn_P(Pref^{\leq k}) = \overline{P}(Pref^{\leq k}) - \underline{P}(Pref^{\leq k}).$$

The above rough approximations of $Pref^{\geq k}$ and $Pref^{\leq k}$ satisfy properties analogous to the rough approximations of upward and downward unions of classes Cl_t^{\geq} and Cl_t^{\leq} ; precisely, these are as follows:

- Inclusion:

$$\underline{P}(Pref^{\geq k}) \subseteq Pref^{\geq k} \subseteq \overline{P}(Pref^{\geq k}),$$

$$\underline{P}(Pref^{\leq k}) \subseteq Pref^{\leq k} \subseteq \overline{P}(Pref^{\leq k}).$$

- Complementarity:

$$\underline{P}(Pref^{\geq k}) = E - \overline{P}(Pref^{\leq k}), \quad \overline{P}(Pref^{\geq k}) = E - \underline{P}(Pref^{\leq k}),$$

$$\underline{P}(Pref^{\leq k}) = E - \overline{P}(Pref^{\geq k}), \quad \overline{P}(Pref^{\leq k}) = E - \underline{P}(Pref^{\geq k}),$$

where $Pref^{\geq k} = E - Pref^{\leq k}$ and $Pref^{\leq k} = E - Pref^{\geq k}$, and the rough approximation of $Pref^{\geq k}$ and $Pref^{\leq k}$ are analogous to those of $Pref^{\geq k}$ and $Pref^{\leq k}$; e.g.,

$$\underline{P}(Pref^{\geq k}) = \{(x, y) \in E : D_p^+(x, y) \subseteq Pref^{\geq k}\}.$$

- Monotonicity:

For each $R, P \subseteq C$, such that $R \subseteq P$,

$$\underline{R}(Pref^{\geq k}) \subseteq \underline{P}(Pref^{\geq k}), \quad \overline{R}(Pref^{\geq k}) \supseteq \overline{P}(Pref^{\geq k}),$$

$$\underline{R}(Pref^{\leq k}) \subseteq \underline{P}(Pref^{\leq k}), \quad \overline{R}(Pref^{\leq k}) \supseteq \overline{P}(Pref^{\leq k}).$$

The concepts of the quality of approximation, reducts, and core can also be extended to the approximation of the comprehensive preference relation by multigraded dominance relations. In particular, the coefficient

$$\gamma_P = \frac{\left| E - \bigcup_k Bn_P(Pref^{\geq k}) \right|}{|E|} = \frac{\left| E - \bigcup_k Bn_P(Pref^{\leq k}) \right|}{|E|}$$

defines the *quality of approximation of comprehensive preference* $Pref(x, y)$ by criteria from $P \subseteq C$. It expresses the ratio of all pairs $(x, y) \in E$ whose degree of preference of x over y is correctly assessed using set P of criteria to all the pairs of objects contained in E . Each minimal subset $P \subseteq C$, such that $\gamma_P = \gamma_C$, is called *reduct* of C (denoted by $RED_{S_{PCT}}$). Let us remark that S_{PCT} can have more than one reduct. The intersection of all reducts is called the *core* (denoted by $CORE_{S_{PCT}}$).

It is also possible to use the variable-consistency model on S_{PCT} [35, 36], relaxing the definitions of *P-lower* approximations of graded comprehensive preference relations represented by sets $Pref^{\geq k}$ and $Pref^{\leq k}$, such that some pairs in *P-dominated* or *P-dominating* sets belong to the opposite relation but at least $l \times 100\%$ of pairs belong to the correct one. Then, the definition of *P-lower* approximations of $Pref^{\geq k}$ and $Pref^{\leq k}$ at consistency level l with respect to set $P \subseteq C$ of criteria boils down to

$$\underline{P}^l(Pref^{\geq k}) = \left\{ (x, y) \in E : \frac{|D_p^+(x, y) \cap Pref^{\geq k}|}{|D_p^+(x, y)|} \geq l \right\},$$

$$\underline{P}^l(Pref^{\leq k}) = \left\{ (x, y) \in E : \frac{|D_p^-(x, y) \cap Pref^{\leq k}|}{|D_p^-(x, y)|} \geq l \right\}.$$

15.6.3 Induction of Decision Rules from Rough Approximations of Graded Preference Relations

Using the rough approximations of sets $Pref^{\geq k}$ and $Pref^{\leq k}$, i.e., rough approximations of comprehensive preference relation $Pref(x, y)$ of degree at least or at most k , respectively, it is possible to induce a generalized description of the preference information contained in a given S_{PCT} in terms of decision

rules with a special syntax. We are considering decision rules of the following types:

1. D_{\geq} -decision rules:

'If $\text{Pref}_{i1}[g_{i1}(x), g_{i1}(y)] \geq k_{i1}$, and ... $\text{Pref}_{ir}[g_{ir}(x), g_{ir}(y)] \geq k_{ir}$, then $\text{Pref}(x, y) \geq k'$,

where $\{g_{i1}, \dots, g_{ir}\} \subseteq C$; e.g., 'if car x is much better than y with respect to maximum speed and at least weakly better with respect to acceleration, then x is comprehensively better than y '; these rules are supported by pairs of objects from the P -lower approximation of sets $\text{Pref}^{\geq k}$ only.

2. D_{\leq} -decision rules:

'If $\text{Pref}_{i1}[g_{i1}(x), g_{i1}(y)] \leq k_{i1}$, and ... $\text{Pref}_{ir}[g_{ir}(x), g_{ir}(y)] \leq k_{ir}$, then $\text{Pref}(x, y) \leq k'$,

where $\{g_{i1}, \dots, g_{ir}\} \subseteq C$; e.g., 'if car x is much worse than y with respect to price and weakly worse with respect to comfort, then x is comprehensively worse than y '; these rules are supported by pairs of objects from the P -lower approximation of sets $\text{Pref}^{\leq k}$ only.

3. $D_{\geq\leq}$ -decision rules:

'If $\text{Pref}_{i1}[g_{i1}(x), g_{i1}(y)] \geq k_{i1}$, and ... $\text{Pref}_{ir}[g_{ir}(x), g_{ir}(y)] \geq k_{ir}$, and

$\text{Pref}_{j1}[g_{j1}(x), g_{j1}(y)] \leq k_{j1}$, and ... $\text{Pref}_{js}[g_{js}(x), g_{js}(y)] \leq k_{js}$, then $h \leq \text{Pref}(x, y) \leq k'$,

where $\{g_{i1}, \dots, g_{ir}\}, \{g_{j1}, \dots, g_{js}\} \subseteq C$; e.g., 'if car x is much worse than y with respect to price and much better with respect to comfort, then x is indifferent to or better than y , and there is not enough information to distinguish between the two situations'; these rules are supported by pairs of objects from the intersection of the P -upper approximation of sets $\text{Pref}^{\geq k}$ and $\text{Pref}^{\leq h}$ ($h < k$) only.

15.6.4 Fuzzy Preferences

Let us consider the case where the preferences $\text{Pref}_i[g_i(x), g_i(y)]$ with respect to each criterion $g_i \in C$, as well as the comprehensive preference $\text{Pref}(x, y)$, can assume values from a finite set. For example, given $x, y \in A$, the preferences $\text{Pref}_i[g_i(x), g_i(y)]$ and $\text{Pref}(x, y)$ can assume the following qualitatively ordinal values: x is much better than y , x is better than y , x is equivalent to y , x is worse than y , and x is much worse than y . Let us suppose, moreover, that each possible value of $\text{Pref}_i[g_i(x), g_i(y)]$ and $\text{Pref}(x, y)$ is fuzzy in the sense that it is true to some degree between 0 and 100%; e.g., ' x is better than y on criterion g_i with credibility 75%', or ' x is comprehensively worse than y with credibility 80%'. Greco *et al.* [3] proved that the fuzzy comprehensive preference $\text{Pref}(x, y)$ can be approximated by means of fuzzy preferences $\text{Pref}_i[g_i(x), g_i(y)]$ after translating the dominance-based rough approximations of S_{PCT} defined for the crisp case by means of fuzzy operators.

15.6.5 Preferences without Degree of Preferences

The values of $\text{Pref}_i[g_i(x), g_i(y)]$ considered in the dominance-based rough approximation of S_{PCT} represent a degree (strength) of preference. It is possible, however, that in some cases, the concept of degree of preference with respect to some criteria is meaningless for a DM. In these cases, there does not exist a function $\text{Pref}_i[g_i(x), g_i(y)]$, expressing how much x is better than y with respect to criterion g_i and, on the contrary, we can directly deal with values $g_i(x)$ and $g_i(y)$ only. For example, let us consider a decision problem concerning four cars x, y, w , and z with the maximum speed of 210 km/h, 180 km/h, 150 km/h, and 140 km/h, respectively. Even if the concept of degree of preference is meaningless, it is possible to say that with respect to the maximum speed, x is preferred to z at least as much as y is preferred to w . On the basis of this observation, Greco *et al.* [3] proved that comprehensive preference $\text{Pref}(x, y)$ can be approximated by means of criteria with only ordinal scales, for which the concept of degree of preference is meaningless. An example of decision rules obtained in this situation is the following:

'If car x has a maximum speed of at least 180 km/h, while car y has a maximum speed of at most 140 km/h, and the comfort of car x is at least good, while the comfort of car y is at most medium, then car x is at least as good as car y '.

15.7 Dominance-Based Rough Approximation of a Fuzzy Set

In this section, we show how the DRSA can be used for rough approximation of fuzzy sets.

A *fuzzy information base* is the 3-tuple $\mathcal{B} = \langle U, F, \varphi \rangle$, where U is a finite set of *objects* (universe), $F = \{f_1, f_2, \dots, f_m\}$ is a finite set of *properties*, and $\varphi : U \times F \rightarrow [0, 1]$ is a function such that $\varphi(x, f_h) \in [0, 1]$ expresses the credibility that object x has property f_h . Each object x from U is described by a vector

$$Des_F(x) = [\varphi(x, f_1), \dots, \varphi(x, f_m)]$$

called *description* of x in terms of the degrees to which it has properties from F ; it represents the available information about x . Obviously, $x \in U$ can be described in terms of any non-empty subset $E \subseteq F$ and in this case we have

$$Des_E(x) = [\varphi(x, f_h), f_h \in E].$$

For any $E \subseteq F$, we can define the dominance relation D_E as follows: for any $x, y \in U$, x dominates y with respect to E (denotation $x D_E y$) if, for any $f_h \in E$,

$$\varphi(x, f_h) \geq \varphi(y, f_h).$$

Given $E \subseteq F$ and $x \in U$, let

$$D_E^+(x) = \{y \in U : y D_E x\}, \quad D_E^-(x) = \{y \in U : x D_E y\}.$$

Let us consider a fuzzy set X in U , with its membership function $\mu_X : U \rightarrow [0, 1]$. For each cutting level $\alpha \in [0, 1]$ and for $* \in \{\geq, >\}$, we can define the E -lower and the E -upper approximation of $X^{*\alpha} = \{y \in U : \mu_X(y) * \alpha\}$ with respect to $E \subseteq F$ (denotation $\underline{E}(X^{*\alpha})$ and $\overline{E}(X^{*\alpha})$, respectively), as

$$\underline{E}(X^{*\alpha}) = \{x \in U : D_E^+(x) \subseteq X^{*\alpha}\} = \bigcup_{x \in U} \{D_E^+(x) : D_E^+(x) \subseteq X^{*\alpha}\},$$

$$\overline{E}(X^{*\alpha}) = \{x \in U : D_E^-(x) \cap X^{*\alpha} \neq \emptyset\} = \bigcup_{x \in U} \{D_E^-(x) : D_E^-(x) \cap X^{*\alpha} \neq \emptyset\}.$$

Analogously, for each cutting level $\alpha \in [0, 1]$ and for $\diamond \in \{\leq, <\}$, we define the E -lower and the E -upper approximation of $X^{\diamond\alpha} = \{y \in U : \mu_X(y) \diamond \alpha\}$ with respect to $E \subseteq F$ (denotation $\underline{E}(X^{\diamond\alpha})$ and $\overline{E}(X^{\diamond\alpha})$, respectively), as

$$\underline{E}(X^{\diamond\alpha}) = \{x \in U : D_E^-(x) \subseteq X^{\diamond\alpha}\} = \bigcup_{x \in U} \{D_E^-(x) : D_E^-(x) \subseteq X^{\diamond\alpha}\},$$

$$\overline{E}(X^{\diamond\alpha}) = \{x \in U : D_E^+(x) \cap X^{\diamond\alpha} \neq \emptyset\} = \bigcup_{x \in U} \{D_E^+(x) : D_E^+(x) \cap X^{\diamond\alpha} \neq \emptyset\}.$$

Let us remark that we can rewrite the rough approximations $\underline{E}(X^{\geq\alpha})$, $\overline{E}(X^{\geq\alpha})$, $\underline{E}(X^{\leq\alpha})$, and $\overline{E}(X^{\leq\alpha})$ as follows:

$$\underline{E}(X^{\geq\alpha}) = \{x \in U : \forall w \in U, w D_E x \Rightarrow w \in X^{\geq\alpha}\},$$

$$\overline{E}(X^{\geq\alpha}) = \{x \in U : \exists w \in U \text{ such that } x D_E w \text{ and } w \in X^{\geq\alpha}\},$$

$$\underline{E}(X^{\leq\alpha}) = \{x \in U : \forall w \in U, x D_E w \Rightarrow w \in X^{\leq\alpha}\},$$

$$\overline{E}(X^{\leq\alpha}) = \{x \in U : \exists w \in U \text{ such that } w D_E x \text{ and } w \in X^{\leq\alpha}\}.$$

Rough approximations $\underline{E}(X^{>\alpha})$, $\overline{E}(X^{>\alpha})$, $\underline{E}(X^{<\alpha})$, and $\overline{E}(X^{<\alpha})$ can be rewritten analogously by a simple replacement of ' \geq ' with ' $>$ ' and ' \leq ' with ' $<$ '.

This reformulation of the rough approximations is concordant with the syntax of decision rules obtained in DRSA. For example, $\underline{E}(X^{\geq\alpha})$ is concordant with decision rules of the type

'if object y has property f_{i1} to degree at least h_{i1} and has property f_{i2} to degree at least h_{i2} , ..., and has property f_{im} to degree at least h_{im} , then object y belongs to set X to degree at least α' ',

where $\{i1, \dots, im\} = E$ and $h_{i1} = \varphi(x, f_{i1}), \dots, h_{im} = \varphi(x, f_{im})$.

Let us remark that in the above approximations, even if $X^{\geq\alpha} = Y^{\leq\alpha}$, their approximations are, in general, different due to the different directions of cutting the membership functions of X and Y . Of course, a similar remark holds also for $X^{>\alpha}$ and $Y^{<\alpha}$. Considerations of the directions in the cuts $X^{\geq\alpha}$, $X^{>\alpha}$ and $X^{\leq\alpha}$, $X^{<\alpha}$ are important in the definition of the rough approximations of unions and intersections of cuts.

The rough approximations $\underline{E}(X^{\geq\alpha})$, $\overline{E}(X^{\geq\alpha})$, $\underline{E}(X^{\leq\alpha})$, $\overline{E}(X^{\leq\alpha})$ and $\underline{E}(X^{>\alpha})$, $\overline{E}(X^{>\alpha})$, $\underline{E}(X^{<\alpha})$, $\overline{E}(X^{<\alpha})$ satisfy the following inclusion properties: for any $0 \leq \alpha \leq 1$,

$$\begin{aligned}\underline{E}(X^{\geq\alpha}) &\subseteq X^{\geq\alpha} \subseteq \overline{E}(X^{\geq\alpha}), & \underline{E}(X^{\leq\alpha}) &\subseteq X^{\leq\alpha} \subseteq \overline{E}(X^{\leq\alpha}), \\ \underline{E}(X^{>\alpha}) &\subseteq X^{>\alpha} \subseteq \overline{E}(X^{>\alpha}), & \underline{E}(X^{<\alpha}) &\subseteq X^{<\alpha} \subseteq \overline{E}(X^{<\alpha}).\end{aligned}$$

Furthermore, the following complementary properties hold: for any $0 \leq \alpha \leq 1$,

$$\begin{aligned}\underline{E}(X^{\geq\alpha}) &= U - \overline{E}(X^{<\alpha}), & \underline{E}(X^{\leq\alpha}) &= U - \overline{E}(X^{>\alpha}), \\ \underline{E}(X^{>\alpha}) &= U - \overline{E}(X^{\leq\alpha}), & \underline{E}(X^{<\alpha}) &= U - \overline{E}(X^{\geq\alpha}).\end{aligned}$$

The following properties of monotonicity with respect to sets of properties also hold: for any $E_1 \subseteq E_2 \subseteq F$ and for any $0 \leq \alpha \leq 1$,

$$\begin{aligned}\underline{E}_1(X^{\geq\alpha}) &\subseteq \underline{E}_2(X^{\geq\alpha}), & \underline{E}_1(X^{>\alpha}) &\subseteq \underline{E}_2(X^{>\alpha}), \\ \underline{E}_1(X^{\leq\alpha}) &\subseteq \underline{E}_2(X^{\leq\alpha}), & \underline{E}_1(X^{<\alpha}) &\subseteq \underline{E}_2(X^{<\alpha}), \\ \overline{E}_1(X^{\geq\alpha}) &\supseteq \overline{E}_2(X^{\geq\alpha}), & \overline{E}_1(X^{>\alpha}) &\supseteq \overline{E}_2(X^{>\alpha}), \\ \overline{E}_1(X^{\leq\alpha}) &\supseteq \overline{E}_2(X^{\leq\alpha}), & \overline{E}_1(X^{<\alpha}) &\supseteq \overline{E}_2(X^{<\alpha}).\end{aligned}$$

We also consider fuzzy rough approximations \underline{X}_E^\uparrow , $\underline{X}_E^\downarrow$, \overline{X}_E^\uparrow , and $\overline{X}_E^\downarrow$, which are fuzzy sets with membership functions defined, respectively, as follows: for any $y \in U$,

$$\begin{aligned}\mu_{\underline{X}_E^\uparrow}(y) &= \max\{\alpha \in [0, 1] : y \in \underline{E}(X^{\geq\alpha})\}, \\ \mu_{\underline{X}_E^\downarrow}(y) &= \min\{\alpha \in [0, 1] : y \in \underline{E}(X^{\leq\alpha})\}, \\ \mu_{\overline{X}_E^\uparrow}(y) &= \max\{\alpha \in [0, 1] : y \in \overline{E}(X^{\geq\alpha})\}, \\ \mu_{\overline{X}_E^\downarrow}(y) &= \min\{\alpha \in [0, 1] : y \in \overline{E}(X^{\leq\alpha})\}.\end{aligned}$$

$\mu_{\underline{X}_E^\uparrow}(y)$ is defined as the upward lower fuzzy rough approximation of X with respect to E and can be interpreted in the following way. For any $\alpha, \beta \in [0, 1]$, we have that $\alpha < \beta$ implies $X^{\geq\alpha} \supseteq X^{\geq\beta}$. Therefore, the greater the cutting level α , the smaller $X^{\geq\alpha}$ and, consequently, the smaller also its lower approximation $\underline{E}(X^{\geq\alpha})$. Thus, for each $y \in U$ and for each fuzzy set X , there is a threshold $k(y)$, $0 \leq k(y) \leq \mu_X(y)$, such that $y \in \underline{E}(X^{\geq\alpha})$ if $\alpha \leq k(y)$ and $y \notin \underline{E}(X^{\geq\alpha})$ if $\alpha > k(y)$. Since $k(y) = \mu_{\underline{X}_E^\uparrow}(y)$, this explains the interest of $\mu_{\underline{X}_E^\uparrow}(y)$. Analogous interpretation holds for $\mu_{\overline{X}_E^\uparrow}(y)$ defined as the upward upper fuzzy rough approximation of X with respect to E .

$\mu_{\underline{X}_E^\downarrow}(y)$ is defined as the downward lower fuzzy rough approximation of X with respect to E and can be interpreted as follows. For any $\alpha, \beta \in [0, 1]$, we have that $\alpha < \beta$ implies $X^{\leq\alpha} \subseteq X^{\leq\beta}$. Therefore, the greater the cutting level α , the greater $X^{\leq\alpha}$ and, consequently, its lower approximation $\underline{E}(X^{\leq\alpha})$. Thus, for each $y \in U$ and for each fuzzy set X , there is a threshold $h(y)$, $\mu_X(y) \leq h(y) \leq 1$, such that $y \in \underline{E}(X^{\leq\alpha})$ if $\alpha \geq h(y)$ and $y \notin \underline{E}(X^{\leq\alpha})$ if $\alpha < h(y)$. We have that $h(y) = \mu_{\underline{X}_E^\downarrow}(y)$. Analogous interpretation holds for $\mu_{\overline{X}_E^\downarrow}(y)$ defined as the downward upper fuzzy rough approximation of X with respect to E .

The upward and downward lower and upper fuzzy rough approximations can also be rewritten in the following equivalent formulation, which has been proposed and investigated by Greco *et al.* [27]:

$$\begin{aligned}\mu_{\underline{X}_E^+}(y) &= \min\{\mu_X(z) : z \in D_E^+(y)\}, & \mu_{\overline{X}_E^+}(y) &= \max\{\mu_X(z) : z \in D_E^-(y)\}, \\ \mu_{\underline{X}_E^-}(y) &= \max\{\mu_X(z) : z \in D_E^-(y)\}, & \mu_{\overline{X}_E^-}(y) &= \min\{\mu_X(z) : z \in D_E^+(y)\}.\end{aligned}$$

The fuzzy rough approximations $\mu_{\underline{X}_E^+}(y)$, $\mu_{\overline{X}_E^+}(y)$, $\mu_{\underline{X}_E^-}(y)$, and $\mu_{\overline{X}_E^-}(y)$ satisfy the following inclusion properties: for any $y \in U$,

$$\mu_{\underline{X}_E^+}(y) \leq \mu_X(y) \leq \mu_{\overline{X}_E^+}(y), \quad \mu_{\underline{X}_E^-}(y) \leq \mu_X(y) \leq \mu_{\overline{X}_E^-}(y).$$

Furthermore, the following complementary properties hold: for any $y \in U$,

$$\mu_{\underline{X}_E^+}(y) = \mu_{\overline{X}_E^-}(y), \quad \mu_{\underline{X}_E^-}(y) = \mu_{\overline{X}_E^+}(y).$$

The following properties of monotonicity with respect to sets of properties also hold: for any $E_1 \subseteq E_2 \subseteq F$ and, for any $0 \leq \alpha \leq 1$,

$$\begin{aligned}\mu_{\underline{X}_{E_1}^+}(y) &\leq \mu_{\underline{X}_{E_2}^+}(y), & \mu_{\underline{X}_{E_1}^-}(y) &\geq \mu_{\underline{X}_{E_2}^-}(y), \\ \mu_{\overline{X}_{E_1}^+}(y) &\geq \mu_{\overline{X}_{E_2}^+}(y), & \mu_{\overline{X}_{E_1}^-}(y) &\leq \mu_{\overline{X}_{E_2}^-}(y).\end{aligned}$$

15.8 Monotonic Rough Approximation of a Fuzzy Set versus Classical Rough Set

What is the relationship between classical rough set and DRSA approximation of a fuzzy set? Greco *et al.* [16, 37] proved that the former is a particular case of the latter. In the following we demonstrate this relationship.

Any information system can be expressed in terms of a specific type of an information base (see Section 15.2). An *information base* is called *Boolean* if $\varphi : U \times F \rightarrow \{0, 1\}$. A partition $F = \{F_1, \dots, F_r\}$ of the set of properties F , with $\text{card}(F_k) \geq 2$ for all $k = 1, \dots, r$, is called *canonical* if, for each $x \in U$ and for each $F_k \subseteq F$, $k = 1, \dots, r$, there exists only one $f_j \in F_k$ for which $\varphi(x, f_j) = 1$ (and, therefore, for each $f_i \in F_k - \{f_j\}$, $\varphi(x, f_i) = 0$). The condition $\text{card}(F_k) \geq 2$ for all $k = 1, \dots, r$ is necessary because, otherwise, we would have at least one element of the partition $F_k = \{f'\}$ such that $\varphi(x, f') = 1$ for all $x \in U$, and this would mean that property f' gives no information and can be removed. We can observe now that any *information system* $S = < U, Q, V, \phi >$ can be transformed to a Boolean information base $B = < U, F, \varphi >$ assigning to each $v \in V_q$, $q \in Q$, one property $f_{qv} \in F$ such that $\varphi(x, f_{qv}) = 1$ if $\phi(x, q) = v$ and $\varphi(x, f_{qv}) = 0$ otherwise. Let us remark that $F = \{F_1, \dots, F_r\}$, with $F_q = \{f_{qv}, v \in V_q\}$, $q \in Q$, is a canonical partition of F . The opposite transformation, from a Boolean information base to an information system, is not always possible; i.e., there may exist Boolean information bases which cannot be transformed into information systems, because their sets of properties do not admit any canonical partition, as shown by the following example.

Example. Let us consider a Boolean information base, such that $U = \{x_1, x_2, x_3\}$, $F = \{f_1, f_2\}$, and function φ is defined by Table 15.1. One can see that $F = \{f_1, f_2\}$ is not a canonical partition because $\varphi(x_3, f_1) = \varphi(x_3, f_2) = 1$, while definition of canonical partition F does not allow that for an object $x \in U$, $\varphi(x, f_1) = \varphi(x, f_2) = 1$. Therefore, this Boolean information base has no equivalent information system. Let us remark that also the Boolean information base presented in Table 15.2, where $U = \{x_1, x_2, x_4\}$ and $F = \{f_1, f_2\}$, cannot be transformed to an information system, because partition $F = \{f_1, f_2\}$ is not canonical. Indeed, $\varphi(x_4, f_1) = \varphi(x_4, f_2) = 0$, while definition of canonical partition F does not allow that for an object $x \in U$, $\varphi(x, f_1) = \varphi(x, f_2) = 0$.

Table 15.1 Information base \mathbf{B}

	f_1	f_2
x_1	0	1
x_2	1	0
x_3	1	1

Table 15.2 Information base \mathbf{B}'

	f_1	f_2
x_1	0	1
x_2	1	0
x_4	0	0

The above says that consideration of rough approximation in the context of a Boolean information base is more general than the same consideration in the context of an information system. This means, of course, that the rough approximation considered in the context of a fuzzy information base is yet more general.

It is worth stressing that the Boolean information bases \mathbf{B} and \mathbf{B}' are not Boolean information systems. In fact, on one hand, a Boolean information base provides information about absence ($\varphi(x, f) = 0$) or presence ($\varphi(x, f) = 1$) of properties $f \in F$ in objects $x \in U$. On the other hand, a Boolean information system provides information about values assigned by attributes $q \in Q$, whose sets of values are $V_q = \{0, 1\}$, to objects $x \in U$, such that $\phi(x, q) = 1$ or $\phi(x, q) = 0$ for all $x \in U$ and $q \in Q$. Observe, therefore, that to transform a Boolean information system \mathbf{S} into a Boolean information base \mathbf{B} , each attribute q of \mathbf{S} corresponds to two properties f_{q0} and f_{q1} of \mathbf{B} , such that for all $x \in U$,

- $\varphi(x, f_{q0}) = 1$ and $\varphi(x, f_{q1}) = 0$ if $\phi(x, q) = 0$,
- $\varphi(x, f_{q0}) = 0$ and $\varphi(x, f_{q1}) = 1$ if $\phi(x, q) = 1$.

Thus, the Boolean information base \mathbf{B} in Table 15.1 and the Boolean information system \mathbf{S} in Table 15.3 are different, despite that they could seem identical. In fact, the Boolean information system \mathbf{S} in Table 15.3 can be transformed into the Boolean information base \mathbf{B}'' in Table 15.4, which is clearly different from \mathbf{B} .

The equivalence between rough approximations in the context of a fuzzy information base and the classical definition of rough approximations in the context of an information system can be stated as follows. Let us consider an information system and the corresponding Boolean information base; for each $P \subseteq Q$, let E^P be the set of all the properties corresponding to values v of attributes in P . Let X be a non-fuzzy set in U (i.e., $\mu_X : U \rightarrow \{0, 1\}$ and, therefore, for any $y \in U$, $\mu_X(y) = 1$ or $\mu_X(y) = 0$). Then, we have

$$\begin{aligned}\underline{E^P}(X^{\geq 1}) &= P(X^{\geq 1}), & \overline{E^P}(X^{\geq 1}) &= \overline{P}(X^{\geq 1}), \\ \underline{E^P}(X^{\leq 0}) &= P(U - X^{\geq 1}), & \overline{E^P}(X^{\leq 0}) &= \overline{P}(U - X^{\geq 1}),\end{aligned}$$

where for any $Y \subseteq U$, $\overline{P}(Y)$ and $\underline{P}(Y)$ represent classical lower and upper approximations of Y with respect to $P \subseteq Q$ in the information system \mathbf{S} [1],[2].

This result proves that the rough approximation of a non-fuzzy set $X \subseteq U$ in a Boolean information base admitting a canonical partition is equivalent to the classical rough approximation of set X in the corresponding information system. Therefore, the classical rough approximation is a particular case of the dominance-based rough approximation in a fuzzy information base.

Table 15.3 Information system \mathbf{S}

	q_1	q_2
x_1	0	1
x_2	1	0
x_3	1	1

Table 15.4 Information base \mathbf{B}''

	f_{q_10}	f_{q_11}	f_{q_20}	f_{q_21}
x_1	1	0	0	1
x_2	0	1	1	0
x_3	0	1	0	1

15.9 Dominance-Based Rough Set Approach to Case-Based Reasoning

In this section, we consider rough approximation of a fuzzy set using a similarity relation in the context of case-based reasoning [28].

Case-based reasoning (for a general introduction to case-based reasoning see, e.g., [38]; for a fuzzy set approach to case-based reasoning see [39]) is a paradigm in machine learning whose idea is that a new problem can be solved by noticing its similarity to a set of problems previously solved. Case-based reasoning regards the inference of some proper conclusions related to a new situation by the analysis of similar cases from a memory of previous cases. It is based on two principles [40]:

1. Similar problems have similar solutions.
2. Types of encountered problems tend to recur.

Gilboa and Schmeidler [41] observed that the basic idea of case-based reasoning can be found in the following sentence of Hume [42]: ‘From causes which appear *similar* we expect similar effects. This is the sum of all our experimental conclusions’. Rephrasing Hume, one can say that ‘the more similar are the causes, the more similar one expects the effects’. Therefore, measuring similarity is the essential point of all case-based reasoning and, particularly, of fuzzy set approach to case-based reasoning [39]. This explains the many problems that measuring similarity generates within case-based reasoning. Problems of modeling similarity are relative to two levels:

1. At the level of similarity with respect to single features: How to define a meaningful similarity measure with respect to a single feature?
2. At the level of similarity with respect to all features: How to properly aggregate the similarity measure with respect to single features in order to obtain a comprehensive similarity measure?

For the above reasons, we proposed in [28] a DRSA to case-based reasoning, which tries to be possibly ‘neutral’ and ‘objective’ with respect to similarity relation, in the sense that at the level of similarity concerning single features, we consider only ordinal properties of similarity, and at the level of aggregation, we do not impose any particular functional aggregation based on some very specific axioms (see, e.g., [41]), but we consider a set of decision rules based on the general monotonicity property of comprehensive similarity with respect to similarity of single features. Therefore, our approach to case-based reasoning is very little ‘invasive’, comparing to the many other existing approaches.

Let us consider a *pairwise fuzzy information base* being the 3-tuple

$$\mathcal{B} = \langle U, F, \sigma \rangle,$$

where U is a finite set of *objects* (universe), $F = \{f_1, f_2, \dots, f_m\}$ is a finite set of *features*, and $\sigma : U \times U \times F \rightarrow [0,1]$ is a function such that $\sigma(x, y, f_h) \in [0,1]$ expresses the credibility that object x is similar to object y with respect to feature f_h . The minimal requirement function σ must satisfy is that, for all $x \in U$ and for all $f_h \in F$, $\sigma(x, x, f_h) = 1$. Therefore, each pair of objects $(x, y) \in U \times U$ is described by a vector

$$Des_F(x, y) = [\sigma(x, y, f_1), \dots, \sigma(x, y, f_m)]$$

called *description* of (x, y) in terms of the credibilities of similarity with respect to features from F ; it represents the available information about similarity between x and y . Obviously, similarity between x and y , $x, y \in U$ can be described in terms of any non-empty subset $E \subseteq F$, and in this case we have

$$Des_E(x, y) = [\sigma(x, y, f_h), f_h \in E].$$

With respect to any $E \subseteq F$, we can define the dominance relation D_E on $U \times U$ as follows: for any $x, y, w, z \in U$, (x, y) dominates (w, z) with respect to E (denotation $(x, y)D_E(w, z)$) if, for any $f_h \in E$,

$$\sigma(x, y, f_h) \geq \sigma(w, z, f_h).$$

Given $E \subseteq F$ and $x, y \in U$, let

$$D_E^+(y, x) = \{w \in U : (w, x)D_E(y, x)\},$$

$$D_E^-(y, x) = \{w \in U : (y, x)D_E(w, x)\}.$$

In the pair (y, x) , x is considered to be a *reference object*, while y can be called a *limit object*, because it is conditioning the membership of w in $D_E^+(y, x)$ and in $D_E^-(y, x)$.

For each $x \in U$, $\alpha \in [0,1]$ and $* \in \{\geq, >\}$, we can define the lower approximation of $X^{*\alpha}$, $\underline{E}_\sigma(X^{*\alpha})$, and the upper approximation of $X^{*\alpha}$, $\overline{E}_\sigma(X^{*\alpha})$, based on similarity σ with respect to $E \subseteq F$ and x , respectively, as

$$\underline{E}(x)_\sigma(X^{*\alpha}) = \{y \in U : D_E^+(y, x) \subseteq X^{*\alpha}\},$$

$$\overline{E}(x)_\sigma(X^{*\alpha}) = \{y \in U : D_E^-(y, x) \cap X^{*\alpha} \neq \emptyset\}.$$

For the sake of simplicity, in the following we shall consider $\underline{E}(x)_\sigma(X^{\geq\alpha})$ and $\overline{E}(x)_\sigma(X^{\geq\alpha})$ with $x \in X^{\geq\alpha}$. Of course, analogous considerations hold for $\underline{E}(x)_\sigma(X^{>\alpha})$ and $\overline{E}(x)_\sigma(X^{>\alpha})$. Let us remark that the lower approximation of $X^{\geq\alpha}$ with respect to x contains all the objects $y \in U$ such that any object w , being similar to x at least as much as y is similar to x with respect to all the considered features $E \subseteq F$, also belongs to $X^{\geq\alpha}$. Thus, the data from the fuzzy pairwise information base B confirm that if w is similar to x not less than $y \in \underline{E}(x)_\sigma(X^{\geq\alpha})$ is similar to x with respect to all the considered features $E \subseteq F$, then w belongs to $X^{\geq\alpha}$. In other words, x is a reference object and $y \in \underline{E}(x)_\sigma(X^{\geq\alpha})$ is a limit object which ‘certainly’ belongs to set X with credibility at least α ; the limit is understood such that all objects w that are similar to x with respect to considered features at least as much as y is similar to x also belong to X with credibility at least α .

Analogously, the upper approximation of $X^{\geq\alpha}$ with respect to x contains all objects $y \in U$ such that there is at least one object w , being similar to x at most as much as y is similar to x with respect to all the considered features $E \subseteq F$, which belongs to $X^{\geq\alpha}$. Thus, the data from the fuzzy pairwise information base B confirm that if w is similar to x not less than $y \in \overline{E}(x)_\sigma(X^{\geq\alpha})$ is similar to x with respect to all the considered features $E \subseteq F$, then it is possible that w belongs to $X^{\geq\alpha}$. In other words, x is a reference object and $y \in \overline{E}(x)_\sigma(X^{\geq\alpha})$ is a limit object, which ‘possibly’ belongs to set X with credibility at least α ; the limit is understood such that all objects $z \in U$ similar to x not less than y with respect to considered features possibly belong to $X^{\geq\alpha}$.

For each $x \in U$ and $\alpha \in [0,1]$ and $\diamond \in \{\leq, <\}$, we can define the lower approximation of $X^{\diamond\alpha}$, $\underline{E}(x)_\sigma(X^{\diamond\alpha})$, and the upper approximation of $X^{\diamond\alpha}$, $\overline{E}(x)_\sigma(X^{\diamond\alpha})$, based on similarity σ with respect to $E \subseteq F$ and x , respectively, as

$$\underline{E}(x)_\sigma(X^{\diamond\alpha}) = \{y \in U : D_E^-(y, x) \subseteq X^{\diamond\alpha}\},$$

$$\overline{E}(x)_\sigma(X^{\diamond\alpha}) = \{y \in U : D_E^+(y, x) \cap X^{\diamond\alpha} \neq \emptyset\}.$$

For the sake of simplicity, in the following we shall consider $\underline{E}(x)_\sigma(X^{\leq\alpha})$ and $\overline{E}(x)_\sigma(X^{\leq\alpha})$ with $x \in X^{\leq\alpha}$. Of course, analogous considerations hold for $\underline{E}(x)_\sigma(X^{<\alpha})$ and $\overline{E}(x)_\sigma(X^{<\alpha})$. Let us remark that the lower approximation of $X^{\leq\alpha}$ with respect to x contains all the objects $y \in U$ such that any object w , being similar to x at most as much as y is similar to x with respect to all the considered features $E \subseteq F$, also belongs to $X^{\leq\alpha}$. Thus, the data from the fuzzy pairwise information base B confirm that if w is similar to x not more than $y \in \underline{E}(x)_\sigma(X^{\leq\alpha})$ is similar to x with respect to all the considered features $E \subseteq F$, then w belongs to $X^{\leq\alpha}$. In other words, x is a reference object and $y \in \underline{E}(x)_\sigma(X^{\leq\alpha})$ is a limit object, which ‘certainly’ belongs to set X with credibility at most α ; the limit is understood such that all

objects w that are similar to x with respect to considered features at most as much as y is similar to x also belong to X with credibility at most α .

Analogously, the upper approximation of $X^{\leq\alpha}$ with respect to x contains all the objects $y \in U$ such that there is at least one object w , being similar to x at least as much as y is similar to x with respect to all the considered features $E \subseteq F$, which belongs to $X^{\leq\alpha}$. Thus, the data from the fuzzy pairwise information base B confirm that if w is similar to x not more than $y \in \overline{E}(x)_\sigma(X^{\leq\alpha})$ is similar to x with respect to all the considered features $E \subseteq F$, then it is possible that w belongs to $X^{\leq\alpha}$. In other words, x is a reference object and $y \in \overline{E}_\sigma(X^{\leq\alpha})$ is a limit object, which ‘possibly’ belongs to set X with credibility at most α ; the limit is understood such that all objects $z \in U$ similar to x not more than y with respect to considered features possibly belong to $X^{\leq\alpha}$.

Let us remark that we can rewrite the rough approximations $\underline{E}(x)_\sigma(X^{\geq\alpha})$, $\overline{E}(x)_\sigma(X^{\geq\alpha})$, $\underline{E}(x)_\sigma(X^{\leq\alpha})$, and $\overline{E}(x)_\sigma(X^{\leq\alpha})$ as follows:

$$\underline{E}(x)_\sigma(X^{\geq\alpha}) = \{y \in U : \forall w \in U, (w, x)D_E(y, x) \Rightarrow w \in X^{\geq\alpha}\},$$

$$\overline{E}(x)_\sigma(X^{\geq\alpha}) = \{y \in U : \exists w \in U \text{ such that } (y, x)D_E(w, x) \text{ and } w \in X^{\geq\alpha}\},$$

$$\underline{E}(x)_\sigma(X^{\leq\alpha}) = \{y \in U : \forall w \in U, (y, x)D_E(w, x) \Rightarrow w \in X^{\leq\alpha}\},$$

$$\overline{E}(x)_\sigma(X^{\leq\alpha}) = \{y \in U : \exists w \in U \text{ such that } (w, x)D_E(y, x) \text{ and } w \in X^{\leq\alpha}\}.$$

This formulation of the rough approximation is concordant with the syntax of the decision rules induced by means of DRSA from a fuzzy pairwise information base. More precisely,

- $\underline{E}(x)_\sigma(X^{\geq\alpha})$ is concordant with decision rules of the type
‘if object w is similar to object x with respect to feature f_{i1} to degree at least h_{i1} and with respect to feature f_{i2} to degree at least h_{i2} and … and with respect to feature f_{im} to degree at least h_{im} , then object w belongs to set X to degree at least α ’.
- $\overline{E}(x)_\sigma(X^{\geq\alpha})$ is concordant with decision rules of the type
‘if object w is similar to object x with respect to feature f_{i1} to degree at least h_{i1} and with respect to feature f_{i2} to degree at least h_{i2} and … and with respect to feature f_{im} to degree at least h_{im} , then object w could belong to set X to degree at least α ’.
- $\underline{E}(x)_\sigma(X^{\leq\alpha})$ is concordant with decision rules of the type
‘if object w is similar to object x with respect to feature f_{i1} to degree at most h_{i1} and with respect to feature f_{i2} to degree at most h_{i2} and … and with respect to feature f_{im} to degree at most h_{im} , then object w belongs to set X to degree at most α ’.
- $\overline{E}(x)_\sigma(X^{\leq\alpha})$ is concordant with decision rules of the type
*‘if object w is similar to object x with respect to feature f_{i1} to degree at most h_{i1} and with respect to feature f_{i2} to degree at most h_{i2} and … and with respect to feature f_{im} to degree at least h_{im} , then object w could belong to set X to degree at most α ’,
where $\{i1, \dots, im\} = E$ and $h_{i1}, \dots, h_{im} \in [0, 1]$.*

The above definitions of rough approximations and the syntax of decision rules are based on ordinal properties of similarity relations only. In fact, no algebraic operations, such as sum or product, involving cardinal properties of function σ measuring credibility of similarity relations are considered. This is an important characteristic of our approach in comparison with alternative approaches to case-based reasoning.

Let us remark that, similarly to DRSA approximation in an information base, in the case of DRSA approximation in a fuzzy pairwise information base, even if for two fuzzy sets X and Y we have $X^{\geq\alpha} = Y^{\leq\alpha}$, their approximations may be different due to the different directions of cutting the membership functions of sets X and Y .

Rough approximations in a fuzzy pairwise information base satisfy the following interesting properties:

Theorem. *Given a fuzzy pairwise information base $\mathbf{B} = \langle U, F, \sigma \rangle$ and a fuzzy set X in U with membership function $\mu_X(\cdot)$, the following properties hold for any $E \subseteq F$:*

1. For any $0 \leq \alpha \leq 1$,

$$\begin{aligned}\underline{E}(x)_\sigma(X^{\leq\alpha}) &\subseteq X^{\leq\alpha} \subseteq \overline{E}(x)_\sigma(X^{\leq\alpha}), \\ \underline{E}(x)_\sigma(X^{\geq\alpha}) &\subseteq X^{\geq\alpha} \subseteq \overline{E}(x)_\sigma(X^{\geq\alpha}), \\ \underline{E}(x)_\sigma(X^{<\alpha}) &\subseteq X^{<\alpha} \subseteq \overline{E}(x)_\sigma(X^{<\alpha}), \\ \underline{E}(x)_\sigma(X^{>\alpha}) &\subseteq X^{>\alpha} \subseteq \overline{E}(x)_\sigma(X^{>\alpha}).\end{aligned}$$

2. For any $0 \leq \alpha \leq 1$,

$$\underline{E}(x)_\sigma(X^{\leq\alpha}) = U - \overline{E}(x)_\sigma(X^{>\alpha}), \quad \underline{E}(x)_\sigma(X^{\geq\alpha}) = U - \overline{E}(x)_\sigma(X^{<\alpha}).$$

3. For any $0 \leq \alpha \leq \beta \leq 1$,

$$\begin{aligned}\underline{E}(x)_\sigma(X^{\leq\alpha}) &\subseteq \underline{E}(x)_\sigma(X^{\leq\beta}), & \underline{E}(x)_\sigma(X^{<\alpha}) &\subseteq \underline{E}(x)_\sigma(X^{<\beta}), \\ \underline{E}(x)_\sigma(X^{\geq\alpha}) &\supseteq \underline{E}(x)_\sigma(X^{\geq\beta}), & \underline{E}(x)_\sigma(X^{>\alpha}) &\supseteq \underline{E}(x)_\sigma(X^{>\beta}), \\ \overline{E}(x)_\sigma(X^{\leq\alpha}) &\subseteq \overline{E}(x)_\sigma(X^{\leq\beta}), & \overline{E}(x)_\sigma(X^{<\alpha}) &\subseteq \overline{E}(x)_\sigma(X^{<\beta}), \\ \overline{E}(x)_\sigma(X^{\geq\alpha}) &\supseteq \overline{E}(x)_\sigma(X^{\geq\beta}), & \overline{E}(x)_\sigma(X^{>\alpha}) &\supseteq \overline{E}(x)_\sigma(X^{>\beta}).\end{aligned}$$

4. For any $x, y, w, z \in U$ and for any $0 \leq \alpha \leq 1$,

$$\begin{aligned}[(y, x)D_E(w, x) \text{ and } w \in \underline{E}(x)_\sigma(X^{\geq\alpha})] &\Rightarrow y \in \underline{E}(x)_\sigma(X^{\geq\alpha}), \\ [(y, x)D_E(w, x) \text{ and } w \in \underline{E}(x)_\sigma(X^{>\alpha})] &\Rightarrow y \in \underline{E}(x)_\sigma(X^{>\alpha}), \\ [(y, x)D_E(w, x) \text{ and } w \in \overline{E}(x)_\sigma(X^{\geq\alpha})] &\Rightarrow y \in \overline{E}(x)_\sigma(X^{\geq\alpha}), \\ [(y, x)D_E(w, x) \text{ and } w \in \overline{E}(x)_\sigma(X^{>\alpha})] &\Rightarrow y \in \overline{E}(x)_\sigma(X^{>\alpha}), \\ [(w, x)D_E(y, x) \text{ and } w \in \underline{E}(x)_\sigma(X^{\leq\alpha})] &\Rightarrow y \in \underline{E}(x)_\sigma(X^{\leq\alpha}), \\ [(w, x)D_E(y, x) \text{ and } w \in \underline{E}(x)_\sigma(X^{<\alpha})] &\Rightarrow y \in \underline{E}(x)_\sigma(X^{<\alpha}), \\ [(w, x)D_E(y, x) \text{ and } w \in \overline{E}(x)_\sigma(X^{\leq\alpha})] &\Rightarrow y \in \overline{E}(x)_\sigma(X^{\leq\alpha}), \\ [(w, x)D_E(y, x) \text{ and } w \in \overline{E}(x)_\sigma(X^{<\alpha})] &\Rightarrow y \in \overline{E}(x)_\sigma(X^{<\alpha}).\end{aligned}$$

5. For any $E_1 \subseteq E_2 \subseteq F$ and for any $0 \leq \alpha \leq 1$,

$$\begin{aligned}\underline{E}_1(x)_\sigma(X^{\leq\alpha}) &\subseteq \underline{E}_2(x)_\sigma(X^{\leq\alpha}), & \underline{E}_1(x)_\sigma(X^{<\alpha}) &\subseteq \underline{E}_2(x)_\sigma(X^{<\alpha}), \\ \underline{E}_1(x)_\sigma(X^{\geq\alpha}) &\subseteq \underline{E}_2(x)_\sigma(X^{\geq\alpha}), & \underline{E}_1(x)_\sigma(X^{>\alpha}) &\subseteq \underline{E}_2(x)_\sigma(X^{>\alpha}), \\ \overline{E}_1(x)_\sigma(X^{\leq\alpha}) &\supseteq \overline{E}_2(x)_\sigma(X^{\leq\alpha}), & \overline{E}_1(x)_\sigma(X^{<\alpha}) &\supseteq \overline{E}_2(x)_\sigma(X^{<\alpha}), \\ \overline{E}_1(x)_\sigma(X^{\geq\alpha}) &\supseteq \overline{E}_2(x)_\sigma(X^{\geq\alpha}), & \overline{E}_1(x)_\sigma(X^{>\alpha}) &\supseteq \overline{E}_2(x)_\sigma(X^{>\alpha}).\end{aligned}$$

15.10 Conclusions and Further Research Directions

In this chapter, we considered the problem of granular computing with monotonically ordered data and we proposed the DRSA as a proper way of handling this kind of data. After a brief review of the classical IRSA and its fuzzy set extensions, we presented fuzzy set extensions of the DRSA and dominance-based rough approximations of fuzzy sets. The fuzzy set extensions of DRSA are based on fuzzy connectives, which is characteristic for almost all fuzzy rough set approaches. The dominance-based rough approximations of fuzzy sets infer, instead, the most cautious conclusions from available imprecise information, without

using any fuzzy connectives, which are always arbitrary to some extent. Another advantage of dominance-based rough approximations of fuzzy sets is that they use only ordinal properties of membership degrees. Knowledge induced from dominance-based rough approximations of fuzzy sets is represented in terms of gradual decision rules. The dominance-based rough approximations of fuzzy sets generalize the classical rough approximations of crisp sets, as proved by showing that the classical rough set approach is one of its particular cases. We believe that, due to considering only ordinal character of the graduality of fuzzy sets and due to eliminating all fuzzy connectives, the dominance-based rough approximations of fuzzy sets give a new insight into both rough sets and fuzzy sets and enable further generalizations of both of them. The recently proposed DRSA for fuzzy case-based reasoning is an example of this capacity. We believe that this chapter exhibits some important merits of DRSA within granular computing, which are the following:

- DRSA extends the paradigm of granular computing to problems involving ordered data.
- It specifies a syntax and modality of information granules, defined by means of dominance-based constraints, which are appropriate for dealing with ordered data.
- It provides a methodology for dealing with this type of information granules, which results in a theory of computing with words and reasoning about ordered data.

We consider the granular computing with ordered data as a very general problem, because also other modalities of information constraints, such as veristic, possibilistic, and probabilistic, have to deal with ordered value sets (with qualifiers relative to grades of truth, possibility, and probability). For this reason, we believe that granular computing with ordered data is a very promising research field, and we hope that this chapter may attract researchers and practitioners to this fascinating field of research and applications.

Acknowledgments

The research of the first two authors has been supported by the Italian Ministry of University and Scientific Research (MUR). The third author acknowledges financial support from the Ministry of Science and Higher Education.

References

- [1] Z. Pawlak. Rough sets. *Int. J. Comput. Inf. Sci.* 11 (1982) 341–356.
- [2] Z. Pawlak. *Rough Sets*. Kluwer, Dordrecht, 1991.
- [3] S. Greco, B. Matarazzo, and R. Słowiński. The use of rough sets and fuzzy sets in MCDM. In: T. Gal, T. Stewart, and T. Hanne (eds), *Advances in Multiple Criteria Decision Making*. Kluwer, Boston, 1999, chapter 14, pp. 14.1–14.59.
- [4] S. Greco, B. Matarazzo, and R. Słowiński. Rough sets theory for multicriteria decision analysis. *Eur. J. Oper. Res.* 129 (2001) 1–47.
- [5] S. Greco, B. Matarazzo, and R. Słowiński. Decision rule approach. In: J. Figueira, S. Greco, and M. Ehrgott (eds), *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer-Verlag, Berlin, 2005, chapter 13, pp. 507–563.
- [6] R. Słowiński, S. Greco, and B. Matarazzo. Rough set based decision support. In: E.K. Burke and G. Kendall (eds), *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer-Verlag, New York, 2005, chapter 16, pp. 475–527.
- [7] J. Figueira, S. Greco, and M. Ehrgott (eds). *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer-Verlag, Berlin, 2005.
- [8] S. Greco, B. Matarazzo, and R. Słowiński. Dominance-based rough set approach to knowledge discovery (I) – general perspective. In: N. Zhong, J. Liu (eds), *Intelligent Technologies for Information Analysis*. Springer-Verlag, Berlin, 2004, chapter 20, pp. 513–552.
- [9] S. Greco, B. Matarazzo, and R. Słowiński. Dominance-based rough set approach to knowledge discovery (II) – extensions and applications. In: N. Zhong and J. Liu (eds), *Intelligent Technologies for Information Analysis*. Springer-Verlag, Berlin, 2004, chapter 21, pp. 553–612.

- [10] L. Zadeh. From computing with numbers to computing with words – from manipulation of measurements to manipulation of perception. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* 45 (1999) 105–119.
- [11] S. Greco, B. Matarazzo, and R. Słowiński. Preference representation by means of conjoint measurement and decision rule model. In: D. Bouyssou, E. Jacquet-Lagrèze, P. Perny, R. Słowiński, D. Vanderpooten, and Ph. Vincke (eds), *Aiding Decisions with Multiple Criteria – Essays in Honor of Bernard Roy*. Kluwer, Dordrecht, 2002, pp. 263–313.
- [12] S. Greco, B. Matarazzo, and R. Słowiński. Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules. *Eur. J. Oper. Res.* 158 (2004) 271–292.
- [13] R. Słowiński, S. Greco, and B. Matarazzo. Axiomatization of utility, outranking and decision-rule preference models for multiple-criteria classification problems under partial inconsistency with the dominance principle. *Control Cybern.* 31 (2002) 1005–1035.
- [14] S. Greco, B. Predki, and R. Słowiński. Searching for an equivalence between decision rules and concordance-discordance preference model in multicriteria choice problems. *Control Cybern.* 31 (2002) 921–935.
- [15] S. Greco, B. Matarazzo, and R. Słowiński. A fuzzy extension of the rough set approach to multicriteria and multiattribute sorting. In: J. Fodor, B. De Baets, and P. Perny (eds), *Preferences and Decisions under Incomplete Information*. Physica-Verlag, Heidelberg, 2000, pp. 131–154.
- [16] S. Greco, B. Matarazzo, and R. Słowiński. Generalizing rough set theory through dominance-based rough set approach. In: D. Slezak, J. Yao, J. Peters, W. Ziarko, and X. Hu (eds), *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, LNAI 3642. Springer-Verlag, Berlin, 2005, pp. 1–11.
- [17] L. Zadeh. Fuzzy sets. *Inf. Control* 8 (1965) 338–353.
- [18] D. Dubois and H. Prade. Rough fuzzy sets and fuzzy rough sets. *Int. J. Gen. Syst.* 17 (23) (1990) 191–209.
- [19] D. Dubois and H. Prade. Putting rough sets and fuzzy sets together. In: R. Słowiński (ed.), *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer, Dordrecht, 1992, pp. 203–232.
- [20] D. Dubois, J. Grzymala-Busse, M. Inuiguchi, and L. Polkowski (eds). *Transactions on Rough Sets II: Rough Sets and Fuzzy Sets*, LNCS 3135. Springer-Verlag, Berlin, 2004.
- [21] A.M. Radzikowska and E.E. Kerre. A comparative study of fuzzy rough sets. *Fuzzy Sets Syst.* 126 (2002) 137–155.
- [22] J. Fodor and M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer, Dordrecht, 1994.
- [23] E.P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*. Kluwer, Dordrecht, 2000.
- [24] S. Greco, M. Inuiguchi, and R. Słowiński. Dominance-based rough set approach using possibility and necessity measures. In: J.J. Alpignani, J.F. Peters, A. Skowron, and N. Zhong (eds), *Rough Sets and Current Trends in Computing*, LNAI 2475. Springer-Verlag, Berlin, 2002, pp. 85–92.
- [25] T. Marchant. The measurement of membership by comparisons. *Fuzzy Sets Syst.* 148 (2004) 157–177.
- [26] S. Greco, M. Inuiguchi, and R. Słowiński. A new proposal for rough fuzzy approximations and decision rule representation. In: D. Dubois, J. Grzymala-Busse, M. Inuiguchi, and L. Polkowski (eds), *Transactions on Rough Sets II: Rough Sets and Fuzzy Sets*, LNCS 3135. Springer-Verlag, Berlin, 2004, pp. 156–164.
- [27] S. Greco, M. Inuiguchi, and R. Słowiński. Fuzzy rough sets and multiple-premise gradual decision rules. *Int. J. Approx. Reason.* 41 (2006) 179–211.
- [28] S. Greco, B. Matarazzo, and R. Słowiński. Dominance-based rough set approach to case-based reasoning. In: V. Torra, Y. Narukawa, A. Valls, and J. Domingo-Ferrer (eds), *Modelling Decisions for Artificial Intelligence*, LNAI 3885. Springer-Verlag, Berlin, 2006, pp. 7–18.
- [29] S. Greco, B. Matarazzo, R. Słowiński, and J. Stefanowski. Variable consistency model of dominance-based rough set approach. In: W. Ziarko and Y. Yao (eds), *Rough Sets and Current Trends in Computing*, LNAI 2005. Springer-Verlag, Berlin, 2001, pp. 170–181.
- [30] W. Ziarko. Variable precision rough sets model. *J. Comput. Syst. Sci.* 46 (1993) 39–59.
- [31] W. Ziarko. Rough sets as a methodology for data mining. In: L. Polkowski and A. Skowron (eds), *Rough Sets in Knowledge Discovery*, Vol. 1. Physica-Verlag, Heidelberg, 1998, pp. 554–576.
- [32] S. Greco, B. Matarazzo, and R. Słowiński. Rough set approach to decisions under risk. In: W. Ziarko and Y. Yao (eds), *Rough Sets and Current Trends in Computing*, LNAI 2005. Springer-Verlag, Berlin, 2001, pp. 160–169.
- [33] J. Dyer. MAUT – Multiattribute utility theory. In: J. Figueira, S. Greco, and M. Ehrgott (eds), *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer-Verlag, Berlin, 2005, chapter 7, pp. 266–294.
- [34] T. Stewart. Dealing with uncertainties in MCDA. In: J. Figueira, S. Greco, and M. Ehrgott (eds), *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer-Verlag, Berlin, 2005, chapter 11, pp. 445–470.

- [35] R. Słowiński, S. Greco, and B. Matarazzo. Mining decision-rule preference model from rough approximation of preference relation. In: *Proceedings of the 26th IEEE Annual Int. Conference on Computer Software & Applications (COMPSAC 2002)*, Oxford, 2002, pp. 1129–1134.
- [36] Ph. Fortemps, S. Greco, and R. Słowiński. Multicriteria decision support using rules that represent rough-graded preference relations. *Eur. J. Oper. Res.* 188 (1) (2008) 206–223.
- [37] S. Greco, B. Matarazzo, and R. Słowiński. Dominance-based rough set approach as a proper way of handling graduality in rough set theory. *Transactions on Rough Sets VII, Lecture Notes in Computer Science*, Vol. 4400. Springer-Verlag, Berlin, 2007, pp. 36–52.
- [38] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA, 1993.
- [39] D. Dubois, H. Prade, F. Esteva, P. Garcia, L. Godo, and R. Lopez de Mantara. Fuzzy set modelling in case-based reasoning. *Int. J. Intell. Syst.* 13 (1998) 345–373.
- [40] D.B. Leake. CBR in context: the present and future. In: D. Leake (ed.), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. AAAI Press/MIT Press, Menlo Park, 1996, pp. 1–30.
- [41] I. Gilboa and D. Schmeidler. *A Theory of Case-Based Decisions*. Cambridge University Press, Cambridge, 2001.
- [42] D. Hume. *An Enquiry Concerning Human Understanding*. Clarendon Press, Oxford, 1748.

16

A Unified Approach to Granulation of Knowledge and Granular Computing Based on Rough Mereology: A Survey

Lech Polkowski

16.1 Introduction

The topic of this chapter, an approach to granulation of knowledge, may be located in the vast area of approximate reasoning whose principal aim is to give descriptions of concepts from premises known to a degree only or approximately.

16.1.1 Sources of Motivation

Our approach stems from two paradigms of approximate reasoning, namely, the rough set theory and the fuzzy set theory. Our concept analysis is carried out in the framework of rough set theory, and our concepts are granules that are exact sets in respective information systems whereas our methodology which is based on the notion of a part to a degree does employ the feature of partial containment that is in analogy to the idea of a membership to a degree on which the fuzzy set theory is founded.

Granulation of information was posed as a problem within the fuzzy set theory by Lotfi Zadeh [1], and the idea was transferred into the realm of the rough set theory as well, see, e.g., Polkowski and Skowron [2] and T.Y. Lin [3].

Granulation of knowledge can be regarded as a form of prototype reasoning, see Duda *et al.* [4], as granules are built about selected objects – granule centers; regarded from this point of view, granulation is a form of reasoning by analogy or similarity and it does extend methods like nearest neighbors [4–6].

Rough set theory does address the problem of approximate description of concepts with the idea of a family of exact concepts with which any concept can be approximated. Exact concepts in turn are induced from knowledge represented in the form of an information system or, more generally, as an approximation space (for the latter notion, see [7, 8]).

Concepts are represented in the rough set theory as sets, more precisely, as subsets of the universe of objects, in the language of naive set theory. However, in order to introduce the notion of an approximation,

one requires the notion of containment, i.e., not any membership relation of naive set theory, but it does fall into the realm of part relations that form the foundation for mereological theories of concepts (and in these theories containment is, indeed, a membership relation).

The two languages for expressing the notion of a concept, namely, set-theoretical and mereological are related, as mentioned, by the fact that the containment relation \subset is a part relation; however, the same fact does witness that the two languages are of distinct types, mereology being a language of a higher type.

From the foundational point of view, the opposition between the naive set theory and the mereological theory of concepts is the opposition between distributive and collective views on the nature of entities.

This opposition is rendered formally in theories of concepts: ontology is concerned with distributive aspects of concepts whereas mereology is concerned with collective aspects of concepts.

In order to illustrate this opposition in less formal and more intuitive way, we turn to a poem. The reader may appreciate, we do hope, a quotation of a poem by Zbigniew Herbert, *The Path* (*Scieżka*, in Polish), cf., the original in [9], in this author's translation, which does address this problem of a choice of a language, as well as the methodological difference between the two possible strategies for concept description:

It was not any path of truth just simply a path/Reddish root branches across it/pine needles on sides/In the forest full of berries and spirits uncertain/It was not any path of truth: it lost suddenly its unity/From there our paths in life uncertain

On the right there was the spring/Approached with steps of darkness into deeper blackness/Guided by touch one came there to mother of elements worshipped by Tales/To meet watery heart of things, dark grain of cause

On the left there was the hill/It gave calmness and a general view/The forest border its dark body with no leaf trunk berry/Without knowledge it is one of many forests

Is it not really possible to have together the spring the hill the idea the leaf/And convey the multitude without dark alchemy too bright abstraction?

The poet gives metaphysical hue to the question whether one can compromise a global view on concepts expressed in the language of parts with a local one rooted in structures built upward from elements. In other, more formal terms, can one mediate between distributive and collective views on concepts? We hope that results in this chapter will also bring at least a partial positive answer to that question.

16.1.2 Principal Tools

Our aim is to define the notion of a granule in a flexible way so as to make it independent of a particular subset of the set of attributes; a standard approach to granulation of knowledge in rough set theory is to consider indiscernibility relation relative to a set of attributes and adopt its equivalence classes as atomic granules that generate the complete Boolean algebra of granules. See a discussion in T.Y. Lin [10], Y.Y. Yao [11], and Y.Y. Yao [12] concerned with generalizations of this approach to arbitrary binary relations.

The approach we describe in this chapter consists in defining granules as classes of some properties of objects expressed in terms of quasi-similarity relations.

Similarity relations as a substitute for more rigid indiscernibility relations were introduced into the rough set theory in Polkowski *et al.* [13] as *tolerance relations* (cf., Zeeman [14]), i.e., relations which are reflexive and symmetric but need not be transitive.

Since that time, granulation relative to a similarity relation has been discussed in the literature (see, e.g., Y.Y. Yao [15], Skowron–Stepaniuk [8], Qing Liu [16]).

However, the requirement of symmetry may be fulfilled by means of standard techniques; thus, there is no need to require symmetry (cf., Słowiński and Vanderpooten [17]). Moreover, the desire to introduce graded containment relations forces one to consider graded families of relations rather than single relations. These motivations prompt us to introduce a notion of a quasi similarity, which does encompass possible variations of similarity relations.

A *quasi-similarity* relation is defined by us as a hierarchy $\tau = \{\tau_r : r \in [0, 1]\}$ of relations such that:

1. $\tau_r(u, u)$ for each $r \in (0, 1)$.
 2. $\tau_r \subseteq \tau_s$ whenever $s \leq r$.
 3. τ_1 is a partial ordering.
- (1)

Quasi-similarity relations we consider in this chapter are more specialized and they are known by the generic term of *rough inclusions* after Polkowski and Skowron [2]; their definition and examples are discussed in Section 16.4.1. The important tool in our approach is the *class operator* which is defined in mereological theory of concepts and is introduced in Section 16.3.

Here, we point to its nature; given a non-vacuous property of objects, say Φ , the class of Φ , relative to a quasi similarity τ , denoted $Cl_{\tau}\Phi$, collects all objects u with the property that for each object v such that $v\tau_1 u$ there exist objects w, z with the properties that $w\tau_1 v, w\tau_1 z, \Phi(z)$.

In plain words, $Cl_{\tau}\Phi$ collects all objects with the property that each of their τ -subordinated objects has an object τ -subordinated to it that is at the same time τ -subordinated to an object with the property Φ ; this description may not seem to be transparent enough, and thus we advise the reader to verify that in the case where the partial ordering τ_1 is a containment \subseteq on a family F of sets, and the property Φ is the membership of a set in F , the class $Cl_{\tau}\Phi$ is the union $\bigcup F$.

Working in the way described, the class operator is making distributive entities into collective entities; hence, it is providing a passage from the ontological realm to the mereological realm.

In applications discussed in this chapter, it is possible to represent classes as sets or lists of objects.

In our approach, properties Φ are localized by defining for each object u in the universe of a considered information system, and each $r \in [0, 1]$, the property $\Phi(u, r) = \{v : v\tau_r u\}$; hence, our granules defined as concepts of the form $Cl_{\tau_r}\Phi(u, r)$ bear resemblance to topological neighborhoods, and we show in Section 16.4.1 that systems of granules defined by us on the lines outlined here retain properties of neighborhood systems in topological spaces.

16.1.3 Applications

Systems of granules allow, in turn, for many applications; as a principal one, we would like to single out the application in definitions of intensional granular logics that capture many essential logical aspects of reasoning by means of rough sets. In this approach, meanings of formulas of logic are defined as functions over possible worlds, and possible worlds are granules; the meaning of a formula at a possible world, i.e., a granule, is a real number between 0 and 1, giving the degree/state of truth of the formula with respect to the given granule.

Granular logics may in turn be applied in a formal analysis of problems of fusion of knowledge, many-agent systems, rough-neural computing, and perception calculus, as discussed in respective sections.

Finally, we consider granular structures induced by a given rough inclusion, in particular, granulated information/decision systems (cf. [18, 19]); by applying algorithms for inducing classifiers to those structures, more compact sets of rules are produced and yet the accuracy of classification is practically the same for large enough radii of granulation.

16.2 Basic Rough Set Theory

Rough set theory was conceived as a tool for reasoning under uncertainty by Zdzisław Pawlak around 1981–1982 (cf., [20]; see also [21]). Knowledge in this theory is understood as an ability to classify objects under discussion; the classification, in turn, can be expressed in its simplest form as a family of equivalence relations [20, 21]. Commonly, these relations are described in terms of certain features (attributes) and their values on objects as indiscernibility relations [20, 21] in the sense that objects on which all considered attributes take the same value should be regarded as indiscernible (the Leibniz identity of indiscernibles principle; see, e.g., [22]).

Information systems are chosen here as the framework for defining knowledge on the basis of indiscernibility relations.

16.2.1 Information Systems

An information system is a pair $I = (U, A)$, where U is a set of objects (the universe of I) and A is a set of attributes; we assume that both sets U and A are finite and non-empty.

Attributes are construed as mappings on objects; i.e., each attribute $a \in A$ is a mapping $a : U \rightarrow V_a$, where the set V_a is the set of values of the attribute a .

As already mentioned, objects $u, v \in U$ are *A-indiscernible* whenever $a(u) = a(v)$ for every $a \in A$. Formally, this fact is rendered by indiscernibility relations $IND(B)$ relativized to subsets $B \subseteq A$ of the attribute set A :

$$(u, v) \in IND(B) \text{ iff } a(u) = a(v) \text{ for each } a \in B.$$

The relation $IND(B)$ does partition the universe U into classes of the form

$$[u]_B = \{v \in U : (u, v) \in IND(B)\},$$

where $u \in U$. These classes form the *B-primitive granule* collection over the information system I .

Unions of primitive granules are called *elementary granules*. Let us observe that they form a complete Boolean algebra under standard set-theoretic operations of the union, the intersection and the complement.

Another terminology calls *B-elementary granules B-exact sets*; the reason for this name is best seen with the help of *description logic* (see [21, 23]).

Primitive formulas of that logic are *descriptors* of the form (a, v) with $a \in A, v \in V_a$. Formulas are constructed from descriptors by means of sentential connectives $\vee, \wedge, \neg, \Rightarrow$.

Semantics of formulas is defined as follows, where $[[p]]$ denotes the meaning of a formula p :

1. $[[a, v]] = \{u \in U : a(u) = v\}.$
 2. $[[p \vee q]] = [[p]] \cup [[q]].$
 3. $[[p \wedge q]] = [[p]] \cap [[q]].$
 4. $[[\neg p]] = U \setminus [[p]].$
- (2)

From equation (2), it follows that the *B-primitive granule* $[u]_B$ is the meaning of the formula $\phi_u : \bigwedge_{a \in B} (a, a(u))$, and each *B-elementary granule* $\bigcup_{j \in J} [u_j]_B$ is the meaning of the formula $\bigvee_{j \in J} \phi_{u_j}$.

In this sense elementary granules are exact: they serve as meanings of formulas in the descriptor logic.

A generalization of this notion of a granule is offered as the notion of a *template* [6]: a template is a formula of the form $(a \in W_a)$, where $W_a \subseteq V_a$ is a set of values of the attribute a . Semantics of templates is defined as with descriptors. Templates can be regarded as aggregates of elementary granules offering a compression of description and, in a sense, a further granulation of descriptor-based elementary granules.

Sets/concepts that are not *B-exact* are called *B-rough* (see [21]).

Let us finally observe that *B-exact sets* are characterized by the following dichotomy:

$$\text{For each } u \in U. [u]_B \subseteq X \vee [u]_B \subseteq U \setminus X. \quad (3)$$

A set $X \subseteq U$ is called *exact* iff it is *B-exact* for some set B of attributes; a set that is not exact is called *rough*.

Rough concepts are perceived in rough set theory by means of their approximations by the exact ones; formally, for each concept X , and a set B of attributes, two exact concepts $B\underline{X}$ and $B\overline{X}$ are defined as follows:

1. $B\underline{X} = \{u \in U : [u]_B \subseteq X\};$
 2. $B\overline{X} = \{u \in U : [u]_B \cap X \neq \emptyset\}.$
- (4)

Clearly, $B\underline{X}$ = l.u.b. of the family of exact sets contained in X ; dually, $B\overline{X}$ is the g.l.b. of the family of exact sets that contain X .

Duality between approximations is expressed by the identity

$$B\underline{X} = U \setminus B\overline{U \setminus X},$$

and its dual in which operators $\underline{\cdot}, \overline{\cdot}$ change places.

16.3 Mereology as a Theory of Concepts

Rough set theory has been formulated in terms of naive set theory, yet formulas such (3) point to the fact that essential notions of rough set theory are formulated in terms of containment rather than membership; it seems, indeed, sensible to assume that concepts may be compared and related one to another by containment.

16.3.1 Motivations

The relation \subset of containment is of type of relations that are irreflexive and transitive; such relations were called *part relations* and made basic for mereological theory proposed by Stanisław Leśniewski [24].

This theory is a theoretical basis for our approach in spite of the existence of other approaches to the notion of a part, notably the one based on a topological notion of connected objects and due to A.N. Whitehead [25], developed later by Leonard–Goodman [26] and Clarke [27], among others.

The reader is asked to assume that relevant theoretical proofs of facts stated here are carried out in mereological language; however, the results may also be obtained, with less elegance, in the language of set theory; this is, actually, a reason for other parallel theories of granulation.

Given a universe U of objects/entities, a *part relation* on U is a relation π that satisfies the following requirements:

1. $u\pi u$ for no $u \in U$.
 2. If $u\pi v$ and $v\pi w$ then $u\pi w$.
- (5)

It follows that any part relation is a strict partial ordering of the set of entities; it can be interpreted as an exact decomposition scheme of complex objects into proper parts.

The associated notion of an improper part is rendered [24] as the relation ing_π of an *ingredient*:

$$u \text{ } ing_\pi \text{ } v \text{ iff } u\pi v \text{ or } u = v. \quad (6)$$

The relation ing_π is a partial ordering on the set of entities giving exact decomposition of complex objects into parts (possibly, improper); thus,

1. $u \text{ } ing_\pi \text{ } u$.
 2. If $u \text{ } ing_\pi \text{ } v$ and $v \text{ } ing_\pi \text{ } u$ then $u = v$.
 3. If $u \text{ } ing_\pi \text{ } v$ and $v \text{ } ing_\pi \text{ } w$ then $u \text{ } ing_\pi \text{ } w$.
- (7)

We will use in our reasoning in terms of parts the following theorem due to Leśniewski [24]:

Theorem 1. Given u, v , if for every object w , from $w \text{ } ing_\pi \text{ } u$ it follows that there exists z such that $z \text{ } ing_\pi \text{ } w$ and $z \text{ } ing_\pi \text{ } v$ then $u \text{ } ing_\pi \text{ } v$.

We will refer to Theorem 1 as to the inference rule in what follows. This rule states that in order to infer that $u \text{ } ing_\pi \text{ } v$, it suffices to check that for any entity w , such that $w \text{ } ing_\pi \text{ } u$, one can find an entity z satisfying $z \text{ } ing_\pi \text{ } w$ and $z \text{ } ing_\pi \text{ } v$.

In mereological theory of Leśniewski, a distinction is made between the two types of entities: individual entities and distributive entities to which the former belong in a sense. This distinction is made formal and precise in ontology to which we do not refer here. The need for an operator that would convert distributive entities (collections) into individual entities is filled with the already mentioned in Section 16.1.2 class operator whose workings we describe now.

Assume that Φ is a non-empty collection/property of entities. The individual *class of Φ* , denoted $Cl\Phi$, is defined as the unique individual X that satisfies the following conditions:

1. If $\Phi(u)$, then $u \text{ } ing_\pi \text{ } X$.
 2. If $u \text{ } ing_\pi \text{ } X$ then there exist w, z ,
such that $w \text{ } ing_\pi \text{ } u$, $w \text{ } ing_\pi \text{ } z$, $\Phi(z)$.
- (8)

The conditions 1 and 2 in (8) state that ingredients of the class $\text{Cls}\Phi$ are those entities whose each ingredient has an ingredient in common with an entity in Φ . The reader may notice the analogy of the requirement 2 to the condition in inference rule (1).

Let us add that the setting in which the class operator has been introduced in Section 16.1.2 conforms to the mereological context when accepting the relation τ_1 as an ingredient relation induced by a part relation.

Example 1. As mentioned above, in Section 16.1 the relation of proper containment \subset is a part relation with the associated ingredient relation of improper containment \subseteq . For a non-empty collection Φ of sets, the class $\text{Cls}\Phi$ is the union $\bigcup\Phi$.

16.4 Rough Mereology

In the setting of information systems, the phenomenon of inexactness, calling for approximate description of concepts, has been taken into account by means of notions of exact, respectively, rough concepts (sets). Inexactness of concepts also bears on part relation, as for inexact (rough) concepts, the part relation cannot be ascertained uniquely in terms of descriptors.

An idea poses itself here, to consider relations (predicates) of *being a part to a degree*.

Predicates of this kind were introduced in Polkowski and Skowron [2] as *rough inclusions*, i.e., predicates of the form $\mu_\pi(x, y, r)$, where x, y are individuals, $r \in [0, 1]$, which satisfy the requirements relative to a given part relation π on a set U of entities;

1. $\mu_\pi(x, y, 1) \Leftrightarrow x \text{ ing}_\pi y$.
 2. $\mu_\pi(x, y, 1) \Rightarrow [\mu_\pi(z, x, r) \Rightarrow \mu_\pi(z, y, r)]$.
 3. $\mu_\pi(x, y, r) \wedge s < r \Rightarrow \mu_\pi(x, y, s)$.
- (9)

These requirements seem to be intuitively clear:

1. It is the requirement that the predicate μ_π is an extension to the relation ing_π of the underlying system of mereology; by this, the exact decomposition scheme of objects into parts, set by the ordering ing_π is a ‘skeleton’ in a sense, along which a decomposition into ‘real’ parts to a degree takes place.
2. It does express monotonicity of μ_π .
3. It assures the reading ‘to degree at least r ’.

Clearly, the family $\{\mu(x, y, r) : r \in [0, 1]\}$ is a quasi similarity in the sense of definition given in Section 16.1.2.

16.4.1 Rough Inclusions

We now describe some means of inducing rough inclusions in information systems; as decision systems, i.e., information systems of the form (U, A, d) with the attribute $d \notin A$ called the *decision*, see [21], can be regarded as a special case of information systems; we do not discuss the former separately.

By now, we are aware of two basic means of inducing rough inclusions (see [18, 19, 28, 29]).

Rough Inclusions from Archimedean t-Norms

The method we describe now is related to and based on properties of *Archimedean t-norms*. We recall that a *t-norm* $T : [0, 1]^2 \rightarrow [0, 1]$ (see, e.g., [30]), is a mapping such that

1. $T(x, y) = T(y, x)$.
2. $T(x, T(y, z)) = T(T(x, y), z)$.
3. T is increasing in each coordinate.
4. $T(1, x) = x$, $T(0, x) = 0$.

A t -norm T is *Archimedean* (see, e.g., [30]), when T is continuous and $T(x, x) < x$ for $x \in (0, 1)$.

It is well known that Archimedean t -norms admit a functional characterization, a very special case of the general Kolmogorov theorem [31], namely, for any Archimedean t -norm T , the following functional equation holds;

$$T(x, y) = g_T(f_T(x) + f_T(y)), \quad (10)$$

where the function $f_T : [0, 1] \rightarrow R$ is continuous decreasing with $f_T(1) = 0$, and $g_T : R \rightarrow [0, 1]$ is the pseudo-inverse to f_T , see Ling [32]. (A discussion may also be found in [33] or [34]).

As our knowledge is encoded by information systems, we consider an information system $I = (U, A)$, and we induce in its universe a rough inclusion μ_T , where T is an archimedean t -norm that satisfies the representation (10). We assume that *indiscernibility* = *identity*; i.e., each indiscernibility class is represented by a unique object.

In order to define μ_T , we consider objects $u, v \in U$ and define the set

$$DIS(u, v) = \{a \in A : a(u) \neq a(v)\}. \quad (11)$$

We let

$$\mu_T(u, v, r) \Leftrightarrow g\left(\frac{|DIS(u, v)|}{|A|}\right) \geq r. \quad (12)$$

Then, condition 1 in (9) is satisfied when $g(0) = 1$ with *ing* = *indiscernibility*. Under these conditions 2 and 3 in (9) are also satisfied. We assume from now on that we consider rough inclusions under these conditions. Rough inclusions defined by means of (12) will be called *Archimedean rough inclusions*.

Example 2. As an example, we consider the Lukasiewicz t -norm, see, e.g., [30],

$$L(x, y) = \max\{0, x + y - 1\}, \quad (13)$$

in which case, $f_L(x) = 1 - x$ and $g_L(y) = 1 - y$ when $y \leq 1$, and $g_L(y) = 0$ for $y > 1$ (see [32]).

The rough inclusion μ_L induced by L is of the form

$$\mu_L(u, v, r) \Leftrightarrow 1 - \frac{|DIS(u, v)|}{|A|} \geq r. \quad (14)$$

Introducing the set $IND(u, v) = A \setminus DIS(u, v)$, we obtain,

$$\mu_L(u, v, r) \Leftrightarrow \frac{|IND(u, v)|}{|A|} \geq r. \quad (15)$$

The formula (15) witnesses that the reasoning based on the rough inclusion μ_L is the probabilistic one. At the same time, we have given a logical proof for formulas such as (15) that are very frequently applied in data mining and knowledge discovery, also in rough set methods in those areas (see, e.g., [35]).

The other example of an Archimedean t -norm is supplied by the *product* t -norm $P(x, y) = x \cdot y$; in this case, $g_P(y) = \exp(-y)$; hence, the recipe (12) yields

$$\mu_P(u, v, r) \Leftrightarrow \exp\left(-\frac{|DIS(u, v)|}{|A|}\right) \geq r.$$

In general, see, e.g. [30], or [33], any Archimedean t -norm is isomorphic either to L or to P ; hence, μ_L, μ_P are the only two rough inclusions obtainable in this way.

For an Archimedean t -norm T -induced rough inclusion μ_T , the transitivity property holds in the form, see [2, 28].

Theorem 2.

$$\text{if } \mu_T(x, y, r) \text{ and } \mu_T(y, z, s) \text{ then } \mu_T(x, z, T(r, s)). \quad (16)$$

An argument will be useful here for the sake of completeness; we begin with the observation that

$$DIS(x, z) \subseteq DIS(x, y) \cup DIS(y, z); \quad (17)$$

hence,

$$\frac{|DIS(x, z)|}{|A|} \leq \frac{|DIS(x, y)|}{|A|} + \frac{|DIS(y, z)|}{|A|}. \quad (18)$$

$$\text{Let } g_T\left(\frac{|DIS(x, y)|}{|A|}\right) = r, g_T\left(\frac{|DIS(y, z)|}{|A|}\right) = s, g_T\left(\frac{|DIS(x, z)|}{|A|}\right) = u.$$

Then,

$$\begin{aligned} \frac{|DIS(x, y)|}{|A|} &= f_T(r); \\ \frac{|DIS(y, z)|}{|A|} &= f_T(s); \\ \frac{|DIS(x, z)|}{|A|} &= f_T(u). \end{aligned} \quad (19)$$

Finally, by (18),

$$f_T(u) \leq f_T(r) + f_T(s), \quad (20)$$

hence,

$$u = g_T(f(u)) \geq g_T(f_T(r) + f_T(s)) = T(r, s), \quad (21)$$

witness to $\mu_T(x, z, T(r, s))$.

Example 3. We restrict ourselves to the Lukasiewicz rough inclusion μ_L and we consider Table 16.1 that consists of selected six rows from the well-known test set Monk1 (see, e.g., [36]).

For this table, we collect in Table 16.2 values of the rough inclusion μ_L , given in triangular form due to the symmetry. The values are computed from the attribute set $A = \{a_1, \dots, a_6\}$ as the last attribute d is the decision.

Table 16.1 Test table: a selection from Monk1

U	a_1	a_2	a_3	a_4	a_5	a_6	d
0 : 1	1	1	1	1	1	1	1
0 : 5	1	1	2	1	2	1	1
0 : 8	1	1	2	2	4	1	1
0 : 49	2	1	1	2	1	2	1
0 : 58	2	1	2	2	3	1	0
0 : 73	2	2	2	1	4	1	1

Table 16.2 The Łukasiewicz rough inclusion for Table 16.1

U	$0 : 1$	$0 : 2$	$0 : 3$	$0 : 4$	$0 : 5$	$0 : 6$
$0 : 1$	1.0	*	*	*	*	*
$0 : 5$	0.66	1.0	*	*	*	*
$0 : 8$	0.5	0.66	1.0	*	*	*
$0 : 49$	0.33	0.16	0.33	1.0	*	*
$0 : 58$	0.5	0.5	0.66	0.5	1.0	*
$0 : 73$	0.33	0.5	0.5	0.16	0.5	1.0

16.4.2 Rough Inclusions from Continuous t -Norms

A more general case is offered by continuous t -norms. It results from the result of Mostert–Shields and Faucett (see [37, 38], see also [30, 33]), that the structure of a continuous t -norm T depends on the set $F(T)$ of idempotents of T , i.e., values of x such that $T(x, x) = x$; we denote with O_T the countable family of open intervals $A_i \subseteq [0, 1]$ with the property that $\bigcup_i A_i = [0, 1] \setminus F(T)$. Then, $T(x, y)$ is an isomorph to either $T_L(x, y)$ or $T_P(x, y)$ when $x, y \in A_i$ for some i , and $T(x, y) = \min\{x, y\}$ otherwise.

It is in principle possible to define a rough inclusion on the lines of the preceding section; however, the t -norm \min admits no regular representation like (10) (a result due to Arnold and Kirillov, quoted in [32]); hence, a simple formula would be hard to produce in this case.

Thus, in this case we propose an other way and we resort to residua of continuous t -norms.

For a continuous t -norm $T(x, y)$, the *residuum*, see, e.g., [30], $x \Rightarrow_T y$ is defined as $\max\{z : T(x, z) \leq y\}$; thus, the equivalence holds

$$x \Rightarrow_T y \geq z \text{ iff } T(x, z) \leq y. \quad (22)$$

Clearly, for each T , $x \Rightarrow_T y = 1$ if and only if $x \leq y$.

For an information system (U, A) , let us select an object $s \in U$ referred to as a *standard*. From application point of view, s may be the best classified case, etc.

For $x \in U$, we let,

$$IND(x, s) = \{a \in A : a(x) = a(s)\}. \quad (23)$$

For a continuous t -norm T , we define a rough inclusion $v_T^{IND, s}$ by letting

$$v_T^{IND, s}(x, y, r) \text{ iff } \frac{|IND(x, s)|}{|A|} \Rightarrow \frac{|IND(y, s)|}{|A|} \geq r. \quad (24)$$

Let us examine the three basic t -norms.

In case of the Lukasiewicz t -norm L , we have

$$x \Rightarrow_L y = \min\{1, 1 - x + y\}; \quad (25)$$

thus,

$$v_L^{IND, s}(x, y, r) \text{ iff } |IND(y, s)| - |IND(x, s)| \geq (1 - r)|A|. \quad (26)$$

In case of the product t -norm P , we have

$$x \Rightarrow_P y = \begin{cases} 1 & \text{when } x \leq y \\ y & \text{when } x > y. \end{cases} \quad (27)$$

Hence,

$$v_P^{IND,s}(x, y, 1) \text{ iff } |IND(x, s)| \leq |IND(y, s)|, \quad (28)$$

and

$$v_P^{IND,s}(x, y, r) \text{ iff } |IND(x, s)| > |IND(y, s)| \geq r \cdot |A|. \quad (29)$$

Finally, in case of $T = \min$, we have

$$x \Rightarrow_{\min} y = \begin{cases} 1 & \text{in case } x \leq y \\ \frac{y}{x} & \text{otherwise.} \end{cases} \quad (30)$$

Thus,

$$v_{\min}^{IND,s}(x, y, r) \text{ iff } \frac{|IND(y, s)|}{|IND(x, s)|} \geq r. \quad (31)$$

Rough inclusions defined according to the recipe (24) will be called *residual*.

They also satisfy the transitivity property.

Theorem 3. For any continuous t-norm T , from $v_T^{IND,s}(x, y, r)$ and $v_T^{IND,s}(y, z, s)$ it follows that $v_T^{IND,s}(x, z, T(r, s))$.

We offer a simple argument to this theorem. We let, to be concise, $i(x, y, r)$ for $v_T^{IND,s}(x, y, r)$, and so on, and we assume that $i(x, y, r), i(y, z, s)$ hold. Thus, we have

$$T(x, r) \leq y, \quad (32)$$

and

$$T(y, s) \leq z. \quad (33)$$

By Property 2 in Section 16.4.1, it follows from (32) and (33) that

$$T(T(x, r), s) = T(x, T(r, s)) \leq z; \quad (34)$$

thus, finally,

$$x \Rightarrow_T z \geq T(r, s); \quad (35)$$

i.e.,

$$i(x, z, T(r, s)). \quad (36)$$

Let us observe, that one may define ‘dual’ similarity measures $v_T^{DIS,s}$ by replacing in the above formulas sets $IND(x, s)$ with sets $DIS(x, s) = \{a \in A : a(x) \neq a(s)\}$. These measures seem especially suited to the case of ‘standard’-based reasoning in multi-agent systems discussed in the further part of this chapter.

16.4.3 Rough Inclusions as Fuzzy Similarity Relations

We include here a result that states (see, e.g., [28, 33]) that any rough inclusion $\mu_\pi(x, y, r)$ does induce on its universe a fuzzy similarity relation in the sense of Zadeh [39].

We include this discussion here, as its results point to the fact that the calculus we develop is also a fuzzy-style calculus in information systems, and granular computing based on rough inclusions may be regarded as a rough-fuzzy calculus.

First, writing $\mu_y^\pi(x) = r$ instead of $\mu_\pi(x, y, r)$, we convert the relational notation into the fuzzy-style one.

We observe that fuzzy sets of the form μ_y^π are higher level sets: values of fuzzy membership degrees here are convex subintervals of the unit interval $[0, 1]$ of the form $[0, r]$, i.e., with the left-end point 0; hence, the formula $\mu_y^\pi(x) = t$ is understood as the statement that the subinterval $\mu_y^\pi(x)$ contains t .

Under this proviso, the *fuzzy tolerance relation* $\tau_y^\pi(x)$ is defined by means of

$$\tau_y^\pi(x) = r \Leftrightarrow \mu_y^\pi(x) = r \text{ and } \mu_x^\pi(y) = r, \quad (37)$$

and it does satisfy, clearly,

1. $\tau_x^\pi(x) = 1$;
 2. $\tau_x^\pi(y) = \tau_y^\pi(x)$.
- (38)

We will now use the notation $\tau_r(x, y)$ for $\tau_x(y) = r$, disregarding the part relation π .

Following Zadeh [39], we define *similarity classes* $[x]_\tau$ as fuzzy sets satisfying the condition,

$$\chi_{[x]_\tau}(y) = r \Leftrightarrow \tau_r(x, y), \quad (39)$$

and in this interpretation, τ becomes a fuzzy equivalence in the sense of Zadeh [39] (see [28]); i.e., the family $\{[x]_\tau : x \subseteq U\}$ does satisfy the requirements for a *T-fuzzy partition* [39], namely,

$$\forall x \exists y. \chi_{[x]_\tau}(y) = 1, \quad (40)$$

$$\begin{aligned} [x]_\tau \neq [z]_\tau \Rightarrow \\ \max_y \{\min\{\chi_{[x]_\tau}(y), \chi_{[z]_\tau}(y)\}\} < 1, \end{aligned} \quad (41)$$

$$\bigcup_x [x]_\tau \times_T [x]_\tau = \tau, \quad (42)$$

where $A \times_T B$ denotes the fuzzy set defined via

$$\chi_{A \times_T B}(u, v) = T(\chi_A(u), \chi_B(v)) \quad (43)$$

(see [28]).

The foundational issues clarified, we may pass to granular calculi and applications.

16.5 Granulation Based on Rough Inclusions

We are going to define granules of knowledge in information systems; our framework is that of rough mereology and we base our approach on rough inclusions. Accordingly, let us assume that an information system $I = (U, A)$ along with a rough inclusion μ on the set U is given.

16.5.1 The Notion of a Granule

We define a granule $g_\mu(u, r)$ about $u \in U$ of the radius r , relative to the rough inclusion μ , as follows:

$$g_\mu(u, r) \text{ is } Cls_\mu \Pi^\mu(u, r), \quad (44)$$

where $\Pi^\mu(u, r)$ is the property defined by means of

$$\Pi^\mu(u, r)(v) \text{ iff } \mu(v, u, r). \quad (45)$$

We adopt the set representation for classes and properties. For Table 16.1, for instance, $g_{0.5}(0 : 1) = \{0 : 1, 0 : 5, 0 : 8, 0 : 58\}$.

General properties of granules are collected below;

1. If $y \text{ ing } x$, then $y \text{ ing } g_r x$;
 2. If $y \text{ ing } g_r x$ and $z \text{ ing } y$, then $z \text{ ing } g_r x$;
 3. If $\mu(y, x, r)$, then $y \text{ ing } g_r x$;
 4. If $s < r$, then $g_r x \text{ ing } g_s x$,
- (46)

which follow from properties in (9) and the fact that *ing* is a partial order – in particular it is transitive.

16.5.2 The Case of Archimedean as well as Residual Rough Inclusions

In case of a rough inclusion μ_T induced by either Archimedean or a continuous *t*-norm *T*, one may give a better description of granule behavior, as the transitivity property holds for respective rough inclusions, namely,

Theorem 4. *For any granule $g_r x$ induced by an Archimedean or a continuous *t*-norm *T*,*

$$y \text{ ing } g_r x \text{ iff } \mu_T(y, x, r) \quad (47)$$

holds.

One way implication is 3 in (46); the reverse implication holds by the class definition (8): $y \text{ ing } g_r x$ implies u, v exist such that $u \text{ ing } y$, $u \text{ ing } v$, $\mu_T(v, x, r)$.

Thus, $\mu_T(u, y, 1)$, $\mu_T(u, v, 1)$, and $\mu_T(v, x, r)$ hold and transitivity (16) or (3), and symmetry properties of μ_T imply $\mu_T(y, x, r)$.

The overlapping relation *Ov* is defined for individual objects as follows,

$$Ov(x, y) \text{ if and only if } z \text{ ing } x \text{ and } z \text{ ing } y \text{ for some } z, \quad (48)$$

and then

$$\text{if } Ov(g_r x, g_r y), \text{ then } \mu_T(x, y, T(r, r)). \quad (49)$$

More generally,

$$\text{if } y \text{ ing } g_r x \text{ then } g_s y \text{ ing } g_{T(r,s)} x. \quad (50)$$

The last statement follows directly from the transitivity property of discussed rough inclusions and from (47).

16.5.3 Rough Inclusions on Granules

Regarding granules as objects calls for a procedure for evaluating rough inclusion degrees among granules. This problem shows advantages of mereological apparatus: due to lack of hierarchy of individuals, all entities are simply individuals; i.e., a granule of granules will automatically be a granule. First, we have to define the notion of an ingredient among granules. On the basis of the inference rule, Theorem 1, for granules *g, h*, we let

$$\begin{aligned} g \text{ ingr } h \text{ if and only if } z \text{ ingr } g \text{ implies that there is } t \text{ such that} \\ z \text{ ingr } t, t \text{ ingr } h, \end{aligned} \quad (51)$$

and, more generally, for granules g, h , and a rough inclusion μ ,

$$\begin{aligned} \mu(g, h, r) \text{ if and only if for } z \text{ ingr } g \text{ there is } w \text{ such that} \\ \mu_t(z, w, r), w \text{ ingr } h. \end{aligned} \quad (52)$$

Then μ is a rough inclusion on granules [28].

This procedure may be iterated to granules of granules giving always a granule as the result due to peculiar properties of the class operator.

It is natural to regard granule system $\{g_r^{\mu_t}(x) : x \in U; r \in (0, 1)\}$ as a neighborhood system for a topology on U that may be called the *granular topology*; the idea of a granule as a neighborhood appeared also in [3].

In order to make this idea explicit, we define classes of the form $N^T(x, r) = \text{Cls}(\psi_{r,x}^{\mu_T})$, where

$$\psi_{r,x}^{\mu_T}(y) \Leftrightarrow \exists s > r \cdot \mu_T(y, x, s). \quad (53)$$

We declare the system $\{N^T(x, r) : x \in U; r \in (0, 1)\}$ to be a neighborhood basis for a topology θ_μ . This is justified by the following:

Theorem 5. *Here are properties of the system $\{N^T(x, r) : x \in U; r \in (0, 1)\}$:*

1. $y \text{ ingr } N^t(x, r) \Rightarrow \exists \delta > 0. N^t(y, \delta) \text{ ingr } N^t(x, r).$
2. $s > r \Rightarrow N^t(x, s) \text{ ingr } N^t(x, r).$
3. $z \text{ ingr } N^t(x, r) \wedge z \text{ ingr } N^t(y, s) \Rightarrow \exists \delta > 0.$
 $N^t(z, \delta) \text{ ingr } N^t(x, r) \wedge N^t(z, \delta) \text{ ingr } N^t(y, s).$

An argument for (54) is as follows.

For Property 1 $y \text{ ingr } N^t(x, r)$ implies, by (53) and 2 in (9), that there exists an $s > r$ such that $\mu_t(y, x, s)$. Let $\delta < 1$ be such that $t(u, s) > r$ whenever $u > \delta$; δ exists by continuity of t and the identity $t(1, s) = s$. Thus, if $z \text{ ingr } N^t(y, \delta)$, then $\mu_t(z, y, \eta)$ with $\eta > \delta$, and by (53), $\mu_t(z, x, t(\eta, s))$; hence, $z \text{ ingr } N^t(x, r)$.

Property 2 follows by Proposition 1 and Property 3 is a consequence to Properties 1 and 2. This concludes the argument for (54).

Granule systems defined above form a basis for applications where approximate reasoning is a crucial ingredient.

We begin with a basic application in which approximate reasoning itself is codified as a many-world (intensional) logic where granules serve as possible worlds.

16.6 Granular Logics

Rough inclusions may be adopted to the task of defining logics reflecting the reasoning mode based on rough mereology and thus related directly to the ideology of rough set theory. Our logics are intensional logics (for a general discussion of intensional logics see, e.g., vanBenthem [40], or Montague [41]). This approach is different from earlier approaches to logical content of the rough set theory (see, e.g., Orłowska [42], Pawlak–Orłowska [43], Rasiowa–Skowron [44, 45], or Vakarelov [46]).

16.6.1 Rough Inclusions on Sets

We will need measures of partial containment on finite sets, and to this end, we follow the path proposed for rough inclusions on entities in information systems, with obvious modifications. We restrict ourselves here to the case of Archimedean t -norms, as our discussion is of an illustrative character; clearly, the rough inclusions based on residua of t -norms may be introduced here as well. Assume that T is an Archimedean t -norm, with the representation of the form (10).

For subsets $X, Y \subseteq U$, we let

$$\mu_T(X, Y, r) \text{ iff } g\left(\frac{|X \setminus Y|}{|X|}\right) \geq r. \quad (55)$$

In case of the Lukasiewicz t -norm L , the formula (55) comes down to the form of

$$\mu_L(X, Y, r) \text{ iff } \frac{|X \cap Y|}{|X|} \geq r. \quad (56)$$

Thus, again, we obtain a probabilistic formula that is basic for prevalent part of theory of soft computing. It may also be compared to the formulas for rough membership functions in Pawlak and Skowron [47], which constitute an early attempt at introducing graded containment into rough set theory.

16.6.2 Rough Mereological Granular Logics

We define an intensional logic [28, 48], whose intension is the mapping $I : \mathcal{E} \times \mathcal{F} \longrightarrow [0, 1]$, where \mathcal{F} is the set of meaningful formulas over a set $Pred$ of unary predicates interpreted in the set U .

We denote with $[[p(x)]]$ the meaning of a unary predicate $p(x) \in Pred$; i.e., $[[p(x)]] = \{u \in U : p(u)\}$.

Given a granule g , we denote with $I_g(\phi)$ the extension of the intension I at the set g , valued at a formula ϕ ; i.e., $I_g(\phi) = I(g, \phi)$.

We adopt the following interpretation of logical connectives N of negation and C of implication;

$$[[Np]] = U \setminus [[p]], \quad (57)$$

and

$$[[Cpq]] = (U \setminus [[p]]) \cup [[q]]. \quad (58)$$

For a rough inclusion μ on $(2^U)^2 \times [0, 1]$, where 2^U is the powerset of U , we define the value $(I_g^\mu)(\phi)$ of the extension of I relative to μ at g, ϕ as follows:

$$(I_g^\mu)(\phi) \geq r \Leftrightarrow \mu(g, [[\phi]], r). \quad (59)$$

We denote the rough mereological logic based on a rough inclusion μ with the symbol RML^μ .

We call a meaningful formula ϕ of RML^μ a *theorem with respect to μ* if and only if $(I_g^\mu)(\phi) = 1$ for each granule $g = g_r x$ with respect to μ .

In what follows, by $(I_g^{\mu_{T_L}})(p(x))$ we understand the maximal value equal to $\frac{|g \cap [[p(x)]]|}{|g|}$.

The extension $(I_g^{\mu_{T_L}})(\phi)$ does satisfy the following with respect to negation and implication:

$$(I_g^{\mu_{T_L}})(Np(x)) = 1 - (I_g^{\mu_{T_L}})(p(x)), \quad (60)$$

and

$$\begin{aligned} (I_g^{\mu_{T_L}})(Cp(x)q(x)) &= \frac{|g \cap (U \setminus [[p(x)]]) \cup [[q(x)]]|}{|g|} \\ &\leq 1 - (I_g^{\mu_{T_L}})(p(x)) + (I_g^{\mu_{T_L}})(q(x)), \end{aligned} \quad (61)$$

so finally,

$$(I_g^{\mu_{T_L}})(Cp(x)q(x)) \leq 1 - (I_g^{\mu_{T_L}})(p(x)) + (I_g^{\mu_{T_L}})(q(x)). \quad (62)$$

The formula on the right-hand side of inequality (62) is of course the Lukasiewicz implication of many-valued logic [49, 50].

We may say that in this case the logic RML^{T_L} is a sub-Łukasiewicz many-valued logic, meaning, in particular, that if $\phi(x)$ is a theorem of the logic RML^{T_L} , then a sentential form of the formula $\phi(x)$ is a theorem of $[0, 1]$ -valued Łukasiewicz logic.

We explain the last statement in more detail; for a formula $p(x)$, we denote with the symbol p^\dagger the formula p regarded as a formula of sentential logic, subject to $(\neg p)^\dagger$ is $\neg(p^\dagger)$ and $Cp(x)q(x))^\dagger$ is $C(p^\dagger)(q^\dagger)$. Then, as one may check, see [28].

Theorem 6. *If a formula $\phi(x)$ is a theorem of RML^μ , then ϕ^\dagger is a theorem of the Lukasiewicz sentential $[0, 1]$ -valued logic for each regular rough inclusion μ , where μ is regular iff $\mu(X, Y, I)$ is equivalent to $X \subseteq Y$.*

One verifies directly [28] that derivation rules:

$$(MP) \frac{p(x), Cp(x)q(x)}{q(x)} \text{ (modus ponens)}$$

and

$$(MT) \frac{\neg q(x), Cp(x)q(x)}{\neg p(x)} \text{ (modus tollens)}$$

are valid in the logic RML^μ for each regular rough inclusion μ .

In the context of intensional logic RML , we may discuss modalities L (of necessity) and M (of possibility). To introduce these modalities into RML , we make use of rough approximations in RZF , \underline{X} , respectively, \overline{X} , the lower, respectively, the upper, approximation to X .

We define, with the help of a regular rough inclusion μ , functors L of necessity and M of possibility (the formula $L\phi$ is read ‘it is necessary that ϕ ’ and the formula $M\phi$ is read ‘it is possible that ϕ ’) with partial states of truth as follows:

$$(I_\Lambda^\mu)(Lp(x)) = r \Leftrightarrow \mu(\Lambda, \underline{[[p(x)]]}, r), \quad (63)$$

and, similarly,

$$(I_\Lambda^\mu)(Mp(x)) = r \Leftrightarrow \mu(\Lambda, \overline{[[p(x)]]}, r). \quad (64)$$

It seems especially interesting to look at operators L, M with respect to the rough inclusion μ_{T_L} of Lukasiewicz. Then,

$$(I_g^{\mu_{T_L}})(Lp(x)) = \frac{|g \cap \underline{[[p(x)]]}|}{|g|}, \quad (65)$$

and,

$$(I_g^{\mu_{T_L}})(Mp(x)) = \frac{|g \cap \overline{[[p(x)]]}|}{|g|}. \quad (66)$$

It follows that

Theorem 7. *In the logic RML^{T_L} , a meaningful formula $\phi(x)$ is satisfied necessarily (i.e., it is necessary to the degree 1) with respect to the granule g iff $g \subseteq \underline{[[\phi(x)]]}$; similarly, $\phi(x)$ is possible (i.e., possible to the degree 1) with respect to the granule g iff $g \subseteq \overline{[[\phi(x)]]}$.*

Clearly, by the duality property of rough set approximations, the crucial relation

$$(I_g^\mu)(Lp(x)) = 1 - (I_g^\mu)(MNP(x)) \quad (67)$$

holds between the two modalities with respect to each rough inclusion μ .

A brief examination shows that the rough set interpretation of necessity presented here (and of possibility as well) differs from the interpretation proposed by fuzzy set theory [51, 52], i.e.,

$$(I_\Lambda^\mu)(L(p(x) \wedge q(x))) \leq \min\{(I_\Lambda^\mu)(Lp(x)), (I_\Lambda^\mu)(Lq(x))\}, \quad (68)$$

and the equality postulated by the fuzzy set theory [51, 52] does not hold in general.

Example 4. We may now present within our intensional logic RML^{T_L} an otherwise known fact, obtained and discussed by different techniques, e.g., in [42, 46], that rough sets support modal logic S5.

Proposition 1. The following formulas of modal logic are theorems of RML^μ with respect to every regular rough inclusion μ :

1. (K) $CL(Cp(x)q(x))CLp(x)Lq(x)$.
2. (T) $CLp(x)p(x)$.
3. (S4) $CLp(x)LLp(x)$.
4. (S5) $CMp(x)LMp(x)$.

Indeed, let us verify that the formula (K) is a theorem of RML^\otimes . Other formulas are theorems by virtue of duality. We have

$$\begin{aligned} [[CL(Cpq)CLpLq]] = \\ U \setminus \underline{U \setminus [[p]]} \cup \underline{[[q]]} \cup (U \setminus \underline{[[p]]}) \cup \underline{[[q]]}. \end{aligned} \quad (69)$$

Assuming that $u \in U$ is such that $u \notin U \setminus \underline{[[p]]} \cup \underline{[[q]]}$, we have that

$$(i) [u] \subseteq [[p]]; (ii) [u] \cap [[q]] = \emptyset, \quad (70)$$

where $[u]$ denotes the equivalence class of u with respect to the indiscernibility relation $IND(A)$.

It follows from (70) that $x \notin (U \setminus \underline{[[p]]}) \cup \underline{[[q]]}$ (as $x \in (U \setminus \underline{[[p]]}) \cup \underline{[[q]]}$ would mean that $[x] \subseteq (U \setminus \underline{[[p]]}) \cup \underline{[[q]]}$; hence, by (70)(ii) one would have $[x] \subseteq U \setminus \underline{[[p]]}$, contradicting (70)(i)); i.e., the meaning of (K) is U . That concludes the proof.

An interesting variation on the topic of rough mereological intensional logic is a 3-valued logic (see [53]).

The construction of that logic is carried out on the lines sketched above: the difference is in the selection of the underlying rough inclusion on sets, which, in this case, is defined as follows:

$$\mu_3(X, Y, r) \text{ iff } \begin{cases} r = 1 & \text{and } X \subseteq Y; \\ r \leq 0 & \text{and } X \cap Y = \emptyset; \\ r \leq \frac{1}{2} & \text{otherwise} \end{cases} \quad (71)$$

This logic has interesting applications to decision rules (see [53]).

16.7 Networks of Granular Agents

We now begin with the first principal applications of the so far developed apparatus. Rough inclusions and granular intensional logics based on them can be applied in describing workings of a collection of intelligent agents which are called here granular agents as they are endowed with granular logics.

A granular agent ag in its simplest form is a tuple

$$ag^* = (U_{ag}, A_{ag}, \mu_{ag}, Pred_{ag}, UncProp_{ag}, GSynt_{ag}, LSynt_{ag}),$$

where

1. $(U_{ag}, A_{ag}) = I_{ag}$ is an information system of the agent ag .
2. μ_{ag} is a rough inclusion induced from I_{ag} .
3. $Pred_{ag}$ is a set of first-order predicates interpreted in U_{ag} .
4. $UncProp_{ag}$ is the function that describes how uncertainty measured by rough inclusions at agents connected to ag propagates to ag .

5. The operator $GSynt_{ag}$, the granular synthesizer at ag , takes granules sent to the agent from agents connected to it and makes those granules into a granule at ag .
6. $LSynt_{ag}$, the logic synthesizer at ag , takes formulas sent to the agent ag by its connecting neighbors and makes them into a formula describing objects at ag .

A network of granular agents is a directed acyclic graph $N = (Ag, C)$, where Ag is its set of vertices, i.e., granular agents, and C is the set of edges, i.e., connections among agents, along with disjoint subsets $In, Out \subset Ag$ of, respectively, input and output agents.

We assume for simplicity that N consists of three agents connected into the tree, and we show a simple analysis of the direct fusion of knowledge; clearly, more complex schemes will require a deeper and more complex analysis but on the lines indicated in the example that follows.

16.7.1 Fusion of Knowledge: An Example

We consider an agent $ag \in Ag$ and – for simplicity reasons – we assume that ag has two incoming connections from agents ag_1 and ag_2 ; the number of outgoing connections is of no importance as ag sends along each of them the same information.

We assume that each agent is applying the rough inclusion μ_L induced by the Lukasiewicz t -norm L (see Section 16.4.1); also, each agent is applying the rough inclusion on sets of the form (56) in evaluations related to extensions of formulae intensions.

Clearly, there exists a fusion operator o_{ag} that assembles from objects $x \in U_{ag_1}, y \in U_{ag_2}$ the object $o(x, y) \in U_{ag}$; we assume that $o_{ag} = id_{ag_1} \times id_{ag_2}$, i.e., $o_{ag}(x, y) = (x, y)$.

Similarly, we assume that the set of attributes at ag equals $A_{ag} = A_{ag_1} \times A_{ag_2}$; i.e., attributes in A_{ag} are pairs (a_1, a_2) with $a_i \in A_{ag_i}$ ($i = 1, 2$) and that the value of this attribute is defined as

$$(a_1, a_2)(x, y) = (a_1(x), a_2(y)).$$

It follows that the condition holds

$$o_{ag}(x, y)IND_{ag}o_{ag}(x', y') \text{ iff } xIND_{ag_1}x' \text{ and } yIND_{ag_2}y'.$$

Concerning the function $UncProp_{ag}$, we consider objects x, x', y, y' ; clearly,

$$\begin{aligned} DIS_{ag}(o_{ag}(x, y), o_{ag}(x', y')) \\ \subseteq DIS_{ag_1}(x, x') \times A_{ag_2} \cup \\ A_{ag_1} \times DIS_{ag_2}(y, y'), \end{aligned} \tag{72}$$

and hence,

$$\begin{aligned} |DIS_{ag}(o_{ag}(x, y), o_{ag}(x', y'))| \\ \leq |DIS_{ag_1}(x, x')| \cdot |A_{ag_2}| \\ + |A_{ag_1}| \cdot |DIS_{ag_2}(y, y')|. \end{aligned} \tag{73}$$

By (73),

$$\begin{aligned} & \mu_{ag}(o_{ag}(x, y), o_{ag}(x', y'), t) \\ &= 1 - \frac{|DIS_{ag}(o_{ag}(x, y), o_{ag}(x', y'))|}{|A_{ag_1}| \cdot |A_{ag_2}|} \\ &\geq 1 - \frac{|DIS_{ag_1}(x, x')| \cdot |A_{ag_2}| + |A_{ag_1}| \cdot |DIS_{ag_2}(y, y')|}{|A_{ag_1}| \cdot |A_{ag_2}|} \\ &= 1 - \frac{|DIS_{ag_1}(x, x')|}{|A_{ag_1}|} + 1 - \frac{|DIS_{ag_2}(y, y')|}{|A_{ag_2}|} - 1. \end{aligned} \tag{74}$$

It follows that

$$\text{if } \mu_{ag_1}(x, x', r), \mu_{ag_2}(y, y', s) \text{ then} \\ \mu_{ag}(o_{ag}(x, y), o_{ag}(x', y'), L(r, s)). \quad (75)$$

Hence, $UncProp(r, s) = L(r, s)$, the value of the Łukasiewicz t -norm L on the pair (r, s) .

In consequence, the granule synthesizer $GSynt_{ag}$ can be defined in our example as

$$GSynt_{ag}(g_{ag_1}(x, r), g_{ag_2}(y, s)) \\ = (g_{ag}(o_{ag}(x, y), L(r, s))). \quad (76)$$

The definition of logic synthesizer $LSynt_{ag}$ follows directly from our assumptions,

$$LSynt_{ag}(\phi_1, \phi_2) = \phi_1 \wedge \phi_2. \quad (77)$$

Finally, we consider extensions of our logical operators of intensional logic.

We have for the extension $I(\mu_{ag})_{GSynt_{ag}(g_1, g_2)}^\vee(LSynt_{ag}(\phi_1, \phi_2))$:

$$I(\mu_{ag})_{GSynt_{ag}(g_1, g_2)}^\vee(LSynt_{ag}(\phi_1, \phi_2)) \\ = I(\mu_{ag_1})_{g_1}^\vee(\phi_1) \cdot I(\mu_{ag_2})_{g_2}^\vee(\phi_2), \quad (78)$$

which follows directly from (76) and (77).

Let us note that

$$I(\mu_{ag_1})_{g_1}^\vee(\phi_1) \cdot I(\mu_{ag_2})_{g_2}^\vee(\phi_2) = P(I(\mu_{ag_1})_{g_1}^\vee(\phi_1), I(\mu_{ag_2})_{g_2}^\vee(\phi_2)),$$

where P is the product Archimedean t -norm.

Thus, in the case of parallel fusion, where each agent works according to the Łukasiewicz t -norm, uncertainty propagation and granule synthesis are described by the Łukasiewicz t -norm L and extensions of logical intensions propagate according to the product t -norm P .

This example, although simple, gives clues as to what line to follow in the case of more complex schemes for fusion of knowledge.

16.7.2 Rough-Neural Computing

A variant of our general approach to multiagent systems presented above, can be adopted to describe a model of rough-neural computing [54].

We give here a concise description of this approach.

In neural models of computation, an essential feature of neurons is differentiability of transfer functions; hence, we introduce a special type of rough inclusions, called *gaussian* because of their form, by letting

$$\mu_G(x, y, r) \text{ iff } e^{-|\sum_{a \in DIS(x, y)} w_a|^2} \geq r, \quad (79)$$

where $w_a \in (0, +\infty)$ is a weight associated with the attribute a for each attribute $a \in A$; clearly, we retain notation of previous sections. One may notice that the gaussian rough inclusion is a modification of the rough inclusion μ_P obtained from the product t -norm.

Let us observe in passing that μ_G can be factored through the indiscernibility relation $IND(A)$, and thus its arguments can be objects as well as indiscernibility classes; we will freely use this fact.

Properties of Gaussian rough inclusions are following (cf., [54]):

- $x \text{ ing } y \text{ iff } DIS(x, y) = \emptyset$.
- There exists a function $\eta(r, s)$, such that $\mu_G(x, y, r), \mu_G(y, z, s)$ imply $\mu_G(x, z, \eta(r, s))$.
- If $x \text{ ing } g_r^{\mu_G} y, x \text{ ing } g_s^{\mu_G} z$, then $g_t^{\mu_G} x \text{ ing } g_r^{\mu_G} y, g_t^{\mu_G} x \text{ ing } g_s^{\mu_G} z$ for $t \geq \max\{r^4, s^4\}$.

Property 1 follows by definition, and Property 2 may be verified with $\eta(r, s) = r \cdot s \cdot e^{2 \cdot (\log r \cdot \log s)^{1/2}}$ [54]. Property 3 can be verified by observing that t should satisfy conditions $\eta(r, t) \geq r$ and $\eta(s, t) \geq s$ because of Property 2 (see [54]).

16.7.3 Rough Mereological Perceptron

The rough mereological perceptron is modeled on the perceptron, see [55], and it consists of an intelligent agent ag , endowed with a Gaussian rough inclusion μ_{ag} on the information system $I_{ag} = (U_{ag}, A_{ag})$ of the agent ag .

The input to ag is in the form of a finite tuple $\bar{x} = (x_1, \dots, x_k)$ of objects, and the input \bar{x} is converted at ag into an object $x = O_{ag}(\bar{x}) \in U_{ag}$ by means of an operator O_{ag} .

The rough mereological perceptron is endowed with a set of *target concepts* $T_{ag} \subseteq U_{ag}/IND(A_{ag})$, each target concept a class of the indiscernibility IND_{ag} .

Formally, a rough mereological perceptron is thus a tuple

$$RMP = (ag, I_{ag}, \mu_{ag}, O_{ag}, T_{ag}).$$

The output $res_{ag}(\bar{x})$ to RMP , at the input \bar{x} , is a granule of knowledge $g_{r(res)}x$ with

$$r(res) = \max\{r : \text{there is } y \in T_{ag} \text{ } \mu_{ag}(x, y, r)\}. \quad (80)$$

Formula (80) tells that the input \bar{x} is classified at ag as the collection of indiscernibility classes that are as close to x as the closest target class (closeness meant with respect to μ_{ag}).

16.7.4 Networks of Perceptrons

We describe a simple network here consisting of an output perceptron $RMP(ag)$ located with the agent ag and connected to input perceptrons $RMP(ag_i)$ for $i = 0, 1, 2, \dots, m$.

Target concept T_{ag} forms along with targets concepts T_{ag_i} for $i = 0, 1, \dots, m$ an *admissible set of targets* when $T_{ag} = O_{ag}(T_{ag_0}, \dots, T_{ag_m})$.

For an admissible set of targets $\Sigma = (T_{ag}, T_{ag_0}, \dots, T_{ag_m})$, one defines the functor $UncProp_\Sigma$ of uncertainty propagation in the following way.

Given objects $x_i \in U_{ag_i}$ for i from 0 to m , with $(x_0, \dots, x_m) = \bar{x}$, let

$$r_i = \max\{r : \mu_{ag_i}(x_i, T_{ag_i}, r)\}, \quad (81)$$

and for $\bar{r} = (r_0, \dots, r_m)$, we let

$$UncProp_\Sigma(\bar{r}) = \max\{s : \mu_{ag}(O_{ag}(\bar{x}, T_{ag}), s)\}. \quad (82)$$

It follows that $UncProp_\Sigma$ is defined over a finite subset, say R , of the cube $J = [0, 1]^{m+1}$.

We order linearly vectors in R by an ordering \prec that orders linearly J (e.g., lexicographically) into a chain r_0, \dots, r_n , with $n = |R|$; clearly, $r_n = (1, \dots, 1, \dots, 1)$.

We would like to produce from finitely many values of $UncProp_\Sigma$, a piece-wise linear function $LinUncProp_\Sigma$, a *linear uncertainty propagation* function; to this end, we call an indicator set, a maximal subchain $R_0 = (r_{i1}, \dots, r_{ik})$ in R with the property that (R_0, \prec) is Pareto ordered; i.e., $r_{ij,u} < r_{i(j+1),u}$ for each coordinate $u \in \{0, \dots, m\}$. Then, subsequent vectors $r_{ij}, r_{i(j+1)}$ in R_0 are antipodal vertices of a cube Q_j .

Let pr_j be the projection of the cube Q_j on its diagonal Δ_j joining vectors $r_{ij}, r_{i(j+1)}$, for each j .

Then, for vectors $r \in Q_j$, we find values of a, b such that

$$pr_j(r) = a \cdot r_{ij} + b \cdot r_{i(j+1)}, \quad (83)$$

and we define the value of

$$\text{LinUncProp}_{\Sigma}(r) = a \cdot \text{UncProp}_{\Sigma}(r_{ij}) + b \cdot \text{UncProp}_{\Sigma}(r_{i(j+1)}). \quad (84)$$

Thus, LinUncProp becomes a piecewise linear function over the union $Q = \bigcup_j Q_j$.

The computation by the network of perceptrons can be described as follows.

Given an input \bar{x} , to input perceptrons $\{\text{RMP}(ag_i)\}_i$, and an admissible set of targets Σ , the vector $r \in Q_p$ for some p is calculated and then the value $\text{res}'_{ag} = \text{LinUncProp}_{\Sigma}(r)$ is determined.

The result of computation on \bar{x} is the granule $\text{res}_{ag}(\bar{x}) = g \text{res}'_{ag} T_{ag}$, where $\Sigma = (T_{ag}, T_{ag_0}, \dots, T_{ag_m})$.

Let us observe that values of LinUncProp depend on the function $f(x, y) = e^{-|\sum_{a \in \text{DIS}(x, y)} w_a|^2}$. The direction of changes is indicated by the gradient

$$\frac{\partial f}{\partial w} = f \cdot \left(-2 \cdot \sum w_a \right). \quad (85)$$

The learning problem for the network of perceptrons can be stated as follows:

Given: A test sample $s = (\bar{x}_1, \dots, \bar{x}_k)$ of positive examples for a target concept $g_r t$ for some $t = T_{ag}$

Find: Weights w_a such that $\text{res}_{ag}(\bar{x}_i)$ ing $g_r t$ for $i = 1, 2, \dots, k$.

16.8 Granular Decision Systems

In this section, we apply our formal apparatus to real-life data. These data come in the form of decision systems [21]. A decision system is a triple (U, A, d) , where (U, A) is an information system and $d \notin A$ is an additional attribute called the decision: $d : U \rightarrow V_d$.

Relations between the decision and a group of attributes $B \subseteq A$ are expressed by means of decision rules, i.e., descriptor formulas of the form

$$\bigwedge_{a \in B} (a = v_a) \Rightarrow (d = v),$$

where $v \in V_d$, $v_a \in V_a$ for each $a \in B$.

Rough sets paradigm has served for working out many algorithms for decision rule induction; as distinguished in [56], there are three main kinds of classifiers searched for: *minimal*, i.e., consisting of minimum possible number of rules describing decision classes in the universe; *exhaustive*, i.e., consisting of all possible rules; *satisfactory*, i.e., containing rules tailored to a specific use. Classifiers are evaluated globally with respect to their ability to properly classify objects, usually by *error* which is the ratio of the number of correctly classified objects to the number of test objects, *total accuracy* being the ratio of the number of correctly classified cases to the number of recognized cases, and *total coverage*, i.e., the ratio of the number of recognized test cases to the number of test cases.

Minimum size algorithms include LEM2 algorithm due to Grzymala-Busse [57] and covering algorithm in RSES package [36]; exhaustive algorithms include, e.g., LERS system due to Grzymala-Busse [58], systems based on discernibility matrices and Boolean reasoning introduced by Skowron [59], see also [60, 61], and implemented in the RSES package [36].

Minimal consistent sets of rules were introduced in Skowron and Rauszer [62]. Further developments include dynamic rules, approximate rules, and relevant rules as described in [60, 61] as well as local rules [61], effective in implementations of algorithms based on minimal consistent sets of rules. Rough-set-based classification algorithms, especially those implemented in the RSES system [36], were discussed extensively in [61].

The idea of forming a granular counterpart to a given information system was proposed in [18] as follows. For an information system (U, A) , and a granulation scheme \mathcal{G} , which yields a granule collection $Gr = \mathcal{G}(U)$, a covering of the universe U , $Cov(U)$, is chosen by a selected strategy \mathcal{C} ; adopting a strategy \mathcal{S} , for each granule $g \in Cov(U)$ and each attribute $a \in A$, a value $\bar{a}(g) = \mathcal{S}\{a(u) : u \in g\}$ is determined. The new information system $(Cov(U), \bar{A})$, where $\bar{A} = \{\bar{a} : a \in A\}$, is a granular approximation to the

Table 16.3 A comparison of errors in classification by rough set and other paradigms on Australian credit data set

Paradigm	System/Method	Australian credit
Stat. methods	Logdisc	0.141
Stat. methods	SMART	0.158
Neural nets	Backpropagation2	0.154
Neural networks	RBF	0.145
Decision trees	CART	0.145
Decision trees	C4.5	0.155
Decision trees	ITrule	0.137
Decision rules	CN2	0.204
Rough sets	NNANR	0.140
Rough sets	DNANR	0.165
Rough sets	Best result	0.130 (SNAPM)

original information system (U, A) . Clearly, the same concerns any decision system (U, A, d) with the reduced decision \overline{d} .

As the mechanism of granule formation is based on an abstractly understood similarity among objects, a conjecture that decision algorithms/classifiers induced from granular approximations should approximate decision algorithms/classifiers induced from the original decision system to a satisfactory degree, at substantial decline in size of both the universe of objects and the decision algorithm, has been posed in [18] and repeated in [29, 63].

16.8.1 Granular Classifiers

We apply the idea of a granulated decision system to Australian credit data set [64]. We apply the well-known and tested algorithms for decision rule induction: the exhaustive and the covering algorithms of RSES system [36] and the LEM2 algorithm due to Grzymala-Busse [57, 58], put into public domain within the RSES system. As the strategy for generating granular coverings, the exhaustive strategy based on ordering of objects and the sequential choice of granules until a covering is found is applied. Granules were computed according to the formula (15); i.e., by Theorem 4 the granule g, x can be computed as the set consisting of objects y such that at least $r \cdot 100\%$ of attributes take on x and y the same value. The strategy for finding values of reduced attributes has been chosen as majority voting with the random resolution of ties.

The rough-set-based rule induction systems as compared to other methods give on Australian credit data set the following results shown in Table 16.3. The results were obtained in [60].

For comparison, we include in Table 16.4 best results obtained by template and similarity methods on Australian credit data set obtained in [6].

The results for Australian credit data set obtained for granulated data are shown in Table 16.5. The method was train-and-test: rules were induced on granulated 50% of data and tested on the remaining

Table 16.4 Accuracy of classification by template and similarity methods

Paradigm	System/Method	Australian credit
Rough sets	Simple.templ. / Hamming	0.8217
Rough sets	Gen.templ. / Hamming	0.855
Rough sets	Simple.templ. / Euclidean	0.8753
Rough sets	Gen.templ. / Euclidean	0.8753
Rough sets	Match. tolerance	0.8747
Rough sets	Clos. tolerance	0.8246

Table 16.5 Australian credit data set

r	tst	trn	rulcov	rulex	rullem	acov	ccov	aex	clex	alem	clem
Nil	345	345	571	5597	49	0.634	0.791	0.872	0.994	0.943	0.354
0.0	345	1	14	0	0	1.0	0.557	0.0	0.0	0.0	0.0
0.0714286	345	1	14	0	0	1.0	0.557	0.0	0.0	0.0	0.0
0.142857	345	2	16	0	1	1.0	0.557	0.0	0.0	1.0	0.383
0.214286	345	3	7	7	1	0.641	1.0	0.641	1.0	0.600	0.014
0.285714	345	4	10	10	1	0.812	1.0	0.812	1.0	0.0	0.0
0.357143	345	8	18	23	2	0.820	1.0	0.786	1.0	0.805	0.252
0.428571	345	20	29	96	2	0.779	0.826	0.791	1.0	0.913	0.301
0.5	345	51	88	293	2	0.825	0.843	0.838	1.0	0.719	0.093
0.571429	345	105	230	933	2	0.835	0.930	0.855	1.0	0.918	0.777
0.642857	345	205	427	3157	20	0.686	0.757	0.867	1.0	0.929	0.449
0.714286	345	309	536	5271	45	0.629	0.774	0.875	1.0	0.938	0.328
0.785714	345	340	569	5563	48	0.629	0.797	0.870	1.0	0.951	0.357
0.857143	345	340	570	5574	48	0.626	0.791	0.864	1.0	0.951	0.357
0.928571	345	342	570	5595	48	0.628	0.794	0.867	1.0	0.951	0.357
1.0	345	345	571	5597	49	0.634	0.791	0.872	0.994	0.943	0.354

r, granule radius; tst, test sample size; trn, training sample size; rulcov, number of rules with covering algorithm; rulex, number of rules with exhaustive algorithm; rullem, number of rules with LEM2; acov, total accuracy with covering algorithm; ccov, total coverage with covering algorithm; aex, total accuracy with exhaustive algorithm; cex, total coverage with exhaustive algorithm; alem, total accuracy with LEM2; clem, total coverage with LEM2.

50%. The granulation radii follow from the formula (15). The value ‘nil’ of the radius denotes results for non-granulated data.

Conclusions for Australian Credit Data Set

With covering algorithm, accuracy is better than or within error of 1% for all radii, coverage is better than or within error of 4.5% from the radius of 0.214860 on where training set size reduction is 99%, and reduction in rule set size is 98%.

With exhaustive algorithm, accuracy is within error of 10% from the radius of 0.285714, and it is better than or within error of 4% from the radius of 0.5, where reduction in training set size is 85% and reduction in rule set size is 95%. The result of 0.875 at $r = 0.714$ is among the best at all (see Table 16.2). Coverage is better from $r = 0.214$ in the granular case; reduction in objects is 99%; reduction in rule size is almost 100%.

LEM2 gives accuracy better than or within 2.6% error from the radius of 0.5, where training set size reduction is 85% and rule set size reduction is 96%. Coverage is better than or within error of 7.3% from the radius of 0.571429, where reduction in training set size is 69.6% and rule set size is reduced by 96%.

An improved technique was proposed as concept-dependent granulation: in this approach, granules of given radii are computed about objects in data but only objects that fall into the same decision class as the granule center are taken into consideration. The results are better in this case; not giving the whole table of results, we conclude this section with Table 16.6 in which we summarize the best classification results obtained by rough set methods on Australian credit data set. The full survey of results obtained by the reviewed-here granular approach to data classification can be found in [19]. Granular results were obtained in the setting of Table 16.5, except for covering selection, which in case of Table 16.6 was a random choice and for using the tenfold cross validation. As an algorithm for rule induction, the RSES [36] exhaustive algorithm was used.

Table 16.6 Best results for Australian credit by rough-set-based algorithms

Source	Method	Accuracy	Coverage
Bazan [60]	SNAPM (0.9)	Error = 0.130	—
Nguyen [6]	Simple.templates	0.929	0.623
Nguyen [6]	General.templates	0.886	0.905
Nguyen [6]	Closest.simple.templates	0.821	1.0
Nguyen [6]	Closest.gen.templates	0.855	1.0
Nguyen [6]	Tolerance.simple.templ.	0.842	1.0
Nguyen [6]	Tolerance.gen.templ.	0.875	1.0
Wroblewski [65]	Adaptive.classifier	0.863	—
(Polkowski and Artiemjew, [19])	Granular*.r = 0.642	0.8990	1.0
(Polkowski and Artiemjew, [19])	Granular**.r = 0.714	0.964	1.0
(Polkowski and Artiemjew, [19])	Granular***.concept.r = 0.785	0.9970	0.9995

In case *, reduction in object size is 49.9%, reduction in rule number is 54.6%; in case **, resp., 19.7, 18.2; in case ***, resp., 3.6, 1.9.

As Table 16.6 does witness, results obtained with the granular approach are better than those previously obtained, especially in case of concept-dependent granulation. Clearly, one has to realize that some role is played by specificity of data and various data sets may give different results with different approaches to rule induction.

16.9 Conclusion

We have presented a substantial fragment of the theory of granular computing in a formalized version. The apparatus adopted here allows for the establishing of many formal properties of granules. It is also possible to produce a variety of rough inclusions according to formulas demonstrated throughout this chapter.

In applications discussed here, in many-agent and distributed reasoning, cognitive computing, knowledge fusion, and classification tasks in data, practically one rough inclusion, namely, μ_L , modeled on the Lukasiewicz t -norm has been applied for demonstration sake. This may prompt the reader to feel that basically the results given in the application sections can almost be obtained without any help of the apparatus presented. Indeed, many results were obtained in rough set theory, for instance, on the basis of intuitive or probabilistic foundations by taking formulas such as (15) as the basis for reasoning. It seems obvious that a possibility for still better results exists with other rough inclusions, applied locally. In our recent research on granulated data sets, other rough inclusions have proved their usefulness and results will be published in near future.

Acknowledgment

The author expresses his debt to colleagues with whom he worked in parallel or sometimes jointly for a number of years on the topic of rough sets, their contributions witnessed by the papers quoted.

References

- [1] L.A. Zadeh. Fuzzy sets and information granularity. In: M. Gupta, R. Ragade, and R. Yager (eds), *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, 1979, pp. 3–18.
- [2] L. Polkowski and A. Skowron. Rough mereological calculi of granules: A rough set approach to computation. *Comput. Intell. Int. J.* 17 (2001) 472–492.

- [3] T.Y. Lin. From rough sets and neighborhood systems to information granulation and computing with words. In: *Proceedings of the European Congress on Intelligent Techniques and Soft Computing*, Verlag Mainz, Aachen, 1997, pp. 1602–1606.
- [4] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley Interscience, New York, 2001.
- [5] A. Wojna. Analogy-based reasoning. *Transactions on Rough Sets IV, Lecture Notes in Computer Science*. Vol. 3700. Springer-Verlag, Berlin, 2005, pp. 277–374.
- [6] N.S. Hoa. Regularity analysis and its applications in data mining. In: L. Polkowski, S. Tsumoto, and T.Y. Lin (eds). *Rough Set Methods and Applications. New Developments in Knowledge Discovery in Information Systems*. Physica-Verlag, Heidelberg, 2000, pp. 289–378.
- [7] A. Skowron and J. Stepaniuk. Information granules: Towards foundations of granular computing. *Int. J. Intell. Syst.* 16 (2001) 57–85.
- [8] A. Skowron and J. Stepaniuk. Information granules and rough-neural computing. In: S. K. Pal, L. Polkowski, and A. Skowron (eds.), *Rough-Neural Computing. Techniques for Computing with Words*. Springer-Verlag, Berlin, 2004, pp. 43–84.
- [9] Z. Herbert. *Wiersze Wybrane (Selected Poems, in Polish)*. Wyd. a5, Kraków, 2004.
- [10] T.Y. Lin. Granular computing: Examples, intuitions and modeling. In: X. Hu, Q. Liu, A. Skowron, T.Y. Lin, R.R. Yager, and B. Zhang (eds), *Proceedings of 2005 IEEE Conference on Granular Computing*, GrC05, Beijing, China, July 2005. IEEE Press, Piscataway, NJ, 2005, pp. 40–44.
- [11] Y.Y. Yao. Perspectives of granular computing. In: X. Hu, Q. Liu, A. Skowron, T.Y. Lin, R.R. Yager, and B. Zhang (eds) *Proceedings of 2005 IEEE Conference on Granular Computing*, GrC05, Beijing, China, July 2005. IEEE Press, Piscataway, NJ, 2005, pp. 85–90.
- [12] Y.Y. Yao. Granular computing: Basic issues and possible solutions. In: P.P. Wang (ed.), *Proceedings of the 5th Joint Conference Information Sciences I*. Association for Intelligent Machinery Atlantic, NJ, 2000, pp. 186–189.
- [13] L. Polkowski, A. Skowron, and J. Zytkow. Tolerance based rough sets. In: T.Y. Lin and M.A. Wildberger (eds), *Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management*. Simulation Councils Inc., San Diego, 1995, pp. 55–58.
- [14] E.C. Zeeman. The topology of the brain and the visual perception. In: K.M. Fort (ed), *Topology of 3-manifolds and Selected Topics*. Prentice Hall, Englewood Cliffs, NJ, 1965, pp. 240–256.
- [15] Y.Y. Yao. Information granulation and approximation in a decision-theoretic model of rough sets. In: S. K. Pal, L. Polkowski, and A. Skowron (eds), *Rough-Neural Computing. Techniques for Computing with Words*. Springer-Verlag, Berlin, 2004, pp. 491–516.
- [16] Q. Liu and H. Sun. Theoretical study of granular computing. In: *Proceedings RSCT06 (First International Conference on Rough Sets and Knowledge Technology)*. Chongqing, China, 2006. Lecture Notes in Artificial Intelligence, Vol. 4062. Springer-Verlag, Berlin, 2006, pp. 92–102.
- [17] R. Słowiński and D. Vanderpoorten. A generalized definition of rough approximations based on similarity. *IEEE Trans. Data Knowl. Eng.* 12 (2000) 331–336.
- [18] L. Polkowski. Formal granular calculi based on rough inclusions (a feature talk). In: X. Hu, Q. Liu, A. Skowron, T.Y. Lin, R.R. Yager, and B. Zhang (eds), *Proceedings of 2005 IEEE Conference on Granular Computing*, GrC05, Beijing, China, July 2005. IEEE Press, Piscataway, NJ, 2005, pp. 57–62.
- [19] L. Polkowski and P. Artemjew. On grnular rough computing: Factoring classifiers through granulated decision systems. In: *Lecture Notes in Artificial Intelligence*. Vol. 4585. Springer-Verlag, Berlin, 2007, pp. 280–289.
- [20] Z. Pawlak. Rough sets, *Int. J. Comput. Inf. Sci.* 11 (1982) 341–356.
- [21] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer, Dordrecht, 1991.
- [22] *Stanford Encyclopedia of Philosophy: Transworld Identity*. <http://plato.stanford.edu/entries/identity-transworld/>; Fall 2006 (September 21) edition.
- [23] F. Baader, D. Calvanese, D. McGuinness, D.Nardi, and P. Patel-Schneider (eds). *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Cambridge, UK, 2004.
- [24] S. Leśniewski. *Podstawy ogólnej teorii mnogości* (On the foundations of set theory, in Polish). The Polish Scientific Circle, Moscow, 1916; see also a later digest in: Topoi 2(1982) 7–52 and Foundations of the General Theory of Sets. I. In: S.J. Surma, J. Srzednicki, D.I. Barnett, and F.V. Rickey (eds), *S. Leśniewski. Collected Works*, Vol. 1. Kluwer, Dordrecht, 1992, pp. 129–173.
- [25] A.N. Whitehead. *Process and Reality. An Essay in Cosmology*. Macmillan, New York, 1929. Corrected edition 1978.
- [26] H. Leonard and N. Goodman. The calculus of individuals and its uses. *J. Symb. Log.* 5 (1940) 45–55.
- [27] B.L. Clarke. A calculus of individuals based on connection. *Notre Dame J. Form. Log.* 22 (1981) 204–218.
- [28] L. Polkowski. Iward rough set foundations. Mereological approach (a plenary lecture). In: *Proceedings RSCTC04 (Rough Sets and Current Trends in Computing)*, Uppsala, Sweden, 2004, *Lecture Notes in Artificial Intelligence*, Vol. 3066. Springer-Verlag, Berlin, 2004, pp. 8–25.

- [29] L. Polkowski. A model of granular computing with applications: Granules from rough inclusions in information systems. In: Y.-Q. Zhan and T.Y. Lin (eds), *Proceedings of 2006 IEEE Conference on Granular Computing*, GrC06, Atlanta, USA, May 2006, IEEE Press, Piscataway, NJ, 2006, pp. 9–16.
- [30] P. Hájek. *Metamathematics of Fuzzy Logic*. Kluwer, Dordrecht, 1998.
- [31] A.N. Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Am. Math. Soc. Transl.* 28 (1963) 55–59.
- [32] C.-H. Ling. Representation of associative functions. *Publ. Math. Debrecen* 12 (1965) 189–212.
- [33] L. Polkowski. *Rough Sets. Mathematical Foundations*. Physica-Verlag, Heidelberg, 2002.
- [34] V. Novák, I. Perfiliewa, and J. Močkoř. *Mathematical Principles of Fuzzy Logic*. Kluwer Academic Publishers, Boston, 1999.
- [35] W. Klösgen and J. Zytkow (eds). *Handbook of Data Mining and Knowledge Discovery*. Oxford University Press, Oxford, 2002.
- [36] A. Skowron, J.G. Bazan, P. Synak, J. Wróblewski, N.H. Son, N.S. Hoa, and A. Wojna. RSES: A system for data analysis. <http://logic.mimuw.edu.pl/rses>, accessed January 22, 2008.
- [37] P.S. Mostert and A.L. Shields. On the structure of semigroups on a compact manifold with a boundary. *Ann. Math.* 65 (1957) 117–143.
- [38] W.M. Faucett. Compact semigroups irreducibly connected between two idempotents. *Proc. Am. Math. Soc.* 6 (1955) 741–747.
- [39] L.A. Zadeh. Similarity relations and fuzzy orderings. *Inf. Sci.* 3 (1971) 177–200.
- [40] J. vanBenthem. *A Manual of Intensional Logic*. CSLI Stanford University, Stanford, CA, 1988.
- [41] R. Thomason (ed.). *Philosophical Writings of R. Montague*. Yale University Press, New Haven, CT, 1972.
- [42] E. Orłowska. Modal logics in the theory of information systems. *Z. Math. Log. Grund. Math.* 35 (1989) 559–572.
- [43] E. Orłowska and Z. Pawlak. Representation of non-deterministic information. *Theory Comput. Sci.* 29 (1984) 27–39.
- [44] H. Rasiowa and A. Skowron. Rough concepts logic. In: A. Skowron (ed.), *Computation Theory, Lecture Notes in Computer Science*. Vol. 208. Springer-Verlag, Berlin, 1985, pp. 288–297.
- [45] H. Rasiowa and A. Skowron. Approximation logic. In: *Proceedings of the Conference on Mathematical Methods of Specification and Synthesis of Software Systems*. Akademie Verlag, Berlin, 1985, pp. 123–139.
- [46] D. Vakarelov. Information systems, similarity relations and modal logics. In: E. Orłowska (ed.), *Incomplete Information: Rough Set Analysis*. Physica-Verlag, Heidelberg, 1998, pp. 492–550.
- [47] Z. Pawlak and A. Skowron. Rough membership functions. In: R.R. Yager, M. Fedrizzi, and J. Kasprzyk (eds), *Advances in the Dempster–Shafer Theory of Evidence*. John Wiley and Sons, New York, 1994, pp. 251–271.
- [48] L. Polkowski and M. Semeniuk-Polkowska. On rough set logics based on similarity relations. *Fundam. Inf.* 64 (2005) 379–390.
- [49] L. Borkowski (ed.), *Jan Łukasiewicz. Selected Works*. North-Holland and Polish Scientific Publishers, Amsterdam and Warsaw, 1970.
- [50] J.B. Rosser and A.R. Turquette. *Many-Valued Logics*. North-Holland, Amsterdam, 1958.
- [51] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets Syst.* 1 (1978) 3–28.
- [52] D. Dubois and H. Prade. Necessity measures and the resolution principle. *IEEE Trans. Syst. Man, Cybern.* 17 (1997) 111–127.
- [53] L. Polkowski. A note on 3-valued rough logic accepting decision rules. *Fundam. Inf.* 61 (2004) 37–45.
- [54] L. Polkowski. A rough-neural computation model based on rough mereology. In: S. K. Pal, L. Polkowski, and A. Skowron (eds), *Rough-Neural Computing. Techniques for Computing with Words*. Springer-Verlag, Berlin, 2004, pp. 85–108.
- [55] C.M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon, Oxford, 1997.
- [56] J. Stefanowski. On rough set based approaches to induction of decision rules. In: L. Polkowski and A. Skowron (eds), *Rough Sets in Knowledge Discovery*. Vol. 1. Physica-Verlag, Heidelberg, 1998, pp. 500–529.
- [57] J.W. Grzymala-Busse. Data with missing attribute values: Generalization of rule indiscernibility relation and rule induction. *Transactions on Rough Sets I. Lecture Notes in Computer Science*. Vol. 3100. Springer-Verlag, Berlin, 2004, pp. 78–95.
- [58] J.W. Grzymala-Busse and Ming Hu. A comparison of several approaches to missing attribute values in data missing. *Lecture Notes in Artificial Intelligence*. Vol. 2005. Springer-Verlag, Berlin, 2000, pp. 378–385.
- [59] A. Skowron. Boolean reasoning for decision rules generation. In: J. Komorowski and Z. Ras (eds), *Proceedings of ISMIS'93. Lecture Notes in Artificial Intelligence*. Vol. 689. Springer-Verlag, Berlin, 1993, pp. 295–305.
- [60] J.G. Bazan. A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables. In: L. Polkowski and A. Skowron (eds). *Rough Sets in Knowledge Discovery*, Vol. 1. Physica-Verlag, Heidelberg, 1998, pp. 321–365.

- [61] J.G. Bazan, N.H. Son, P. Synak, J. Wróblewski, N.S. Hoa. Rough set algorithms in classification problems. In: L. Polkowski, S. Tsumoto, and T.Y. Lin (eds), and *Rough Set Methods and Applications. New Developments in Knowledge Discovery in Information Systems*. Physica-Verlag, Heidelberg, 2000, pp. 49–88.
- [62] A. Skowron and C. Rauszer. The discernibility matrices and functions in decision systems. In: R. Słowiński (ed.), *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer, Dordrecht, 1992, pp. 311–362.
- [63] L. Polkowski. Rough mereological reasoning in rough set theory: Recent results and problems (plenary talk). In: *Proceedings RSKT06 (First International Conference on Rough Sets and Knowledge Technology)*, Chongqing, China, 2006. *Lecture Notes in Artificial Intelligence*. Vol. 4062, Springer-Verlag, Berlin, 2006, pp. 79–92.
- [64] UCI Repository. <http://www.ics.uci.edu/ml/datasets.html>, accessed January 22, 2008.
- [65] J. Wróblewski. Adaptive aspects of combining approximation spaces. In: S. K. Pal, L. Polkowski, and A. Skowron (eds), *Rough-Neural Computing. Techniques for Computing with Words*. Springer-Verlag, Berlin, 2004, pp. 139–156.

Further Reading

- [1] S. K. Pal, L. Polkowski, and A. Skowron (eds.). *Rough-Neural Computing. Techniques for Computing with Words*. Springer-Verlag, Berlin, 2004.
- [2] X. Hu, Q. Liu, A. Skowron, T.Y. Lin, R.R. Yager, and B. Zhang (eds). *Proceedings of 2005 IEEE Conference on Granular Computing*, GrC05, Beijing, China, July 2005. IEEE Press, Piscataway, NJ, 2005.
- [3] Y.-Q. Zhan and T.Y. Lin (eds.). *Proceedings of 2006 IEEE Conference on Granular Computing*, GrC06, Atlanta, USA, May 2006, IEEE Press, Piscataway, NJ, 2006.
- [4] L.A. Zadeh. Graduation and Granulation are keys to computation with information described in Natural Language, In: X. Hu, Q. Liu, A. Skowron *et al.* (eds), *Proceedings of 2005 IEEE Conference on Granular Computing*, GrC05, Beijing, China, July 2005. IEEE Press, Piscataway, NJ, 2005. p. 30.

17

A Unified Framework of Granular Computing

Yiyu Yao

17.1 Introduction

An early developing stage of a theory or a methodology is typically characterized by a diversity of many views, proposals, and models, but the lack of a unified framework. Ideas are scattered, fragmentary, and isolated, instead of forming an integrated whole. With extensive studies and better understanding, it is expected that a much smaller set of well-accepted and dominant views will eventually converge. A challenge is how to speed up this process so that the theory can be effectively used by many more people.

As an emerging field of study, granular computing faces the same challenge. In the past few years we have witnessed a fast-growing interest in this area [1–7]. On the one hand, many interpretations, models, paradigms, methodologies, techniques, and tools have been proposed and investigated [8–23]. On the other hand, there does not exist a commonly accepted definition nor a commonly agreed framework [24–29]. The lack of a conceptual framework may slow down the further development of granular computing and make it difficult for us to see and exploit the universal applicability, flexibility, and effectiveness of granular computing. At this early stage, it may be impossible to define precisely without controversy what granular computing is, its scopes, its theories, and its methodologies [27]. Nevertheless, results from the existing studies suggest that we are making good progress toward a conceptual framework for granular computing [26–29]. The main objective of this chapter is to examine the basic components of such a framework.

From the existing studies, we can observe several limitations and problems. Many studies focus on specific issues, concrete models, and domain-specific methodologies. The existing studies are dominated by computational intelligence theories, including fuzzy sets, rough sets, neural networks, interval computing, and many more. It is also not surprising to find that some studies are simply reformulations of existing results by using the terminologies of granular computing without the necessary new insight. A conceptual framework of granular computing would enable us to avoid such problems. Studies of granular computing must be pursued in depth and results from those studies be integrated in breadth. Granular computing needs to be an interdisciplinary study related to many branches of science, moving away from the current domination of fuzzy sets and rough sets. It is also necessary to investigate distinguishing properties that justify granular computing as a separate field of study in its own right.

The basic principles and ideas of granular computing have, in fact, long appeared in many branches of science and many fields of computer science [26–29]. Unfortunately, they are scattered over many places

in isolation and are not readily accessible, as they are either described and discussed under different names or buried in domain-specific details. Yet, those effective ideas and principles lend themselves immediately for a conceptual framework of granular computing. Two tasks are involved in building this framework. One is to extract high-level commonalities of different disciplines and synthesize their results into an integrated whole by ignoring low-level details. The other is to make explicit ideas hidden in discipline-specific discussions in order to arrive at a set of discipline-independent principles.

In our view, granular computing is a new field of study that has emerged from many different disciplines and fields, including general systems theory [30–32], hierarchy theory [33–37], social networks [31, 38, 39], artificial intelligence [40–44], human problem solving [45], learning [46, 47], programming [48–51], theory of computation [52], and information processing [53, 54]. Although granular computing draws heavily on results from other fields, it has unique and distinguishing characteristics. The proposed framework of granular computing is based on three related perspectives [55]. From the philosophical perspective, granular computing offers a new world view that leads to structured thinking. From the methodological perspective, granular computing deals with structured problem solving. From the computational perspective, granular computing concerns structured information processing. The integration of the three perspectives results in a holistic understanding of granular computing that emphasizes structures embedded in a web of granules.

The main aim of the framework is to bring a clear understanding of granular computing. The framework may not be completely accurate and many of its components and views may have to be refined with time. Although it is not certain if every perspective of the framework will be accepted eventually, there is no doubt that the study of a unified framework will play a crucial role in the development of a full theory of granular computing.

17.2 Philosophical Perspective: Structured Thinking

The philosophical view of granular computing may have a great impact on the current research in the field. Such a philosophical foundation, however, has hardly been examined. Although it may be too early to pinpoint this philosophical view, we can at least discuss and elaborate on some of its important features [26–29]. We believe that the philosophy of granular computing is structured thinking, characterized by hierarchical modeling, understanding, processing, and learning. These hierarchical processing tasks are essential to human intelligence [41, 46, 47, 56]. They have significant implications for knowledge-intensive information systems.

To put granular computing in its right perspective, we need to first briefly mention two complementary philosophical views dealing with the complexity of real-world problems, namely, the traditional reductionist thinking and the new systems thinking. According to reductionist thinking, a complex system or problem can be divided into simpler and more fundamental parts, and they can be further divided. An understanding of the system can be reduced to the understanding of its parts. In other words, we can deduce fully the properties of the system based solely on the properties of its parts. In contrast, systems thinking shifts from parts to the whole, in terms of connectedness, relationships, and context [30–32]. A complex system is viewed as an integrated whole consisting of a web of interconnected, interacting, and highly organized parts. The properties of the whole are not present in any of its parts, but emerge from the interactions and relationships of the parts.

The reductionist thinking and systems thinking are considered by many as competing views. Since each of them is effective in modeling and solving different types of problems, we consider the two as complementary views. The existing research on granular computing is mainly influenced by the reductionist view. For example, phrases representing reductionist thinking, such as ‘divide and conquer’ and ‘granulate and conquer,’ are used to explain the workings of granular computing. This bias may be corrected if granular computing can also draw results from systems thinking.

The reductionist thinking and systems thinking agree on the modeling of a complex system in terms of whole and parts, but differ in how to make inference with the parts. The two views exploit a common structure known as the hierarchical structure characterized by multiple levels. According to reductionist thinking, a system can be continually divided into smaller and smaller parts to form a multilevel hierarchical

representation and understanding. In systems thinking, one can form different systems levels so that systems can be nested within other systems. Based on this common hierarchical structure, granular computing attempts to unify reductionist thinking and systems thinking.

Hierarchical structures and organizations exist in the real world or, more precisely, our perception of the real world. They can be found in many natural, social, and man-made systems [34, 36, 37, 39]. Humans have evolved to deal with such hierarchical structures effectively and efficiently [6, 36, 41, 42, 45]. Human perception and understanding of the world depends, to a large extent, on nested and hierarchical structures. We view and represent the world using various grain sizes and abstract only those things that serve the present interests. The ability to conceptualize the world at different levels of granularity and to switch among these levels is fundamental to human intelligence and flexibility [42]. It is also interesting to comment that hierarchical structures have been used by some authors to explain the human brain and intelligence. For example, Hawkins [41] proposes that the human brain can be conceptually understood by a cortical hierarchy model that mirrors the hierarchical structures of the real world.

The notion of hierarchical structures captures the essential features of our perception and understanding of the world at multiple levels of granularity [6, 9, 42]. Granular computing, formulated on the basis of hierarchical structures, promotes a way of structured thinking by combining analytic and synthetic methods. Analytic thinking involves the division of a whole into relatively independent parts. This allows us to move to a lower level in a hierarchical structure where individual properties of parts can be studied. On the other hand, synthetic thinking enables us to combine parts into a complex whole. This enables us to move to a higher level in a hierarchical structure where the emergent properties of the whole can be examined.

17.3 Methodological Perspective: Structured Problem Solving

The philosophical foundations of granular computing is a view of the world in terms of granules and multiple levels of granularity. In search of methods of problem solving, this hierarchical structure plays a crucial role. From the methodological perspective, granular computing is structured problem solving guided by the structured thinking. By drawing results from structured programming, artificial intelligence, hierarchy theory, rough set theory [57], quotient space theory [44, 58], and others, one may extract a set of fundamental principles for systematic problem solving [26–29, 55].

The philosophy of granular computing implies two mutually dependent tasks of structured problem solving, namely, constructing a hierarchical view and working with the associated hierarchy. In some cases, the separation of the two tasks is not clear. It may happen that the two tasks are tied together instead of one following the other. Many principles can be applied to both tasks. As examples, we examine three such principles.

A fundamental principle of granular computing is ‘the principle of multilevel granularity.’ This principle stresses the importance of breaking a large problem into smaller problems and understanding a problem at many levels of detail. One may construct many hierarchical views and select the most suitable view. A levelwise construction process can be done in either a top-down or a bottom-up manner, based on the properties of loose coupling of parts and near decomposability [36]. A top-down levelwise construction process is consistent with the breadth-first search strategy of artificial intelligence. Alternatively, one may construct a hierarchical view based on the depth-first search strategy. Once a hierarchical view is created, working with a hierarchy is natural at multiple levels of granularity.

The principle of multilevel granularity can in fact be applied to the problem of constructing a hierarchical view. One may consecutively build different versions of a hierarchy with differing details. For example, a level in one version may be divided into two or more levels in the next version. These different versions naturally reflect our multiple understandings of the problem.

Another principle of granular computing is ‘the principle of focused effort.’ This principle states that, at a given stage of constructing a hierarchy and working with the hierarchy, effort is to be concentrated on a particular granule or a specific level, relatively independent of other granules or levels. In doing so, one abstracts only those things that are relevant to the present interests and ignores irrelevant lower level

details or relationships to other things. The principle does not rule out the needs for some effort to be made on the study of related things. It requires the concentration of the major effort on a part, instead of the whole, at a specific point in time. The applications of this principle result in a concrete sequence of steps toward a complete and structured solution to a problem.

The third principle of granular computing is ‘the principle of granularity conversion.’ It calls for an easy switch between levels of abstraction. According to this principle, a hierarchy describing a problem should be constructed in a way to facilitate easy granularity conversion. When working with a hierarchy, one can fluently switch levels of granularity as well as pass information between levels.

The top-down analytic methods may be helpful in switching from a higher level to a lower level. By analysis, a large granule is broken into smaller granules. A solution with respect to larger granules can be derived by combining solutions from the corresponding families of smaller granules. In contrast, the bottom-up synthetic methods may be used for switching from a lower level to a higher level. By synthesis, the connections and interactions of lower level granules may be studied and integrated. This may reveal emergent properties, the properties that none of the lower-level granules has, at the higher level.

The three principles are not new and have been either explicitly or implicitly used in many fields. For example, they can be easily seen from the principles of structured programming, although they are stated differently [48, 49, 51]. Our main objective is to demonstrate that, from the methodological perspective, granular computing is structured problem solving based on principles proved to be effective across different disciplines. Many other principles can be similarly reinterpreted in light of granular computing. Collecting and presenting coherently these principles remain a great challenge.

17.4 Computational Perspective: Structured Information Processing

From the computational perspective, granular computing is structured information processing. It is about the applications of the granular computing philosophy and principles in the design and implementation of intelligent information systems. Our exploration of the computational perspective is based on two studies: the pyramid approach suggested by Bargiela and Pedrycz [1] for granular computing and the multiple-level approach proposed by Marr [54] for human and computer vision.

In the information-processing paradigm proposed by Bargiela and Pedrycz [1], granular computing works with a pyramid consisting of levels of different-sized information granules, i.e., a hierarchical structure. The processing at different levels of information granulation is a necessary feature of any knowledge-intensive system. In the study of human representation and processing of visual information, Marr [54] makes a convincing argument for a multilevel understanding of an information-processing system. At different levels of description, one explores different kinds of explanations. A coherent explanation, hopefully, may be obtained from explanations at those linked levels. The philosophical views and basic working principles, though presented with reference to information processing, are indeed much similar to ones of granular computing we discussed earlier.

For the explanation of specific information-processing mechanisms and systems, Marr [54] uses the two notions of representation and process. A representation is a formal system that makes explicit certain entities or types of information and a specification of how the system does this. The result of using a representation to describe an entity is called a description of the entity in the representation. A process may simply be interpreted as actions or procedures for carrying out information-processing tasks. It may also be interpreted as a mapping from one representation to another representation. In general, a representation may determine the effectiveness of processes under the representation. It may be necessary to choose a set of the most appropriate processes.

It is easy to describe the computational perspective on granular computing based on representation and process. As a minimum requirement, a representation of granules must capture essential features of granules and make a particular aspect of their physical meanings explicit. Furthermore, the representation of granules needs to be closely connected to representations of granular structures with respect to granules, levels, and hierarchies. In some cases, it may be possible to derive a representation of granular structures from the representation of granules.

Processes of granular computing may be broadly divided into the two classes of granulation and computation with granules [19, 55]. Granulation involves the construction of the building blocks and structures, namely, granules, levels, and hierarchies. Many issues are involved in granulation, including granulation criteria, granulation algorithms, and characterization of both granules and granular structures. Computation processes systematically explore the granular structures. This involves two-way communications up and down in a hierarchy, as well as switching between levels. For these tasks, we can define mappings connecting granules and levels, modes of granularity conversion, and operators of computing. For the consistency of computation at different levels, we need to study the issues of consistency preservation in terms of invariant properties.

Computation at a certain level may produce an approximate, a partial, or a schematic solution. Such a solution may be made more precise, more complete, or more detailed at another level. This suggests that granular computing is a stepwise refinement process that has been successfully applied in structured programming [51].

Information processing at multiple levels serves the practical needs of an approximate solution within a tolerance range in order to gain in efficiency, low costs, or understandability. This trade-off is essential for solving many real-world problems. The hierarchical way of problem solving makes it easy to find the right level of approximation.

17.5 A Unified Framework

On the basis of the three perspectives, we are ready to develop a unified framework of granular computing and to consider its specific issues. A real-world problem consists of a web of interacting and interrelated granules. For effective problem solving, granular computing must extract easily understandable structures to approximately represent the problem. This results in an understanding in terms of multiple hierarchies and multiple levels in each hierarchy.

17.5.1 *The Granular Computing Triangle*

In the earlier papers [26–29, 55], we suggest a linear dependency between the three perspectives. The philosophy of granular computing guides its methodologies and the methodologies are used to implement information-processing systems. A further investigation convinced us that such an ordering may be inappropriate and perhaps misleading. The three distinctive perspectives mutually depend on each other. This new understanding leads to a triarchic view of granular computing.

The three perspectives are distinct because they deal with issues of different categories and nature. Since each perspective can be further divided, one may study the divisions again from the three perspectives. The three perspectives are thus interrelated and form an integrated whole. Instead of putting them in a linear ordering, we can represent them as three points in a triangle called the granular computing triangle. In this way, any perspective is related to the other two.

In the unified triarchic framework of granular computing, the three perspectives are closely tied together by their common focus on granular structures. The exploration of structures defined by multilevel granularity makes granular computing a promising field of study. The three perspectives represent the basic angles of such an exploration. This suggests another interpretation of the three perspectives. If they are understood as a basis of a three-dimensional space, each point in the space represents a type of study with a specific emphasis on the three perspectives.

The three-dimensional space interpretation enables us to review and compare the existing studies. It is not surprising to observe that the majority of studies have a strong bias to methodological and computational perspectives, and even more so to the computational perspective. The unified triarchic framework stresses the fact that each perspective significantly contributes to the understanding of granular computing. None of them can be overlooked.

The unified conceptual framework considers more abstract discipline-independent principles. It stresses the flexibility and universal applicability of granular computing. As a consequence, granular

computing is applicable to solve a wide range of problems. For example, we may apply the principles of granular computing to the study of the subject of ‘granular computing.’ The result is a multilevel and multiview understanding. Our elaboration and description of granular computing in this chapter is, in fact, one such example.

17.5.2 The Web of Granules

A basic task of granular computing is to build a hierarchical model for a complex system or problem. The basic ingredients of granular computing are granules, a web of granules, and granular structures.

Granules

A complex problem consists of interconnected and interacting parts. Each part, in turn, consists of other parts. Intuitively, each part or a group of parts may be considered as a granule. We therefore have a web of granules as a representation of the problem under consideration. While granules provide local descriptions, the web of granules gives a complete picture of the problem.

We treat granules as a primitive notion of granular computing. From it, other notions can be derived. Furthermore, granules are an abstract notion. The physical meaning of granules can be made clear only when a particular application is considered. In modeling complex systems, granules may be the components of systems at different levels [30, 31]. In programming, granules may be various modules of a software system [50]. In the theory of small groups, granules are small groups [38]. In any organization, granules may be various divisions and departments at different levels. Abstracting from all these concrete examples, a granule can be considered as a focal point of our interest at a certain stage in problem solving. Granules may correspond to either real-world objects or their abstractions.

A granule can be either simple or compound. A simple granule cannot be further decomposed into, or formed by, other granules. A compound granule consists of a group of its interconnected and interacting element granules. (They, in turn, may be simple or compound.) A granule is related to other granules by its dual roles. A granule can be considered as a whole when it is viewed as a part of another granule. A granule is considered to be a group of interconnected and interacting granules when some other granules are viewed as its parts. Consequently, we need to characterize granules by a minimum set of three types of properties. The internal properties of a granule reflect its organizational structures, the relationships, and interaction of its element granules. The external properties of a granule reveal its interaction with other granules. The emergent properties of the granule may be viewed as one type of external property. In many cases, both the internal and external properties are not static, but change with its environment. The contextual properties of a granule show its relative existence in a particular environment. The three types of properties together provide us a full understanding of the notion of a granule.

Granular Structures

A complex problem is represented as a web of granules, in which a granule may itself be a web of smaller granules. With such a representation, an important issue is to study various granular structures embedded in this web. These structures are crucial to an understanding of the problem at different levels.

The types of granular structures are related to the earlier-discussed roles of granules and properties of granules. We consider three levels of granular structures [28, 29]. The granule structures represent the internal structures of a compound granule. The structures are determined by its element granules in terms of their composition, organization, and relationships. Each element granule is simply considered as a single point when studying the structure of a compound granule. From the viewpoint of element granules, the internal structures of a compound granule are indeed their collective structures. In general, one can study the collective structures of a family of granules. Each granule in the family captures and represents a particular and local aspect of the problem. Collectively, they represent the entire problem at a level of granularity defined by the granules in the family. One may use many families of granules to examine the problem at multiple levels of granularity. Structurally, the multiple levels form a hierarchy.

17.5.3 Multilevel and Multiview

With the introduction of granular structures, a problem is understood in terms of granules, levels, and hierarchies. Specifically, a level is made up of a family of granules and a hierarchy is made up of multiple such levels. This not only makes a complex problem more easily understandable, but also leads to efficient, although perhaps approximate, solutions.

Our discussion so far is based on the assumption that a complex problem can be expressed as a web of granules. A fundamental question regarding how those granules are formulated in the first place is not discussed. In fact, a complex problem in the real world is a web in which everything is connected to everything else [30]. The formation of granules is related to the notion of approximations and loose coupling of parts [30, 36]. In forming a granule, one may ignore the subtle differences between its elements and between their individual connections to others. That is, a group of elements may be treated approximately as a whole when studying their relations to others. Each granule is a focal point of our investigation. As an example, the study of cluster analysis in fact relies on such granulated views. The knowledge obtained based on granules, although approximate, may be good enough for practical uses.

In building a hierarchical structure, we need to have a vertical separation of levels and a horizontal separation of granules at the same hierarchical level. Like the formation of individual granules, these separations explore the property of loose coupling of parts. The multiple hierarchical structure thus provides a practical model of a near-decomposable problem. The relationship between levels can be interpreted in terms of abstraction, control, complexity, detail, resolution, and so on.

Granular computing searches for a multilevel hierarchical view of a problem based on near decomposability. Some useful information may be lost with such a hierarchy instead of a web. However, we gain in a model that is easier to understand, tractable, and economic.

A hierarchy represents the results of a study of a problem from one particular angle or point of view. For the same problem, many interpretations and descriptions may coexist [59, 60]. It may be necessary to construct and compare multiple hierarchies [39]. A comparative study of these hierarchies may provide a complete understanding of the problem.

The conceptualization of a problem through multiple hierarchies (i.e., multiview) and multilevel in each hierarchy is general and flexible. Such an approach has been widely used in the investigations of many branches of science. Each hierarchy represents one view of the problem. One may either focus on a particular view or compare various views. The latter requires the connections between different views. More important, emergent properties of a family of views may be observed, which are absent in any specific view.

17.6 Concluding Remarks

Our study of granular computing can be considered to be both old and new. It is old in the sense that the basic ideas and principles of granular computing have appeared time and again in many branches of science and fields of computer science. It is new in the sense that we attempt to extract a set of abstract discipline-independent ideas and principles, and in many cases make them explicit, to arrive at a unified triadic framework of granular computing.

The framework is based on three perspectives. The philosophical perspective focuses on structured thinking, the methodological perspective on structured problem solving, and the computational perspective on structured information processing. The exploration of hierarchical structures at multilevel granularity is the foundation of granular computing.

The field of granular computing is taking shape and will be immensely important. Its success can be predicted from the following observations. Scientists working at different disciplines deal with different subject matters. However, their research processes and methodologies are remarkably similar at a higher level [61]. What distinguishes scientists is their ways of thinking, rather than the subject matter. In general, human problem-solving methodologies and skills share high-level similarities, independent of the problems being solved. The unified framework of granular computing is based on the extraction of

such high-level ideas and principles. Although specific components may be refined or even be entirely changed, this framework may guide research of granular computing in the right direction.

In this chapter, we focus on a high-level examination of granular computing as a new field of study in its own right. The basic ideas and principles are discussed in more abstract terms. Instead of giving more detailed examples to illustrate them, we provide an extensive list of references. One can easily find detailed discussions about the specific applications of these ideas and principles from the references. This high-level investigation enables us to understand granular computing as a new theory for its full potential power without distracted by minute details. The next logical steps, and a real challenge, are to study granular computing at lower levels, with many examples to illustrate its basic ideas and principles.

References

- [1] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer, Boston, 2002.
- [2] M. Inuiguchi, S. Hirano, and S. Tsumoto (eds). *Rough Set Theory and Granular Computing*. Springer, Berlin, 2003.
- [3] T.Y. Lin. Granular computing. Announcement of the BISC Special Interest Group on Granular Computing, 1997.
- [4] T.Y. Lin, Y.Y. Yao, and L.A. Zadeh (eds). *Data Mining. Rough Sets and Granular Computing*. Physica-Verlag, Heidelberg, 2002.
- [5] W. Pedrycz (ed.). *Granular Computing: An Emerging Paradigm*. Physica-Verlag, Heidelberg, 2001.
- [6] L.A. Zadeh. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.
- [7] L.A. Zadeh. Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. *Soft Comput.* 2 (1998) 23–25.
- [8] A. Bargiela and W. Pedrycz. Granular mappings. *IEEE Trans. Syst. Man Cybern. Part A* 35 (2005) 292–297.
- [9] Z. Pawlak. Granularity of knowledge, indiscernibility and rough sets. In: *Proceedings of the 1998 IEEE International Conference on Fuzzy Systems*, Anchorage, AK, May 4–9, 1998, pp. 106–110.
- [10] W. Pedrycz. Granular computing with shadowed sets. In: *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, RSFDGrC'05*, LNAI 3641. Springer, Berlin, 2005, pp. 23–32.
- [11] W. Pedrycz and A. Bargiela. Granular clustering: A granular signature of data. *IEEE Trans. Syst. Man Cybern. Part B* 32 (2002) 212–224.
- [12] J.F. Peters, Z. Pawlak and A. Skowron. A rough set approach to measuring information granules. In: *Proceedings of the International Conference on Computer Software and Applications, COMPSAC'02*, Oxford, England, August 26–29, 2002, pp. 1135–1139.
- [13] L. Polkowski and A. Skowron. Towards adaptive calculus of granules. In: *Proceedings of the 1998 IEEE International Conference on Fuzzy Systems*, Anchorage, AK, May 4–9, 1998, pp. 111–116.
- [14] A. Skowron. Toward intelligent systems: Calculi of information granules. *Bull. Int. Rough Set Soc.* 5 (2001) 9–30.
- [15] A. Skowron and J. Stepaniuk. Towards discovery of information granules. In: *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD'99*, LNCS 1704. Springer, Berlin, 1999, pp. 542–547.
- [16] A. Skowron and J. Stepaniuk. Information granules: Towards foundations of granular computing. *Int. J. Intell. Syst.* 16 (2001) 57–85.
- [17] J.T. Yao and Y.Y. Yao. Induction of classification rules by granular computing. In: *Proceedings of the 3rd International Conference on Rough Sets and Current Trends in Computing, RSCTC'02*, LNAI 2475. Springer, Berlin, 2002, pp. 331–338.
- [18] Y.Y. Yao. Granular computing using neighborhood systems. In: R. Roy, T. Furuhashi, and P.K. Chawdhry (eds). *Advances in Soft Computing: Engineering Design and Manufacturing*. Springer, London, 1999, pp. 539–553.
- [19] Y.Y. Yao. Granular computing: basic issues and possible solutions. In: *Proceedings of the 5th Joint Conference on Information Sciences*, Atlantic City, NJ, February 27–March 3, 2000, pp. 186–189.
- [20] Y.Y. Yao. Information granulation and rough set approximation. *Int. J. Intell. Syst.* 16 (2001) 87–104.
- [21] Y.Y. Yao. Information granulation and approximation in a decision-theoretical model of rough sets. In: S.K. Pal, L. Polkowski, and A. Skowron (eds), *Rough-Neural Computing: Techniques for Computing with Words*. Springer, Berlin, 2003, pp. 491–518.
- [22] Y.Q. Zhang, M.D. Fraser, R.A. Gagliano, and A. Kandel. Granular neural networks for numerical-linguistic data fusion and knowledge discovery. *IEEE Trans. Neural Netw.* 11 (2000) 658–667.

- [23] N. Zhong. Multi-database mining: A granular computing approach. In: *Proceedings of the 5th Joint Conference on Information Sciences*, Atlantic City, NJ, February 27–March 3, 2000, pp. 198–201.
- [24] A. Bargiela and W. Pedrycz. The roots of granular computing. In: *Proceedings of the 2006 IEEE International Conference on Granular Computing*, Atlanta, GA, May 10–12, 2006, pp. 806–809.
- [25] S. Tsumoto, T.Y. Lin, and J.F. Peters. Foundations of data mining via granular and rough computing. In: *Proceedings of the 26th International Conference on Computer Software and Applications, COMPSAC'02*, Oxford, England, August 26–29, 2002, pp. 1123–1125.
- [26] Y.Y. Yao. A partition model of granular computing. *Trans. Rough Sets* 1 (2004) 232–253.
- [27] Y.Y. Yao. Granular computing. *Comput. Sci. (Ji Suan Ji Ke Xue)* 31 (2004) 1–5.
- [28] Y.Y. Yao. Perspectives of granular computing. In: *Proceedings of the 2005 IEEE International Conference on Granular Computing*, Beijing, China, July 25–27, 2005, pp. 85–90.
- [29] Y.Y. Yao. Granular computing for data mining. In: *Proceedings of the SPIE Conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security*, Kissimmee, FL, April 17–18, 2006, pp. 1–12, paper no. 624105.
- [30] F. Capra. *The Web of Life*. Anchor Books, New York, 1997.
- [31] F. Capra. *The Hidden Connections: A Science for Sustainable Living*. Anchor Books, New York, 2002.
- [32] E. Laszlo. *The Systems View of the World: The Natural Philosophy of the New Developments in the Science*. George Brasiller, New York, 1972.
- [33] V. Ahl and T.F.H. Allen. *Hierarchy Theory, a Vision, Vocabulary and Epistemology*. Columbia University Press, New York, 1996.
- [34] H.H. Pattee (ed.). *Hierarchy Theory, the Challenge of Complex Systems*. George Braziller, New York, 1973.
- [35] S.N. Salthe. *Evolving Hierarchical Systems. Their Structure and Representation*. Columbia University Press, New York, 1985.
- [36] H.A. Simon. The organization of complex systems. In: H.H. Pattee (ed.), *Hierarchy Theory, The Challenge of Complex Systems*. George Braziller, New York, 1963, pp. 1–27.
- [37] L.L. Whyte, A.G. Wilson, and D. Wilson (eds). *Hierarchical Structures*. American Elsevier Publishing, New York, 1969.
- [38] H. Arrow, J.E. McGrath, and J.L. Berdahl. *Small Groups as Complex Systems: Formation, Coordination, Development, and Applications*. Sage Publications, Thousand Oaks, CA, 2000.
- [39] V. Jeffries and H.E. Ransford. *Social Stratification: A Multiple Hierarchy Approach*. Allyn and Bacon Inc., Boston, 1980.
- [40] F. Giunchiglia and T. Walsh. A theory of abstraction. *Artif. Intell.* 56 (1992) 323–390.
- [41] J. Hawkins and S. Blakeslee. *On Intelligence*. Henry Holt and Company, New York, 2004.
- [42] J.R. Hobbs. Granularity. In: *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, Los Angeles, CA, May 4–9, 1985, pp. 432–435.
- [43] C.A. Knoblock. *Generating Abstraction Hierarchies: An Automated Approach to Reducing Search in Planning*. Kluwer, Boston, 1993.
- [44] B. Zhang and L. Zhang. *Theory and Applications of Problem Solving*. North-Holland, Amsterdam, 1992.
- [45] A. Newell and H.A. Simon. *Human Problem Solving*. Prentice Hall, Englewood Cliffs, NJ, 1972.
- [46] C.M. Conway and M.H. Christiansen. Sequential learning in non-human primates. *Trends in Cognit. Sci.* 12 (2001) 539–546.
- [47] T. Poggio and S. Smale. The mathematics of learning: Dealing with data. *Not. AMS* 50 (2003) 537–544.
- [48] O.-J. Dahl, E.W. Dijkstra, and C.A.R. Hoare. *Structured Programming*. Academic Press, New York, 1972.
- [49] D.E. Knuth. Structured programming with go to statements. *Comput. Surv.* 6 (1974) 261–301.
- [50] H.F. Ledgard, J.F. Gueras, and P.A. Nagin. *PASCAL with Style: Programming Proverbs*. Hayden Book Company, Rochelle Park, NJ, 1979.
- [51] N. Wirth. Program development by stepwise refinement. *Commun. ACM* 14 (1971) 221–227.
- [52] M. Sipser. *Introduction to the Theory of Computation*. 2nd ed. Thomson Course Technology, Boston, MA, 2006.
- [53] D. Klahr and K. Kotovsky (eds). *Complex Information Processing: The Impact of Herbert A. Simon*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1989.
- [54] D. Marr. *Vision, a Computational Investigation into Human Representation and Processing of Visual Information*. W.H. Freeman and Company, San Francisco, 1982.
- [55] Y.Y. Yao. Three perspectives of granular computing. *J. Nanchang Inst. Technol* 25 (2006) 16–21.
- [56] A. Skowron and P. Synak. Hierarchical information maps. In: *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. RSFDGrC'05*, LNAI 3641. Springer, Berlin, 2005, pp. 622–631.
- [57] Z. Pawlak. *Rough Sets, Theoretical Aspects of Reasoning about Data*. Kluwer, Dordrecht, 1991.

- [58] L. Zhang and B. Zhang. The quotient space theory of problem solving. *Fundam. Inf.* 59 (2004) 287–298.
- [59] G. Bateson. *Mind and Nature: A Necessary Unit*. E.P. Button, New York, 1979.
- [60] Y.H. Chen and Y.Y. Yao. Multiview intelligent data analysis based on granular computing. In: *Proceedings of the 2006 IEEE International Conference on Granular Computing*, Atlanta, GA, May 10–12, 2006, pp. 281–286.
- [61] R.C. Martella, R. Nelson, and N.E. Marchard-Martella. *Research Methods: Learning to Become a Critical Research Consumer*. Allyn and Bacon, Boston, 1999.

18

Quotient Spaces and Granular Computing

Ling Zhang and Bo Zhang

18.1 Granulation Problem

Granulation is one of the most common phenomena in the world. On the one hand, any system (or organization) in the world, either natural or artificial, has a multigranular structure. Human brain is just a multigranular system in nature [1]. In general, any artifact is multilevel structured. For example, a country is a hierarchical organization. It is made up of state, city, town, and so on. These are called structural granulation. They are the granulation in real world. On the other hand, human always conceptualizes the world at different granularities and deals with it hierarchically. This is the granulation in human cognition or information granulation. There is a certain connection between these two granulations. It is possible that the multigranular structure of real world affects the way of human thinking and problem solving. Granular computing is intended to investigate the granulation both in human cognition and in real world. Although we will focus on human cognition aspect of granular computing, the idea we propose can be extended to investigating the real world as well. Therefore, the notions of world, system, and problem will be used alternately in the following discussion.

We present a mathematical model (quotient space model) of granulation that is used to analyze both the human cognition and the real world, especially human problem-solving behaviors. The model was developed in order to represent granules and compute with them easily [2, 3]. The motivation of our research works was based on the belief that one of the basic characteristics in human cognition is the ability to conceptualize the world at different granularities and translate from one abstraction level to the others easily, i.e., deal with them hierarchically [4]. The computers are generally capable of dealing with a problem in only one abstraction level so far. In order to endow the computers with the human intelligence, it is necessary to formalize the granulation problem.

In our model, a system (or problem) is described by a triple (X, F, f) . Universe (or domain) X represents the whole objects of the system that we intended to deal with. It may be a point set and its power may either be finite or infinite. F is the structure of X and represents the relationship among objects of X . Structure F may have different forms but will be described by a topology on X in this chapter. Attribute f is a set of functions defined on universe X and $f : X \rightarrow Y$ may be multicomponent, such as $f = (f_1, f_2, \dots, f_n)$, $f_i : X \rightarrow Y_i$, where Y_i is either a set of real numbers or an other kind of set. Triple (X, F, f) is called a problem space (or simply space). For example, to survey a country, we confront with a problem called (X, F, f) . If each ‘town’ of the country is the finest grain-size element

that we intended to deal with, then universe X represents the whole towns of the country. Attribute $f(x) = (f_1(x), f_2(x), \dots, f_n(x))$ is defined on each town x ; e.g., $f_1(x)$ is the population of town x , $f_2(x)$ the area of x , $f_3(x)$ the GDP of x , and so on. Structure F may represent the geometric, economic, or communication connection among the towns, F may be a network (map) linked by these connections, and then (X, F) can be regarded as a point set in Euclidean plane. The problem (X, F, f) is a survey of the country from its towns. It is noted that the structure F play an important role in our model.

Suppose that X represents the universe composed by the finest grain-size elements. When we view the same universe X at a coarser grain size, we have a coarse-grained universe denoted by $[X]$. Then we have a new problem space $([X], [F], [f])$, where $[F]$ is the corresponding quotient structure and $[f]$ the quotient attributes. Taking the same example above and considering each ‘state’ as a new element of the country, we have a new (quotient) universe $[X]$ composed by the whole states of the country; each state consists of a set of towns. Then $[X]$ is a coarser grain-size world. The corresponding quotient structure $[F]$ represents the geometric, economic, or communication relationship among states and is a simplified country network (map). Quotient attribute $[f]$ represents the population, area, and GDP of each state $[x]$. Then, problem $([X], [F], [f])$ is a survey of the country from its states, the elements with coarser grain size.

The coarser universe $[X]$ can be defined in many ways. Here we define the $[X]$ by an equivalence relation R on X . Then $[X]$ consists of the whole equivalence classes obtained by R ; each equivalence class in X is regarded as an element in $[X]$. The coarse space $([X], [F], [f])$ is called a quotient space of space (X, F, f) .

Assume \mathfrak{R} is a family of equivalence relations on X . Define an order relation ‘ $<$ ’ on \mathfrak{R} as follows. Assume $R_1, R_2 \in \mathfrak{R}$; then, $R_2 < R_1$ if and only if for each pair $x, y \in X$, if $x R_1 y$, then $x R_2 y$, where $x R y$ indicates that x and y are R -equivalent. It implies that the universe X_1 corresponding to R_1 is finer than the universe X_2 corresponding to R_2 . So the family of quotient spaces defined by \mathfrak{R} is a proper mathematical model of granulation in cognition.

There have been several models to deal with the granulation problem, such as fuzzy set [5], rough set [6], and others [7–9]. These models and our quotient space model are related, but each has its own characteristics. Both in the rough set approach and in quotient space models, the universes X with different grain sizes are similarly defined by equivalence relations R . For example, in rough set, a problem is represented by a pair (U, A) , $\{I_a, a \in A\}$, $\{V_a, a \in A\}$, where U denotes universe, A is a set of attributes, I_a attribute function, and V_a the domain (discrete value) of attribute a . Universe U is generally partitioned by the combination of I_a corresponding to a certain equivalence relation. If the value of each attribute function $I_a(x)$ is mapped into the real interval $[0, 1]$, i.e., $0 \leq I_a(x) \leq 1$, then $I_a(x)$ can be regarded as a fuzzy membership function on X , i.e., a fuzzy set in U . If given a family $\{A_i\}$ of fuzzy sets on X , each fuzzy set $A : \mu_A(x)$, letting $\{A(\lambda) = \{x | \mu_A(x) > \lambda\}, 0 \leq \lambda \leq 1\}$ be a family of open sets, then we have a collection of families of open sets from $\{A_i\}$. From the collection, a unique topological structure F on X can be constructed. Then we have a topological space (X, F) and the quotient space approach can be used to deal with it. On the contrary, given a topological space (X, F) , $\forall x \in X$, letting $U(x)$ be all open sets containing x , $U(x)$ is called a neighborhood system and can be regarded as a qualitative fuzzy set [10]. Therefore, a topological space is transformed into a collection of families of fuzzy sets.

The main characteristic of our model is the introduction of spatial structure F (or the interdependency among elements) of universe X into the problem model explicitly, i.e., the triplet (X, F, f) instead of a couple (X, f) . We will show below that the introduction of spatial structure F plays an important role in revealing the relationships among problem spaces with different grain sizes.

Because the quotient spaces have a set of favorable structural properties they become a suitable model of granulation. We will discuss in Sections 18.2 and 18.3.

18.2 The Completeness of the Quotient Space Structure

As mentioned earlier, the multigranular world either in human cognition or in real world can be represented by a family of quotient spaces. In order to reveal the relationships among quotient spaces with different

grain sizes, it is necessary to analyze the structure of the family of quotient spaces; i.e.,

$$\{([X]_n, [F]_n, [f]_n), ([X]_{n-1}, [F]_{n-1}, [f]_{n-1}), \dots, ([X]_1, [F]_1, [f]_1), (X, F, f)\},$$

where (X, F, f) and $([X]_n, [F]_n, [f]_n)$ are the finest and coarsest spaces, respectively.

First, consider the family $\{[X]_n, [X]_{n-1}, \dots, [X]_1, X\}$ of quotient sets, the structure of the multigranular quotient sets. We have the following proposition.

Let \mathfrak{R} be all equivalence relations defined on X .

Definition 1. Assume that $R_1, R_2 \in \mathfrak{R}$, $x, y \in X$, $xR_1y \Rightarrow xR_2y$, then R_1 is called finer than R_2 , denoted by $R_2 < R_1$.

Given a topological space (X, F) and let \mathfrak{R} be all equivalence relations defined on X . We now investigate the structure of the family of topological quotient spaces constructed from space (X, F) .

Let \mathbf{A} be the set of all possible quotient sets on X and $X_i \in \mathbf{A}$. Define a quotient space $(X_i, [F_i])$, where $[F_i]$ is a quotient topology on X_i induced from topological space (X, F) . Let

$U = \{(X_i, [F_i]), X_i \in \mathbf{A}, [F_i] \in \mathbf{N}\}$, where $[F_i]$ is an induced topology of X_i and \mathbf{N} a set of all quotient topologies induced from quotient sets.

It is noted that the quotient topology $[F_i]$ can be defined in many ways; $[F_i]$ is defined as discussed below.

$$[F_i] = \{u \mid p_i^{-1}(u) \in F, u \subset [X_i]\}, \quad \text{where } p_i : X \rightarrow [X_i] \text{ is a nature projection.}$$

That is, we define a set u in $[X_i]$ as open when its corresponding set in X is open.

Definition 2. Assume $(X_i, [F_i]), (X_j, [F_j]) \in U$. If $X_i < X_j$, we define $(X_i, [F_i]) < (X_j, [F_j])$.

Note that any quotient topology $[F_i]$ is a topology on X as well. It is a family of subsets on X . We may use the set inclusion relation among the collection of families of subsets to define the semiorder relation among quotient topologies. Therefore, $[F_i] < [F_j]$ can be defined as $[F_i] \subset [F_j]$. If $X_i < X_j$, then $[F_i] < [F_j]$.

Theorem 1. U is a complete lattice under the order relation ' $<$ ' defined in Definition 2.

It is noted that if the quotient topology $[F_i]$ is not induced from quotient sets, the theorem does not necessarily hold. It means that there exists a subset of U , which does not have the supremum (or infimum) on U . For example,

Example 1. Letting $X = \{1, 2, 3, 4\}$, $F = \{\emptyset, (1, 2), (3, 4), X\}$, we have a topological space (X, F) . And assume $X_1 = \{(1), (2, 3, 4)\}$ and $X_2 = \{2, (1, 3, 4)\}$.

On the one hand, we have quotient topologies $[F_1] = \{\emptyset, X\} = [F_2]$ induced from X_1 and X_2 , respectively. Thus, their supremum is $F' = [F_1]$.

On the other hand, the supremum of X_1 and X_2 is $X_3 = \{\{1\}, \{2\}, (3, 4)\}$. Its induced quotient topology is $[F_3] = \{\emptyset, \{1, 2\}, \{3, 4\}, X\}$. But $[F_3] \neq F'$.

Definition 3. Assume a topological space (X, F) . If F_1 is a topology on X such that $F_1 < F$, then F_1 is called an infratopology of F . Let \mathbf{M} be a set of all infratopologies of F .

Definition 4. Let $W = (\mathbf{A}, \mathbf{M}) = \{([X], F[X]) \mid [X] \in \mathbf{A}, F([X]) < [F], F([X]) \in \mathbf{M}\}$, where $[F]$ is a quotient topology on $[X]$ induced from (X, F) . Define a semiorder relation ' $<$ ' on $W = (\mathbf{A}, \mathbf{M})$ as follows. For $(X_1, F_1), (X_2, F_2) \in (\mathbf{A}, \mathbf{M})$ if $X_1 < X_2$ and $F_1 < F_2$, then $(X_1, F_1) < (X_2, F_2)$.

There are two lemmas [11]:

Lemma 1.. Assume $\forall \alpha \in I$, $f_\alpha : (X_\alpha, F_\alpha) \rightarrow Y$. There exists a maximal (finest) topology on Y among all topologies that make each f_α continuous.

Lemma 2.. Assume $\forall \alpha \in I, f_\alpha : X \rightarrow (Y_\alpha, F_\alpha)$. There exists a minimal (coarsest) topology on X among all topologies that make each f_α continuous.

Definition 5. Given $X_\alpha \in A$. Let

$$F(X_\alpha) = \{F' \mid \exists \{(X_i, F_i), i \in I\} \subset U, F' = \sup\{F_i\}, X_\alpha = \sup\{X_i\} \text{ or } F' = \inf\{F_i\}, X_\alpha = \inf\{X_i\}\}.$$

Define $V = \{(X_\alpha, F_\alpha) \mid X_\alpha \in X, F_\alpha \in F(X_\alpha)\}$.

V is the whole topological spaces obtained from the supremum (or infimum) operation on subsets of U .

Definition 6. Let $A = \{(X_\alpha, F_\alpha), \alpha \in I\}$ be a subset of W , X' be the supremum of $\{X_\alpha\}$. Construct a map $p_\alpha : X' \rightarrow (X_\alpha, F_\alpha)$, $\alpha \in I$, where p_α is a natural projection. From Lemma 2, we have a minimal topology F' on X' among all topologies that make all maps continuous. Define (X', F') as the supremum quotient space of A .

Definition 7. Let $A = \{(X_\alpha, F_\alpha), \alpha \in I\}$ be a subset of W and \underline{X} be the infimum of $\{X_\alpha\}$. Construct a map $p_\alpha : (X_\alpha, F_\alpha) \rightarrow X$, $\alpha \in I$, where p_α is a natural projection. From Lemma 1, we have a maximal topology F on \underline{X} among all topologies that make all maps continuous. Let (X, F) be the infimum quotient space of A .

Similarly, for a subset of V , we can define the corresponding supremum (or infimum) quotient space as Definitions 6 and 7.

Theorem 2. W (or V) forms a complete lattice under the supremum and infimum operations defined in Definitions 6 and 7.

Proof. Given a subset $A = \{(X_\alpha, F_\alpha), \alpha \in I\}$ of W . Let X^1 be the supremum of $\{X_\alpha\}$. Construct a map $p_\alpha : X^1 \rightarrow (X_\alpha, T_\alpha)$, $\alpha \in I$, where p_α is a natural projection. From Lemma 2, we have a coarsest topology F' on X^1 such that all maps become continuous. Since p_α is continuous, $(p_\alpha)^{-1}$ is an open map; i.e., all open sets on $\forall \alpha \in I, (X_\alpha, F_\alpha)$ are mapped onto open sets on (X^1, F') . From Definition 3, we obtain $\forall \alpha \in I, F_\alpha < F'$. F' is the upper bound of $\{F_\alpha, \alpha \in I\}$. Since F' is a coarsest topology, $(X, F') \in W$ is the supremum of A . ■

Similarly, given a subset B , there exists the infimum. Therefore, W is a complete lattice.

Similarly, we can prove that V is a complete lattice as well.

18.3 The Characteristics of the Quotient Space Structure

We have constructed three families of quotient spaces, i.e., U , V , and W ; each forms a complete lattice. Obviously, V is a sublattice of W and a complete lattice constructed from the elements of U via a set of the supremum and infimum operations. W is a complete lattice constructed from the elements of U and the elements formed by all infratopologies of every induced quotient topologies.

The complete lattice U is a basic quotient space structure in quotient space theory. From space (X, F) , all quotient spaces in U can be constructed by using the induced quotient topology. On the contrary, all quotient spaces constructed by the induced quotient topology must be a member of U . This implies that the operation constructing quotient spaces by using the induced quotient topology is closed in U . The completeness of U underlies the construction of quotient spaces via the induced quotient topology approach.

The ‘combination’ is one of main operations in quotient space theory. V is the maximal complete lattice that can be constructed from U by the combination operation, i.e., the supremum and infimum operations based on the induced quotient topology. This will guarantee the completeness of the family of quotient spaces constructed by the combination operation. Therefore, the supremum (infimum) operation

on quotient topology is closed in V . The V constructed from U above is similar to the topology constructed from the topological bases. Finally, through the supremum and infimum operations above, the complete lattice W is obtained from all possible quotient spaces constructed from U both by the granulation of the universe X and by the granulation of the topology F . So both topological granular operation and supremum (infimum) operation are closed in lattice W .

We now discuss the relationships among the three lattices V , U , and W .

Definition 8. Given a topological space (X, F) and an open set B . If there exists a non-empty bipartition $B = B_1 \cup B_2$ and one of them is open at most, then B is called minimal open set.

Lemma 3.. Given an open set A . Set A is either a discrete topology or a minimal open set.

Proof. Assume that A is not a minimal open set, $\forall x \in A$, and construct a non-empty bipartition $A = \{x\} \cup (A/\{x\})$. Any singleton $\{x\}$ in A is open. Then A is a discrete topology. ■

It is obvious that $U \subset V \subset W$. And we will prove that they are proper subset relations generally.

Theorem 3. Given a topological space (X, F) . The necessary and sufficient conditions of $U = V$ are the following:

1. $|X| < 3$.
2. F is a trivial topology.
3. F is a discrete topology.

Proof. If $|X| < 3$, obviously $U = V$.

\Leftarrow : Since (X, F) is a trivial (discrete) topology, for any quotient topology in quotient set $[X]$ is trivial (or discrete). Therefore, $U = V$.

\Rightarrow Reduction to absurdity: Assume that F is neither trivial nor discrete; we show below that $V \neq U$.

Let $F = (B_\alpha, \alpha \in I)$. From Lemma 3, for each B_α , if there exists a non-empty bipartition and only one of them is open at most (the minimal open set), then keep B_α . Otherwise, partition B_α as the union of singletons (open sets).

After the above treatment, F is denoted by $\{B_\alpha\}$ as well, with each B_α either a singleton or a minimal open set. If there is only one B_α in F , then F is a trivial topology. There is a contradiction. If every B_α is a singleton, then F is a discrete topology. There is a contradiction too.

Therefore, there exist some B_α 's such that $|B_\alpha| > 1$. Let B_1 be the one with the minimal cardinality among B_α .

There are two cases:

1. If $B_1 = X$ and any proper subset of X is not a minimal open set, then $B_i, i \neq 1$, must be a singleton. Let $C = \cup B_j, j \neq 1, D = X/C$. Since $|X| > 2$, the number of elements in one of sets C and D must be > 1 . If $|C| > 1$, partition C into $C = C_1 \cup C_2, C_i \neq \emptyset, i = 1, 2$. Let $X_1 = \{C_1, C_2 \cup D\}$ and $X_2 = \{C, D\}$. Set D is not open; otherwise from Lemma 3, D is a singleton. There is a contradiction. Set $C_2 \cup D$ is not open too; otherwise $C_2 \cup D$ is a minimal open set. There is a contradiction too.

We have the induced quotient topologies $[F_1] = \{\emptyset, C_1, X\}$ and $[F_2] = \{\emptyset, C, X\}$ and its supremum $F' = \{\emptyset, C_1, C, X\}$.

On the other hand, $X_3 = \{C_1, C_2, D\}$ is the supremum of X_1 and X_2 . The quotient topology induced from X_3 is $[F_3] = \{\emptyset, C_1, C_2, C, X\}$ and $F_3 \neq F'$. Therefore, $U \neq V$. There is a contradiction.

If C is a singleton, there exists a non-empty bipartition $X/C = \{A_1, A_2\}$ of X/C . Construct $X_1 = (C \cup A_1, A_2)$ and $X_2 = (C \cup A_2, A_1)$. Both A_i and $C \cup A_i, i = 1, 2$, are not open sets. Otherwise, assume A_1 is an open set; from Lemma 3, A_1 is either a discrete topology or a minimal open set. Since C is a singleton, A_1 must not be a discrete topology. If A_1 is a minimal open set, but a

proper subset of X , there is a contradiction. Therefore, each $A_i, i = 1, 2$, is not an open set. Similarly, $C \cup A_i, i = 1, 2$, is not open either.

We have $[F_1] = [\emptyset, X]$ and $[F_2] = [\emptyset, X]$. Then $F' = [\emptyset, X]$.

On the other hand, $X_3 = \{A_1, A_2, C\}$, we have $F_3 = [\emptyset, C, X]$. Then $[F_3] \neq F'$, $U \neq V$. There is a contradiction.

2. $B \neq X, |B| > 1$. Let $B = A_1 \cup A_2, A_i \neq \emptyset, i = 1, 2$. At least one A_i of them is not an open set. Construct a partition $X_1 = \{A_1, X/A_1\}, X_2 = \{A_2, X/A_2\}$.

2.1. If both A_1 and A_2 are not open, the quotient topologies induced from X_1 and X_2 are $[F_1] = [F_2] = [\emptyset, X]$. The supremum of $[F_1]$ and $[F_2]$ is $F' = [\emptyset, X]$.

On the other hand, the supremum of X_1 and X_2 is $X_3 = \{A_1, A_2, X/(A_1 \cup A_2)\}$. The quotient topology induced from X is $[F_3] = [\emptyset, B, \dots, X]$. Then, $[F_3] \neq F'$.

2.2. If one of $A_i, i = 1, 2$, is open, then assume A_1 is open. X/A_1 is not open, $A_2 = (X/A_1) \cap B$ is open; this is a contradiction. We obtain $[F_1] = [\emptyset, A_1, X]$ and $[F_2] = [\emptyset, X]$ (or $[\emptyset, X/A_2, X]$, when X/A_2 is open). Their supremum is $F' = [\emptyset, A_1, X]$ (or $[\emptyset, A_1, X/A_2, X]$, when X/A_2 is open). The quotient topology induced from $X_3 = \{A_1, A_2, X/(A_1 \cup A_2)\}$ is $[F_3] = [\emptyset, A_1, B, \dots, X]$. Then, $[F_3] \neq F'$.

Finally, we have $U \neq V$. This is a contradiction.

From Lemma 3, we present an example as follows. ■

Example 2. Given $X = \{1, 2, 3, 4\}$, and F is a discrete topology. From Lemma 3, we obtain U and V corresponding to (X, F) and $U = V$. Let

$F_1 = \{\emptyset, \{1\}, \{1, 2\}, \{1, 2, 3\}, X\}$. We have $F_1 < F$. Then $(X, F_1) \in W$ and $(X, F_1) \notin V$. Thus, V is a proper subset of W .

18.4 The Relationships Among Quotient Spaces

Any problem is represented at different granularities in human cognition. Its corresponding mathematical model is a family of quotient spaces; each represents an abstraction level of the original problem. The translation from one abstraction level to the other is a common phenomenon in human cognition. It corresponds to three main operations in the quotient space model, i.e., decomposition, projection, and combination. The decomposition (or partition) operation is the translation from a coarse level to a fine one, as the granulation problem that we have discussed in Section 18.1. The ‘projection’ means the translation from a fine level to a coarse one. By this projection operation, we will have a profile of the problem or a simplified version of the problem. The ‘combination’ means the translation from several coarse levels to a fine one. By this combination operation, we will have more details from several cursory observations. One of the main goals of granular computing or information granulation in human cognition is intended to enhance the efficiency of the translation, therefore, to reduce the ‘complexity’ (computation, communication costs, etc.) of multigranular processing. So it is necessary to investigate the relationships among different quotient spaces. One of the advantages of the quotient space model is to be able to deal with the problem easily.

Solving a problem can generally be transformed into that of finding a connected path from a given starting point to a goal (point) in a given space. For example, robotic path planning can be regarded as finding a collision-free path in a high-dimensional space. Reasoning can be considered as finding a connected path from promise A to conclusion B in a semi-order space as well. The favorable structural property of quotient spaces is one of the underlying advantages of our model. The following are some of the results.

18.4.1 Projection

In human hierarchical problem solving, ‘projection’ is one of the main operations. Assume a topological space (X, F) . By projection $p : (X, F) \rightarrow ([X], [F])$, we construct a simplified (coarse) space from

(X, F) . Since the granule in $[X]$ is larger than that in X , the structure F will be simplified as quotient structure $[F]$ in $[X]$. Therefore, the details of (X, F) are missing in space $([X], [F])$. The goal is that the properties that we are interested in should be preserved in $([X], [F])$ after the simplification.

Assume that R is an equivalence relation on X . From R , we have a quotient set $[X]$. A quotient topology $[F]$ induced from F can be defined as follows:

$$[F] = \{u \mid p^{-1}(u) \in F, u \subset [X]\}$$

In addition, $p : X \rightarrow [X]$ is a natural projection and can be defined as follows:

$$\begin{aligned} p(x) &= [x], \\ p^{-1}(u) &= \{x \mid p(x) \in u\}. \end{aligned}$$

From topology [2], it is known that some properties in topological space (X, F) can be preserved in its quotient space $([X], [F])$. We have the following proposition.

Proposition 1 (Falsity-preserving property). *If there is no solution (connected path) between $[A]$ and $[B]$ in a quotient space $([X], [F])$, then there is no solution (path) between A and B in its original space (X, F) as well.*

Proof. Let $p : (X, F) \rightarrow ([X], [F])$ be a natural projection. p is a continuous mapping. Since there is no solution path between $[A]$ and $[B]$, $[A]$ and $[B]$ do not belong to a connected component of $([X], [F])$. By reduction to absurdity, if there is a solution path from the corresponding A to B , then A and B belong to the same connected component S of (X, F) . Since p is a continuous mapping, the image of a connected set by the mapping is still connected. Then $p(A) = [A]$ and $p(B) = [B]$ belong to the connected component $p(S)$. There is a contradiction. ■

It means that the connectivity of sets is grain-size invariant. This is a very useful property. In fact, in quotient space model the human problem solving (or reasoning) can be treated as finding the connectivity of sets in the problem space. The proposition shows that if there is a solution (connected) path in the original space (X, F) , there exists a solution path in its proper coarse-grained space $([X], [F])$. Conversely, in the coarse-grained space, if there does not exist a solution path, there is no solution in the original space. This is called the ‘falsity-preserving’ property; i.e., ‘no-solution’ (region) property is preserved between quotient spaces. The property underlies the power of human hierarchical problem solving. If at the coarse level we do not find any solution in some regions, then there is no solution in the corresponding regions at the fine level. Therefore, the results obtained from the coarse level can guide the problem solving in the fine level effectively. In general, the coarse space is simpler than the fine one, so the computational complexity will be reduced by the hierarchical problem solving.

Proposition 2 (Truth-preserving property). *If a problem $[A] \rightarrow [B]$ on $([X], [F], [f])$ has a solution path, $\forall [x], p^{-1}([x])$ is connected on X , then the corresponding problem $A \rightarrow B$ has a solution path on (X, F, f) consequentially.*

Proof. Since $[A] \rightarrow [B]$ has a solution path on $([X], [F], [f])$, $[A]$ and $[B]$ fall on the same connected component C . Let $D = p^{-1}(C)$. We show that D is a connected set on X . ■

By reduction to absurdity, assume that D is a union of two disjoint non-empty open closed sets D_1 and D_2 , $\forall a \in C$, and $p^{-1}(a)$ is connected on X . $p^{-1}(a)$ only belongs to one of D_1 and D_2 . Therefore, $D_i, i = 1, 2$, consists of elements of $[X]$; i.e., there exist C_1 and C_2 such that $D_1 = p^{-1}(C_1)$ and $D_2 = p^{-1}(C_2)$. Since $D_i, i = 1, 2$, are open closed sets, from the definition of nature map p , C_1 and C_2 are non-empty open closed sets on $[X]$ either. Since C_1 and C_2 are the partition of C , C is not a connected set. There is a contradiction.

By the proposition, a problem-solving process will be simplified if that problem can be solved in its coarse level. Based on the ‘truth-preserving’ principle, we presented a topological approach for robotic path planning on a geometric space [12]. In the approach, by the homotopically equivalent classification in topology, the geometric space is transformed into a finite network called characteristic network, i.e., a coarse-grained space. Then, the find-path problem in geometrically fine-grained space is transformed into that of finding a connected path in the finite network. So the computational complexity is reduced dramatically.

18.4.2 Combination

However, in human cognition, one usually learns things from local fragments, integrates them, and gradually forms a global picture. It means inferring the fine-level representation from the information collected at coarse levels. The process is called combination (or information fusion).

For a problem space (X, F, f) , given the knowledge of its two quotient spaces (X_1, F_1, f_1) and (X_2, F_2, f_2) , the ‘combination’ operation is intended to have an overall understanding of (X, F, f) from the known knowledge.

Let (X_3, F_3, f_3) be the combination of (X_1, F_1, f_1) and (X_2, F_2, f_2) , and $p_i : (X, F, f) \rightarrow (X_i, F_i, f_i)$, $i = 1, 2$. In order to have a proper (X_3, F_3, f_3) , at least the following three combination principles should be satisfied:

$$\begin{cases} p_i X_3 = X_i, \\ p_i F_3 = F_i, \\ p_i f_3 = f_i, \quad i = 1, 2. \end{cases}$$

In general, the solution satisfying the three principles is not unique. In order to have a unique result, some criteria must be added so that the solution is optimal. First, we propose the above X_3 and F_3 as the combination universe and topology, respectively. Let the combination universe X_3 be the least upper bound of universes X_1 and X_2 . This implies that X_3 is the coarsest one among the universes that satisfy the first combination principle and the finest one that we can get from the known universes X_1 and X_2 . And let the combination topology F_3 be the least upper bound of topologies F_1 and F_2 . This implies that F_3 is the coarsest one among the topologies that satisfy the second combination principle and the finest one that we can get from the known topologies F_1 and F_2 . This is the maximal amount of information that we can obtain from the known knowledge. Therefore, the proposed X_3 and F_3 are optimal in some sense.

In most cases, the optimal criteria are domain dependent. However, here we present a general criterion as follows:

$$D(f_3, f_1, f_2) = \min_f D(f, f_1, f_2) \text{ or } \max_f D(f, f_1, f_2),$$

where f ranges over all attribute functions on X_3 that satisfy the third combination principle.

It is noted that in the combination principle, $p_i f_3 = f_i$, $i = 1, 2$, where p_i may be a non-deterministic mapping. We will discuss the problem in Section 18.5.

To show the rationality of the above combination principles, in [12] we deduced the famous Dempster–Shafer combination rule in belief theory [13] by the principles. It shows that the Dempster–Shafer rule is the outcome of the combination principles under certain optimal criteria.

18.5 The Extension of Quotient Space Model

In the above discussion, we only deal with the crisp granulation issue, where the universe is divided by equivalence relations, i.e., the partition. Certainly, the quotient space model can be extended to fuzzy granulation, consistency relation, and so on. In these cases, either the boundaries among granules are blurry or there is a superposition among granules.

18.5.1 Fuzzy Granulation

There are several ways in which fuzzy set theory can be introduced into quotient space model. Here we discuss some of them.

Fuzzy Sets under Quotient Space Structure

Assume that \tilde{A} is a fuzzy subset on X and its membership function is $\mu_A(x) : X \rightarrow [0, 1]$. How to represent the fuzzy set \tilde{A} on the quotient space $[X]$ of X ? To this end, we first consider the membership function $\mu_A(x)$ as an attribute function f on X , and then we have a problem space (X, f) . In order to construct a coarse space $([X], [f])$, it is needed to define the corresponding fuzzy subset $[\tilde{A}]$ on $[X]$. The induced fuzzy subset $[\tilde{A}]$ of \tilde{A} can be defined as $\mu_{[A]}([x]) : [X] \rightarrow [0, 1]$, where $\mu_{[A]}([x]) = g(\mu_A(x), x \in [x])$, g is a set function from $P[0, 1]$ to $[0, 1]$, and $P[0, 1]$ is a set composed of all subsets on $[0, 1]$. For example, one possible way is

$$\mu_{[A]}([x]) = \max \{\mu_A(x) | x \in [x], [x] \in [X]\}.$$

Then, we have a corresponding (quotient) fuzzy subset $[\tilde{A}]$ on $[X]$ induced from \tilde{A} , and function $\mu_{[A]}([x])$ can be regarded as a quotient attribute function of $\mu_A(x)$. The fuzzy subset $[\tilde{A}]$ is a coarse representation on $[X]$ of fuzzy subset \tilde{A} . Then the fuzzy set (concept) \tilde{A} can be addressed by the quotient space model hierarchically. For example, in multilevel fuzzy reasoning, the ‘falsity-preserving’ property of the membership functions still holds between the quotient spaces. In [14, 15], we have the following result. If the value of the membership function of the fuzzy conclusion (solution, or quotient fuzzy subset) obtained from a coarse grain-size space is less than a , then the value of the membership function of the corresponding fuzzy conclusion (fuzzy subset) made in the fine grain-size space is less than a consequentially. This means that the conclusion made in the coarse abstraction level is blurrier (weaker) than in the fine abstraction level, since there is a lack of information in the coarse level. The property can be used in multilevel fuzzy reasoning to delete some conclusions (regions) with lower value (of membership function).

Fuzzy Equivalence Relations

The quotient space $[X]$ can be defined by a fuzzy equivalence relation \tilde{R} instead of the conventional equivalence relation R . Assume $\tilde{R} \in T(X \times X)$, where $T(X \times X)$ stands for all fuzzy sets on $X \times X$. If it satisfies

1. $\forall x \in X, \tilde{R}(x, x) = 1$,
2. $\forall x, y \in X, \tilde{R}(x, y) = \tilde{R}(y, x)$,
3. $\forall x, y, z \in X, \tilde{R}(x, z) \geq \sup_y (\min(\tilde{R}(x, y), \tilde{R}(y, z)))$,

then \tilde{R} is called a fuzzy equivalence relation on X .

In [16], given a fuzzy equivalence relation \tilde{R} on X , we have a set $R_\lambda = \{(x, y) | \tilde{R}(x, y) \geq \lambda\}, 0 \leq \lambda \leq 1$, of crisp equivalence relations, where R_λ is called a cut relation of \tilde{R} . Contrarily, given a hierarchical structure $\{X(\lambda) | 0 \leq \lambda \leq 1\}$ on X , there exists a fuzzy equivalence relation \tilde{R} on X such that $X(\lambda), \lambda \in [0, 1]$, is a quotient space corresponding to R_λ and R_λ is a cut relation of \tilde{R} . Therefore, a fuzzy equivalence relation is equivalent to a family of (hierarchical) conventional equivalence (cut) relations. So the fuzzy granulation defined by a fuzzy equivalence relation can be handled by a hierarchical quotient space model.

Fuzzy and Crisp

From the above discussion, it is noted that fuzzy and crisp are relative. A concept is crisp in a fine level but might become blurry in the coarse level. For example, subset A defined by function $\mu_A(x) : X \rightarrow \{0, 1\}$ on X is certain. The corresponding subset $[A]$ on $[X]$ defined by $\mu_{[A]}([x]) = \text{mean } \{\mu_A(x) | x \in [x], [x] \in [X]\}$ becomes uncertainty since $\mu_{[A]}([x]) : [X] \rightarrow [0, 1]$. If a problem $A \rightarrow B$ has a solution path on (X, F, f) , the corresponding problem $[A] \rightarrow [B]$ on

$([X], [F], [f])$ may not necessarily have a solution path. There exists some uncertainty (or possibility). This means that a certain problem may become uncertain in some coarse level. Although there are some uncertainties in the coarser level, the coarse representations are more simple and robust. They are not sensitive to the noises (or disturbance). So both coarse and fine representations have their own superiority and can be handled by the quotient space model easily.

18.5.2 Consistency Relation

If R' is a binary relation on X and satisfies self-reflexive and symmetrical properties, R' is called a consistency relation.

Definition 9. $\forall x \in X$ define $\langle x \rangle_{R'} = \{y \in X | x R' y\}$ as an R' -consistency class of x and $\langle X \rangle_{R'} = \{\langle x \rangle_{R'} | x \in X\}$ as the whole of $\langle x \rangle_{R'}$, for simplicity, denoted by $\langle x \rangle$ and $\langle X \rangle$, respectively.

Theorem 4. Assume that $S = \{R'_i | i \in I\}$ is a set of all consistency relations on X , and I is a set of subscripts.

1. Both $\bigcap_{i \in I} R'_i$ and $\bigcup_{i \in I} R'_i$ are consistency relations on X .
2. Define a binary relation on S as follows:

$$\forall i, j \in I, \forall x, y \in X, R'_j \leq R'_i, \text{ iff } x R'_i y \Rightarrow x R'_j y.$$

Then S is a complete lattice under the binary relation ' \leq ' denoted by (S, \leq) . The intersection and union operations on the lattice, i.e., $\hat{\wedge}$ and $\hat{\vee}$, are defined as follows, respectively.

$\forall i, j \in I, R'_i \vee R'_j = R'_i \cup R'_j, R'_i \wedge R'_j = R'_i \cap R'_j$, where \cap and \cup are the conventional intersection and union operations on sets, respectively.

3. $\forall x \in X, \forall J \subseteq I, \langle x \rangle_{\bigcap_{i \in J} R'_i} = \bigcap \{\langle x \rangle_{R'_i} | i \in J\}, \langle x \rangle_{\bigcup_{i \in J} R'_i} = \bigcup \{\langle x \rangle_{R'_i} | i \in J\}.$

The proof of the theorem is trivial.

It is noted that the complete lattice S composed by all consistency relations is similar to the complete lattice R composed by all equivalence relations. Both can be used to represent the world at different granularities. Their different is the following. Based on the partition (equivalence relation) the subsets on X corresponding to any pair of elements on $[X]$ are disjoint, but based on the consistency relation, the corresponding subsets on X are not necessarily pairwise disjoint. There exists some overlapping among the subsets. When the consistency relation satisfies transitivity as well, it becomes the equivalence relation.

Now we discuss the quotient space model based on the consistency relation. Assume (X, F, f) , where F and f are topology and attribute function on X , respectively. $R' \in S$ is a consistency relation.

Definition 10. Assume that $t : X \rightarrow Y$ is a mapping on X . From t , we can induce an equivalence relation \equiv_f by letting $\equiv_f : \forall x_1, x_2 \in X, x_1 \equiv_f x_2 \Leftrightarrow t(x_1) = t(x_2)$, for simplicity, denoted by \equiv . $[x], x \in X$, are equivalence classes corresponding to relation \equiv . $p : X \rightarrow [X]$ is a natural projection and $[X]$ is a quotient set.

Definition 11. Assume that X and Y are topological spaces. Mapping $t : X \rightarrow Y$ is called a quotient mapping if

1. t is a surject mapping.
2. Assuming $A \subseteq Y$, if $t^{-1}(A)$ is an open set on X , then A is an open set on Y .

Accordingly, the topology on Y is called the quotient topology with respect to quotient mapping t .

Proposition 4. Assume $t : (X, F) \rightarrow Y$ is a quotient mapping, where (X, F) and Y are topological spaces. \equiv is an equivalence relation on $[X]$ induced from t . $([X], [F])$ is a quotient topology space corresponding to natural projection $p : X \rightarrow [X]$. Then $([X], [F])$ and Y are homeomorphous, where the homeomorphous mapping $h : [X] \rightarrow Y$ satisfies $h \circ p = t$; i.e., $h^{-1} \circ t = p$.

The proposition shows that when $t : X \rightarrow Y$ is a quotient mapping, Y can be regarded as a quotient space of X and t is a corresponding identification mapping. In fact, the quotient space and the quotient mapping are correlative concepts. Both provide a certain tool for representing the different views of the same world. The natural projection p is a special case of quotient mappings that satisfy the two conditions in definition 11.

Definition 12. Assume that $t : X \rightarrow \langle X \rangle$ is a surject mapping and $\forall x \in X, t(x) = \langle x \rangle$. Define a topology $\langle F \rangle$ on $\langle X \rangle$ such that $\langle F \rangle = \{A \subseteq \langle X \rangle | t^{-1}(A) \in F\}$; i.e., $\langle F \rangle$ is the finest one among the topologies that make the mapping $t : (X, F) \rightarrow ([X], [F])$ continuous.

Proposition 5. Assume that $t : X \rightarrow \langle X \rangle$ is a quotient mapping. Topological spaces $((X), \langle F \rangle)$ and $([X], [F])$ are homeomorphous, where $([X], [F])$ is an identification space induced from t .

The proposition shows that although R' is not an equivalence relation, i.e., $\langle X \rangle$ does not compose a partition of X , due to the homeomorphous relation, the classical quotient space theory can still be extended to the consistency relation. But the elements of $\langle X \rangle$ as the subsets on X are not necessarily pairwise disjointed; the computational complexity would be increased in some cases.

The quotient attribute function $\langle f \rangle$ can be constructed by the approaches provided from the classical quotient space theory. Therefore, given a consistency relation R' and problem space (X, F, f) , we can construct a quotient space $((X), \langle F \rangle, \langle f \rangle)$ corresponding to R' . Many results in classical quotient space theory can be used in the consistency relations due to the homeomorphism between spaces $((X), \langle F \rangle, \langle f \rangle)$ and $([X], [F], [f])$. For example,

Proposition 6. If $U \subseteq X$ is a connected set on X , then $t(U)$ is a connected set on $\langle X \rangle$.

From the proposition, we can have the similar ‘falsity-preserving’ property among the quotient spaces formed by the consistency relations.

18.5.3 The Applications

In [12], we analyzed the top-down hierarchical problem solving by using the quotient space model and had a set of results. For example, we proved the conditions in which the multigranular problem solving can reduce the computational complexity and constructed a set of efficiently hierarchical planning and heuristic search algorithms. Now we will analyze the bottom-up data processing by using the same model.

First, taking the neural network learning as an example, we show that efficient learning algorithms can be obtained from the combination principles in quotient space model.

Learning Problem.. Given a set $K = \{(x^1, y^1), \dots, (x^p, y^p)\}$, $x^i \in R^n$, $y^i \in \{0, 1\}^k$ of training samples, construct a feedforward neural network (a learning machine) N such that $N(x^i) = y^i$, $i = 1, \dots, p$. This implies that the sample set K is classified into k categories by network N . In [16], we showed that by the transformation $f : R^n \rightarrow S^{n+1}$, $f(x) = (x, \sqrt{R^2 - |x|^2})$, an input sample (point), $x^i \in R^n$, in R^n is mapped onto a point on a superspherical surface S^{n+1} . Therefore, the learning problem in space R^n is transformed into the (point set) covering one in space S^{n+1} . By the covering algorithm we proposed, we may have a set $C = \{C_1^1, C_2^1, \dots, C_{n_1}^1, C_1^2, C_2^2, \dots, C_{n_2}^2, \dots, C_1^k, C_2^k, \dots, C_{n_k}^k\}$ of covers, where the subset $C^i = \{C_1^i, C_2^i, \dots, C_{n_i}^i\}$, $i = 1, 2, \dots, k$, of covers overlays all samples of the i th category and C_j^i covers a part of the samples.

The learning problem can be depicted by the quotient space terminology as well. First, we construct a problem (sample) space (X, f) , where universe $X = \{x^1, x^2, \dots, x^p\}$ with p samples, the attribute function f is defined as $y^i = f(x^i)$. After learning (covering), we have a new space $([X], [f])$, where universe $[X] = \{C^1, C^2, \dots, C^k\}$ with k elements, $k = p$, and $C^i = \{C_1^i, C_2^i, \dots, C_{n_i}^i\}$, $i = 1, 2, \dots, k$. The key is how to define the attribution function $[f]$ on $[X]$.

Since element $C^i = \{C_1^i, C_2^i, \dots, C_{n_i}^i\}$, $i = 1, 2, \dots, k$, we regard each cover C_j^i , $j = 1, 2, \dots, n_i$, as a projection of C^i . Then C^i is the combination of the projections. Therefore, the attribute function $[f]$ on C^i can be obtained from the combination of functions $[f]_j$ defined on C_j^i , $j = 1, 2, \dots, n_i$.

For simplicity, let C^i be $C = \{C_1, \dots, C_g\}$. Assume that each sample x^i , $i = 1, 2, \dots, p$, is an identically independent distributed n -dimensional random variable. As a set of random variables, each cover C_j , $j = 1, 2, \dots, g$, in C (C^i) can be described by a probability density function. We choose normal distribution function $N(x, \mu, \Sigma)$ as its probabilistic model, where mean μ may be defined as the center of the cover and Σ as the variance matrix. The normal distribution function $N(x, \mu, \Sigma)$ can also be regarded as the attribute function $[f]_j$ of C_j , $j = 1, 2, \dots, g$. Then the attribute function $[f]$ on C can be defined as the combination of $[f]_j$ as follows:

$$[f] = F(x) = \sum_j \alpha_j N(x, \mu_j, \Sigma_j), \quad j = 1, 2, \dots, g.$$

From the combination principles in quotient space model, the weights α_j can be chosen by some sort of optimization techniques. Since $[f] = F(x)$ is a g -component finite mixture density, the maximum likelihood estimator can be used to estimate the weights. According to the iterative expectation maximization (EM) algorithm presented in [12], we have the following optimization procedures.

Let $K = \{(x^1, y^1), \dots, (x^p, y^p)\}$, $x^i \in R^n$, $y^i \in \{0, 1\}^k$, be a set of samples. And its corresponding covers $C = \{C_1^1, C_2^1, \dots, C_{n_1}^1, C_1^2, C_2^2, \dots, C_{n_2}^2, \dots, C_1^k, C_2^k, \dots, C_{n_k}^k\}$, where $C^i = \{C_1^i, C_2^i, \dots, C_{n_i}^i\}$, $i = 1, 2, \dots, k$, for simplicity, letting $C^i = C = \{C_1, \dots, C_g\}$.

Initialization.. Let $\alpha_j^{(0)} = d_j$, where d_j is the proportion of the number of points in the i th category to the total number of points in cover C_j .

$$\sigma_j^{(0)} = r_j, \quad r_j \text{ is the radius of } C_j.$$

$$\mu_j^{(0)} = a_j, \quad a_j \text{ is the center of } C_j.$$

$$\Sigma_j^{(0)} = (\sigma_j^{(0)})^2 I_n, \text{ where } I_n \text{ is a } n\text{-dimensional unit matrix.}$$

E step:

The $(k + 1)$ -iteration, calculate the posterior probability of sample x^i from the j th component as follows:

$$\beta_{ij}^{(k)} = \beta_j(x^i, \Theta^{(k)}) = \frac{\alpha_j^{(k)} N(x^i, \mu_j^{(k)}, \Sigma_j^{(k)})}{\sum_{j=1}^g \alpha_j^{(k)} N(x^i, \mu_j^{(k)}, \Sigma_j^{(k)})}, \quad j = 1, \dots, g; i = 1, \dots, p, \quad (\text{E-1})$$

$$\alpha_j^{(k+1)} = \frac{1}{p} \sum_{i=1}^p \beta_{ij}^{(k)}, \quad j = 1, \dots, g. \quad (\text{E-2})$$

M step:

Find the mean and variance matrices by iteration.

$$\mu_j^{(k+1)} = \frac{\sum_{i=1}^p \beta_{ij}^{(k)} x^i}{\sum_{i=1}^p \beta_{ij}^{(k)}}, \quad \text{where } \beta_{ij}^{(k)} = \beta_j(x^i, \Theta^{(k)}), \quad j = 1, \dots, g; i = 1, \dots, p, \quad (\text{M-1})$$

$$\Sigma_j^{(k+1)} = \frac{\sum_{i=1}^p \beta_{ij}^{(k)} (x^i - \mu_j^{(k+1)}) (x^i - \mu_j^{(k+1)})^T}{\sum_{i=1}^p \beta_{ij}^{(k)}}, \quad j = 1, \dots, g. \quad (\text{M-2})$$

Assume $N(x, \mu, \Sigma)$ is a one-dimensional normal distribution:

$$(\sigma_j^2)^{(k+1)} = \frac{\sum_{i=1}^p \beta_{ij}^{(k)} \left| (x^i - \mu_j^{(k+1)}) \right|^2}{\sum_{i=1}^p \beta_{ij}^{(k)}}.$$

Finally, the function $F_i(x)$ that we have is the attribute function of C^i . The function $[f] = F(x) = \{F_1(x), \dots, F_p(x)\}$ is the decision function of the classification $C = \{C_1^1, C_2^1, \dots, C_{n_1}^1, C_1^2, C_2^2, \dots, C_{n_2}^2, \dots, C_1^k, C_2^k, \dots, C_{n_k}^k\}$, i.e., the learned classification boundary from the training sample set $K = \{(x^1, y^1), \dots, (x^p, y^p)\}, x^i \in R^n, y^i \in \{0, 1\}^k\}$.

18.6 Conclusion

In the chapter, we introduce the theoretical framework of the quotient space model. First, we show that the model can be used to depict the granulation both in human cognition and in real world. By the model, the relations among different abstraction levels of a problem can be revealed clearly. Second, the model can be extended to deal with fuzzy and non-deterministic relations. Finally, taking machine-learning problem as an example, we show that a learning algorithm can be obtained by the combination operation in quotient space model. This implies that quotient space model can be used to manage not only the top-down problem solving but also the bottom-up data processing.

Acknowledgments

The work was supported in part by National Nature Science Foundation of China, grant no. 60575017 and 60621062, National Key Foundation Research Program of China, grant no. 2003CB317007, 2004CB318108, and the Foundation of Doctoral Program of Ministry of Education, grant no. 20040357002.

References

- [1] Y. Wang, Y. Wang, S. Patel, and D. Patel. A layered reference model of the brain (LRMB). *IEEE Trans. Syst. Man Cybern. (C)* 36 (2006) 124–133.
- [2] B. Zhang and L. Zhang. *Theory and Applications of Problem Solving*. North-Holland Elsevier Science Publishers B.V., Amsterdam, London, New York, Tokyo, 1992.
- [3] L. Zhang and B. Zhang. The quotient space theory of problem solving. *Fundam. Inf.* 59(2–3) (2004) 287–298.
- [4] J.R. Hobbs. Granularity. In: *Proceedings of International Joint Conference on Artificial Intelligence*, Los Angeles, 1985, pp. 432–435.
- [5] L.A. Zadeh. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 19 (1997) 111–127.
- [6] Z. Pawlak. Granularity of knowledge, indiscernibility, and rough sets. In: *Proc. IEEE World Congr. Computat. Intell.* 1 (1998) 106–110.
- [7] S.H. Nguyen, A. Skowron, and J. Stepaniuk. Granular computing: A rough set approach. *Comput. Intell.* 17 (2001) 514–544.
- [8] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer, Dordrecht, 2002.
- [9] Y.Y. Yao. A partition of model of granular computing. *LNCS Trans. Rough Set I* (2004) 232–253.
- [10] Y.Y. Yao and N. Zhong. Potential applications of granular computing in knowledge discovery and data mining. In: *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, FL, July 14–18 1999, pp. 573–580.
- [11] M. Eisenberg. *Topology*. Holt, Rinehart and Winston, Inc., New York, 1974.
- [12] A.P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data using the EM algorithm (with discussion). *J. R. Stat. Soc. Ser. B* 39, (1977) 1–38.
- [13] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ, 1976.

- [14] L. Zhang and B. Zhang. Fuzzy reasoning model under quotient space structure. *Inf. Sci.* 173 (2005) 353–364.
- [15] L. Zhang and B. Zhang. The structure analysis of fuzzy sets. *Int. J. Approx. Reason.* 40 (2005) 92–108.
- [16] L. Zhang and B. Zhang. A Geometrical Representation of McCulloch-Pitts neural model and its applications. *IEEE Trans. Neural Netw.* 10(4) (1999) 925–929.

19

Rough Sets and Granular Computing: Toward Rough-Granular Computing

Andrzej Skowron and Jaroslaw Stepaniuk

19.1 Introduction

The concept approximation problem is the basic problem investigated in machine learning, pattern recognition [1], and data mining [2]. It is necessary to induce approximations of concepts (models of concepts) from available experimental data. The data models developed so far in such areas as statistical learning, machine learning, and pattern recognition are not satisfactory for approximation of complex concepts that occur in the perception process. Researchers from different areas have recognized the necessity to work on new methods for concept approximation (see, e.g., [3, 4]). The main reason for this is that these complex concepts are, in a sense, too far from measurements which render the searching for relevant features in a huge feature space infeasible. There are several research directions aiming at overcoming this difficulty. One of them is based on the interdisciplinary research where the knowledge pertaining to perception in psychology or neuroscience is used to help dealing with complex concepts (see, e.g., [5]). There is a great effort in neuroscience toward understanding the hierarchical structures of neural networks in living organisms [5, 6]. Also mathematicians are recognizing problems of learning as the main problem of the current century [5]. These problems are closely related to complex system modeling as well. In such systems again the problem of concept approximation and its role in reasoning about perceptions is one of the challenges nowadays. One should take into account that modeling complex phenomena entails the use of local models (captured by local agents, if one would like to use the multiagent terminology [7]) that should be fused afterward. This process involves negotiations between agents [7] to resolve contradictions and conflicts in local modeling. This kind of modeling is becoming more and more important in dealing with complex real-life phenomena which we are unable to model using traditional analytical approaches. The latter approaches lead to exact models. However, the necessary assumptions used to develop them result in solutions that are too far from reality to be accepted. New methods or even a new science should therefore be developed for such modeling [8].

One of the possible approaches in developing methods for complex concept approximations can be based on the layered learning [9]. Inducing concept approximation should be developed hierarchically starting from concepts that can be directly approximated using sensor measurements toward complex target concepts related to perception. This general idea can be realized using additional domain knowledge

represented in natural language. For example, one can use some rules of behavior on the roads, expressed in natural language, to assess from recordings (made, e.g., by camera and other sensors) of actual traffic situations if a particular situation is safe or not [10]. To deal with such problems one should develop methods for concept approximations together with methods aiming at approximation of reasoning schemes (over such concepts) expressed in natural language. The foundations of such an approach, creating a core of perception logic, are based on rough set theory [11] and its extension rough mereology [12, 13], both invented in Poland, in combination with other soft computing tools, in particular with fuzzy sets.

The outlined problems are some special problems which can be formulated in a more general setting in granular computing (GC) [14, 15].

Information granulation can be viewed as a human way of achieving data compression and it plays a key role in implementing the divide-and-conquer strategy in human problem solving [16]. Granules are obtained in the process of information granulation (see, e.g., [17–19]).

GC is based on processing of complex information entities called granules. Generally speaking, granules are collections of entities that are arranged together due to their similarity, functional adjacency, or indistinguishability [16, 17, 20, 21].

One of the main branch of GC is computing with words and perceptions. GC ‘derives from the fact that it opens the door to computation and reasoning with information which is perception – rather than measurement – based. Perceptions play a key role in human cognition, and underlie the remarkable human capability to perform a wide variety of physical and mental tasks without any measurements and any computations. Everyday examples of such tasks are driving a car in city traffic, playing tennis and summarizing a story’ [16, 22].

We consider the optimization tasks in which we are searching for optimal solutions satisfying some constraints. These constraints are often vague, imprecise, and/or specifications of concepts and their dependencies which constitute the constraints are incomplete. Decision tables [11] are examples of such constraints. Another example of constraints can be found, e.g., in [23–26], where a specification is given by a domain knowledge and data sets. Domain knowledge is represented by an ontology of vague concepts and the dependencies between them. In a more general case, the constraints can be specified in a simplified fragment of a natural language [16].

Granules are constructed using information calculi. Granules are objects constructed in computations aiming at solving the above-mentioned optimization tasks. In our approach, we use the general optimization criterion based on the minimal length principle [27, 28]. In searching for (sub)optimal solutions it is necessary to construct many compound granules using some specific operations, such as generalization, specification, or fusion. Granules are labeled by parameters. By tuning these parameters we optimize the granules relative to their description size and the quality of data description, i.e., two basic components on which the optimization measures are defined.

From this general description of tasks in GC it follows that together with specification of elementary granules and operation on them it is necessary to define quality measures for granules (e.g., measures of their inclusion, covering, or closeness) and tools for measuring the size of granules. Very important are also optimization strategies of already-constructed (parameterized) granules.

In this chapter, we discuss the rough-granular computing, i.e., GC based on the rough set approach.

Rough set theory due to Zdzisław Pawlak [11, 29–31] is a mathematical approach to imperfect knowledge. The problem of imperfect knowledge has been tackled for a long time by philosophers, logicians, and mathematicians. Recently, it has become a crucial issue for computer scientists as well, particularly in the area of artificial intelligence. There are many approaches to the problem of how to understand and manipulate imperfect knowledge. The most successful one is, no doubt, the fuzzy set theory proposed by Lotfi A. Zadeh [16]. Rough set theory presents still another attempt to solve this problem. It is based on an assumption that objects are perceived by partial information about them. Due to this some objects can be indiscernible. Indiscernible objects form elementary granules. From this fact it follows that some sets cannot be exactly described by available information about objects; they are rough, not crisp. Any rough set is characterized by its (lower and upper) approximations. The difference between the upper and lower approximation of a given set is called its boundary. Rough set theory expresses vagueness by employing a boundary region of a set. If the boundary region of a set is empty, it means that the set is crisp; otherwise the set is rough (inexact). A non-empty boundary region of a set indicates that our knowledge about the set is not sufficient to define the set precisely.

One can recognize that rough set theory is, in a sense, a formalization of the idea presented by Gotlob Frege [32].

One of the consequences of perceiving objects using only available information about them is that for some objects one cannot decide if they belong to a given set or not. However, one can estimate the degree to which objects belong to sets. This is another crucial observation in building foundations for approximate reasoning. In dealing with imperfect knowledge one can characterize satisfiability of relations between objects only to a degree, not precisely. Among relations on objects, the rough inclusion relation, which describes to what degree objects are parts of other objects, plays a special role. A rough mereological approach (see, e.g., [12, 13, 26]) is an extension of the Leśniewski mereology [33] and is based on the relation *to be a part to a degree*. It will be interesting to note here that Jan Łukasiewicz was the first who started to investigate the inclusion to a degree of concepts in his discussion on relationships between probability and logical calculi [34].

In the rough set approach, we are searching for data models using the minimal length principles. Searching for models with small size is performed by means of many different kinds of reducts, i.e., minimal sets of attributes preserving some constraints.

One of the very successful techniques for rough set methods is Boolean reasoning. The idea of Boolean reasoning is based on constructing for a given problem P a corresponding Boolean function f_P with the following property: the solutions for the problem P can be decoded from prime implicants of the Boolean function f_P . It is worth to mention that to solve real-life problems it is necessary to deal with Boolean functions having a large number of variables.

A successful methodology based on the discernibility of objects and Boolean reasoning has been developed in rough set theory for computing of many key constructs such as reducts and their approximations, decision rules, association rules, discretization of real-value attributes, symbolic value grouping, searching for new features defined by oblique hyperplanes or higher order surfaces, pattern extraction from data, as well as conflict resolution or negotiation (see, e.g., [35, 36]). Most of the problems involving the computation of these entities are NP-complete or NP-hard. However, we have been successful in developing efficient heuristics yielding suboptimal solutions for these problems. The results of experiments on many data sets are very promising. They show very good quality of solutions generated by the heuristics in comparison with other methods reported in literature (e.g., with respect to the classification quality of unseen objects). Moreover, they are very time efficient. It is important to note that the methodology makes it possible to construct heuristics having a very important approximation property. Namely, expressions generated by heuristics (i.e., implicants) close to prime implicants define approximate solutions for the problem (see, e.g., [37]).

The rough set approach was further developed to deal with more compound granules than elementary granules. In this chapter, we present a methodology for modeling of compound granules using the rough set approach. The methodology is based on operations on information systems. There are two basic steps in such a modeling. In the first step, new granules are constructed from objects represented by granules in some already-constructed information systems. These new granules are used as objects in the newly constructed information systems. In the second step the features of the new granules are added. This approach can be used for modeling, e.g., compound granules in spatiotemporal reasoning.

This chapter is structured as follows. In Section 19.2 we recall the definition of information granulation. We also discuss systems of granules and examples of granules. In Section 19.3 we investigate granules in multiagent systems. In Section 19.4 we discuss modeling of compound granules based on information systems. This chapter is complementary to chapters 13 and 14. To make this chapter self-contained, we recall some definitions such as generalized approximation spaces or extensions of approximation spaces as well as some examples of hierarchical learning of complex concepts from data and domain knowledge.

19.2 Information Granulation and Granules

The concept of information granulation is rooted in several papers starting with [20], in which the concepts of a linguistic variable and granulation were introduced. Information granulation is performed on granules. In this chapter, we assume that any granule is a pair:

$$(name, content), \quad (1)$$

where *name* is the granule (in some external language, e.g., a natural language) and *content* describes details of the granule name construction (in an internal language e.g., system language) together with their meaning (semantics) approximating the concept from external language.

In many examples (where the external language in the same as the internal language), the granule names (labels) are formulas from some language and the granule contents are interpreted as the semantics of such formulas. In other examples, granule contents can have more compound structures defined by some other granules. For example, one can consider a granule representing a cluster

$$(patient_cluster, cluster), \quad (2)$$

where *patient_cluster* denotes the name of a patient cluster in a medical database having similar symptoms and *cluster* consists of a cluster (approximate) definition definition including the cluster construction and its semantics. Let us consider one more example:

$$(safe, classifier), \quad (3)$$

where *safe* is a vague concept describing that the situation on the road is *safe* and *classifier* is the induced approximation of the vague concept *safe*. Each classifier can be treated as a granule with *name* describing the classifier construction and *content* describing the classifier semantics. The granule presented in the last example describes, in a sense, the *meaning* of the vague concept *safe* relative, e.g., to an agent implemented in a computer system.

In the following sections, we present some systems of granules, making it possible to describe construction of different kinds of basic granules.

19.2.1 Granule Systems

In this section, we present a basic notion for our approach, i.e., *granule* system. Any such system S consists of a set of granules G . Moreover, a family of relations with the intended meaning *to be a part to a degree* between granules is distinguished. The degree structure is described by a relation *used for comparing degrees*. More formally, a granule system is any tuple

$$S = (G, H, <, \{v_p\}_{p \in H}, size), \quad (4)$$

where

1. G is a non-empty set of granules.
2. H is a non-empty set of granule inclusion degrees with a binary relation $<$ (usually a strict partial order), which defines on H a structure used to compare the degrees.
3. $v_p \subseteq G \times G$ is a binary relation *to be a part to a degree at least p* between granules from G , called *rough inclusion*.
4. $size : G \longrightarrow R_+$ is the granule size function, where R_+ is the set of non-negative reals.

In constructing of granule systems, it is necessary to give a constructive definition of all their components.

In particular, one should specify how more compound granules are defined from already-defined granules or given elementary granules. Usually, the set of granules is defined as the least set generated from distinguished elementary granules by some operations on the granules. In the following sections, we discuss several examples of such operations.

One can consider the following examples of formulas defining elementary granules:

1. a set of descriptors (selectors) of the form (a, v) , where $a \in A$ and $v \in V_a$ for some finite attribute set A and value sets V_a ;
2. a set of descriptor conjunctions.

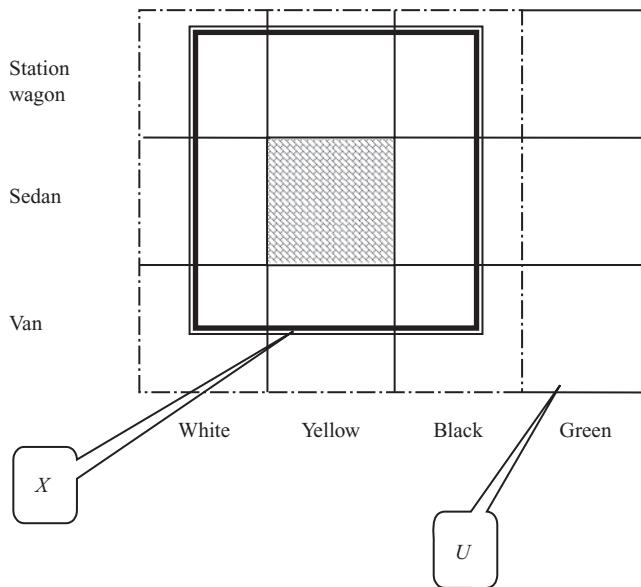


Figure 19.1 Granules in the standard rough set model

In the standard rough set model, granules are corresponding to indiscernibility classes of an equivalence relation. Let, for example, U be a set of cars (see, Figure 19.1) and we consider two attributes *color* and *type* of the car's body. Let

$$V_{\text{color}} = \{\text{white}, \text{yellow}, \text{black}, \text{green}\}$$

and

$$V_{\text{type}} = \{\text{van}, \text{sedan}, \text{station wagon}\}.$$

In this case we obtain 12 granules corresponding to conjunctions of descriptors; e.g.,

$$(\text{color}, \text{white}) \wedge (\text{type}, \text{van}), (\text{color}, \text{yellow}) \wedge (\text{type}, \text{van}).$$

For a set of cars X the lower and the upper approximation is also depicted in Figure 19.1.

Examples of more complex granules are tolerance granules created by means of similarity (tolerance) relation between elementary granules, decision rules, or sets of decision rules.

Notice that the existing measures of inclusion should be extended on more compound granules. Strategies for these extensions are selected so that constructed granules allow us to make a progress in constructing the target granules. In the following sections, we outline some of these issues in modeling of granules.

19.2.2 Name and Content: Syntax and Semantics

In this section, we present examples of granules with names defined by some formulas and contents defined by semantics of these formulas.

Formulas are used to express properties of objects. Hence, we assume that together with a given information system there are defined

- a set of formulas Φ over some language;
- semantics Sem of formulas from Φ , i.e., a function from Φ into the power set $P(U)$.

Let us consider an example [11]. We define a language L_{IS} used for elementary granule description, where $IS = (U, A)$ is an information system. The syntax of L_{IS} is defined recursively by

1. $(a \text{ in } V) \in L_{IS}$, for any $a \in A$ and $V \subseteq V_a$.
2. If $\alpha \in L_{IS}$, then $\neg\alpha \in L_{IS}$.
3. If $\alpha, \beta \in L_{IS}$, then $\alpha \wedge \beta \in L_{IS}$.
4. If $\alpha, \beta \in L_{IS}$, then $\alpha \vee \beta \in L_{IS}$.

The semantics of formulas from L_{IS} with respect to an information system IS is defined recursively by

1. $Sem_{IS}(a \text{ in } V) = \{x \in U : a(x) \in V\}$.
2. $Sem_{IS}(\neg\alpha) = U - Sem_{IS}(\alpha)$.
3. $Sem_{IS}(\alpha \wedge \beta) = Sem_{IS}(\alpha) \cap Sem_{IS}(\beta)$.
4. $Sem_{IS}(\alpha \vee \beta) = Sem_{IS}(\alpha) \cup Sem_{IS}(\beta)$.

We now present the syntax and the semantics of examples of granules. These granules are constructed by taking collections of already-specified granules. They comprise parameters which can be adjusted in applications. In the following sections, we discuss some other kinds of operations on granules as well as the inclusion and closeness relations for such granules.

Let us note that any granule g can formally be defined by the granules syntax $Syn(g)$ and semantics $Sem(g)$. However, for simplicity of notation we often use only one component of the granules to denote it.

19.2.3 Examples of Granules

Elementary Granules

In an information system $IS = (U, A)$, elementary granules are defined by $EF_B(x)$, where $EF_B(x)$ is a conjunction of selectors (descriptors) of the form $a = a(x)$, and $x \in U$. For example, the meaning of an elementary granule $a = 1 \wedge b = 1$ is defined by

$$Sem_{IS}(a = 1 \wedge b = 1) = \{x \in U : a(x) = 1 \& b(x) = 1\}.$$

Thus, in the system

$$S_B = (G_B, H, <, \{\nu_p\}_{p \in H}, size) \quad (5)$$

of elementary granules, G_B is a set of conjunctions of selectors, $H = [0, 1]$ and $\nu_p(EF_B, EF'_B)$, if and only if

$$\frac{card(Sem_{IS}(EF_B) \cap Sem_{IS}(EF'_B))}{card(Sem_{IS}(EF_B))} \geq p.$$

The number of conjuncts in the granule can be taken as the granule size and it is one of the parameters to be tuned, e.g., by the dropping condition technique used in machine learning.

One can extend the set of elementary granules assuming that if α is any Boolean combination of descriptors over A , then $(\bar{B}\alpha)$ and $(\underline{B}\alpha)$ define syntax of elementary granules too, for any $B \subseteq A$.

Sequences of Granules

Let us assume that S is a sequence of granules and the semantics $Sem_{IS}(\bullet)$ in IS of its elements have been defined. We extend $Sem_{IS}(\bullet)$ on S by

$$Sem_{IS}(S) = \{Sem_{IS}(g)\}_{g \in S}.$$

Example 1. *Granules defined by rules in information systems are examples of sequences of granules. Let IS be an information system and let (α, β) be a new granule received from the rule **if** α **then** β , where α and β are elementary granules of IS . The semantics $Sem_{IS}((\alpha, \beta))$ of (α, β) is the pair of sets*

$$(Sem_{IS}(\alpha), Sem_{IS}(\beta)).$$

If the right-hand sides of rules represent decision classes, then the number of conjuncts on the left-hand sides is one of the parameters to be adjusted during classifier construction. A typical goal is to search for minimal (or less than minimal) number of such conjuncts (corresponding to the largest generalization), which still guarantee the satisfactory degree of inclusion in a decision class.

Sets of Granules

Let us assume that a set G of granules and the semantics $Sem_{IS}(\bullet)$ in IS for granules from G have been defined. We extend $Sem_{IS}(\bullet)$ on the family of sets $H \subseteq G$ by $Sem_{IS}(H) = \{Sem_{IS}(g) : g \in H\}$. One can consider as a parameter of any such granule its cardinality or its size (e.g., the length of such granule representation). In the first case, a typical problem is to search in a given family of granules for a granule of the smallest cardinality sufficiently close to a given one.

Example 2. *One can consider granules defined by sets of rules. Assume that there is a set of rules $Rule_Set = \{(\alpha_i, \beta_i) : i = 1, \dots, k\}$. The semantics of $Rule_Set$ is defined by*

$$Sem_{IS}(Rule_Set) = \{Sem_{IS}((\alpha_i, \beta_i)) : i = 1, \dots, k\}.$$

The above-mentioned searching problem for a set of granules corresponds in the case of rule sets to searching for the simplest representation of a given rule collection by another set of rules (or a single rule) sufficiently close to the collection.

Example 3. *Let us consider a set G of elementary granules – describing possible situations together – with decision table DT_α representing decision tables for any situation $\alpha \in G$. Assume $Rule_Set(DT_\alpha)$ to be a set of decision rules generated from decision table DT_α (e.g., in the minimal form [28]). Now let us consider a new granule*

$$\{(\alpha, Rule_Set(DT_\alpha)) : \alpha \in G\},$$

with semantics defined by

$$\{Sem_{DT}((\alpha, Rule_Set(DT_\alpha))) : \alpha \in G\} = \{(Sem_{IS}(\alpha), Sem_{DT}(Rule_Set(DT_\alpha))) : \alpha \in G\}.$$

An example of a parameter to be tuned is the number of situations represented in such granule. A typical task is to search for a granule with the minimal number of situations creating together with the sets of rules, corresponding to them, a granule sufficiently close to the original one.

Extension of Granules Defined by Tolerance Relation

Now we present examples of granules obtained by application of a tolerance relation (i.e., reflexive and symmetric relation; for more information see, e.g., [38]).

Example 4. *One can consider extension of elementary granules defined by tolerance relation. Let $IS = (U, A)$ be an information system and let τ a tolerance relation on elementary granules of IS . Any pair $(\tau : \alpha)$ is called a τ -elementary granule. The semantics $Sem_{IS}((\tau : \alpha))$ of $(\tau : \alpha)$ is the family $\{Sem_{IS}(\beta) : (\beta, \alpha) \in \tau\}$. Parameters to be tuned in searching for relevant tolerance granule can be its support (represented by the number of supporting it objects) and the degree of its inclusion (or closeness) in some other granules as well as parameters specifying the tolerance relation (for more information, see, e.g., [39]).*

Example 5. *Let us consider granules defined by rules of a tolerance information systems [38]. Let $IS = (U, A)$ be an information system and let τ a tolerance relation on elementary granules of IS . If if*

α **then** β is a rule in IS, then the semantics of a new granule $(\tau : \alpha, \beta)$ is defined by $Sem_{IS}((\tau : \alpha, \beta)) = Sem_{IS}((\alpha, \tau)) \times Sem_{IS}((\beta, \tau))$. Parameters to be tuned are the same as in the case of granules being sets of more elementary granules as well as parameters of the tolerance relation.

Example 6. We consider granules defined by sets of decision rules corresponding to a given evidence (α) in tolerance decision tables. Let $DT = (U, A, d)$ a decision table and let τ a tolerance on elementary granules of $IS = (U, A)$. Now, any granule $(\alpha, Rule_Set(DT_\alpha))$ can be considered as a representative for the granule cluster

$$(\tau : (\alpha, Rule_Set(DT_\alpha))),$$

with the semantics

$$Sem_{DT}((\tau : (\alpha, Rule_Set(DT_\alpha)))) = \{Sem_{DT}((\beta, Rule_Set(DT_\beta))) : (\beta, \alpha) \in \tau\}.$$

One can see that the considered case is a special case of the granules from Example 3, with G defined by a tolerance relation.

19.3 Granules in Multiagent Systems

Granules are involved in many tasks of approximate reasoning in multiagent systems [7]. Among them are

1. understanding granules used by other agents;
2. interaction of granules in searching for patterns used for compound concepts approximation;
3. discovery of new granules in interaction with environments used for prediction of behavior.

In the following sections, we will present several compound granules used in solving such tasks.

We begin with a short discussion on approximation spaces, i.e., granules used in concept approximation. Approximation spaces can be used by a given agent for approximating concepts used by another agent [40, 41]. In the simplest case, as the result of such an interaction, a decision table is obtained. This table is then used for concept approximation. We show that concept approximation on an extension of a given sample of objects can be treated as a searching task for an extension of approximation space. We also outline applications of granules in compound concept approximation where compound hierarchical patterns (hierarchical pattern granules) for approximation of such concepts are constructed by interaction of simpler patterns (pattern granules). Granulation of such patterns is used in searching for concept approximations (data models) with (sub)minimal size and with satisfactory quality. It is worthwhile to mention that granulation can also be applied to rules used for patterns construction. In approximation of compound concepts we propose to use a domain ontology that makes the searching for compound concept approximation feasible [9, 23–26]. In the ontology [42], (vague) concepts and local dependencies between them are specified. Global dependencies can be derived from local dependencies. Such derivations can be used as hints in searching for relevant compound patterns (granules) in approximation of more compound concepts from the ontology. The ontology approximation problem is one of the fundamental problems related to approximate reasoning in distributed environments. One should construct (in a given language that is different from the ontology specification language) not only approximations of concepts from ontology but also vague dependencies specified in the ontology. It is worthwhile to mention that an ontology approximation should be induced on the basis of incomplete information about concepts and dependencies specified in the ontology. Granule calculi based on rough sets have been proposed as tools making it possible to solve this problem. Vague dependencies have vague concepts in premises and conclusions. The approach to approximation of vague dependencies based only on degrees of closeness of concepts from dependencies and their approximations (classifiers) is not satisfactory for approximate reasoning. Hence, more advanced approach should be developed. Approximation of any vague dependency is a method which allows for any object to compute the arguments ‘for’ and ‘against’ its membership to the dependency’s conclusion on the basis of the analogous

arguments relative to the dependency's premises. Any argument is a compound granule (compound pattern). Arguments are fused by local schemes (production rules) discovered from data. Further fusions are possible through composition of local schemes, called approximate reasoning schemes (AR schemes) (see, e.g., [13, 23]). To estimate the degree to which (at least) an object belongs to concepts from ontology, the arguments 'for' and 'against' those concepts are collected and next a conflict resolution strategy is applied to them to predict the degree. This inference process is analogous to the inference process used in fuzzy logic [43] with numerical degree of membership functions. In the considered case, the numerical values are substituted by arguments 'for' and 'against' and the fuzzification is replaced by rules defining how the arguments from the left-hand sides of dependencies are transformed into arguments for the concepts on the right-hand sides. The defuzzification is substituted by conflict resolution strategy. In the discussed approach, it is assumed that the rules are discovered from data and domain knowledge.

The performed experiments based on approximation of concept ontology (see, e.g., [13, 26, 38, 40, 44–47]) showed that domain knowledge enables us to discover relevant patterns in sample of objects used for compound concept approximation. Our approach to compound concept approximation and approximate reasoning about compound concepts is based on the rough-granular approach.

For modeling computations of multiagent systems more compound granules are needed. Let us observe that granule systems themselves can also be treated as granules. For example, each agent can be represented by a granule system. Moreover, in granular computations modeling the behavior of multiagent systems some specific operations on granules representing granule systems should be defined. There are several reasons for introducing such operations. For example,

- Granule systems of agents should be adaptively changed in interaction of agents with environments.
- During coalition formation by a team of agents their granule systems should be fused into a new granule system relevant for the new coalition.

Other compound granules are needed for reasoning about behavior of agents in multiagent systems.

19.3.1 Approximation Spaces

In this section, we recall the definition of an approximation space from [38, 48]. Approximation spaces can be treated as granules used for concept approximation. They are some special parameterized relational structures. Tuning of parameters makes it possible to search for relevant approximation spaces relative to given concepts.

Definition 7. A parameterized approximation space is a system $AS_{\#, \$} = (U, I_{\#}, v_{\$})$, where

- U is a non-empty set of objects;
- $I_{\#} : U \rightarrow P(U)$ is an uncertainty function, where $P(U)$ denotes the power set of U ;
- $v_{\$} : P(U) \times P(U) \rightarrow [0, 1]$ is a rough inclusion function;

and $\#, \$$ denote vectors of parameters. (The indexes $\#, \$$ will be omitted if it does not lead to misunderstanding.)

Uncertainty Function

The uncertainty function defines for every object x , a set of objects described similarly to x . The set $I(x)$ is called the neighborhood of x (see, e.g., [11, 38]).

We assume that the values of the uncertainty function are defined using a *sensory environment*, i.e., a pair $(L, \|\cdot\|_U)$, where L is a set of formulas, called the *sensory formulas*, and $\|\cdot\|_U : L \longrightarrow P(U)$ is the *sensory semantics*. We assume that for any sensory formula α and any object $x \in U$, the information if $x \in \|\alpha\|_U$ holds is available. The set $\{\alpha : x \in \|\alpha\|_U\}$ is called the *signature of x in AS* and is denoted by $Inf_{AS}(x)$. For any $x \in U$, the set $N_{AS}(x)$ of *neighborhoods of x in AS* is defined by $\{\|\alpha\|_U : x \in \|\alpha\|_U\}$

and from this set the neighborhood $I(x)$ is constructed. For example, $I(x)$ is defined by selecting an element from the set $\{\|\alpha\|_U : x \in \|\alpha\|_U\}$ or by $I(x) = \bigcap \mathcal{N}_{AS}(x)$. Observe that any sensory environment $(L, \|\cdot\|_U)$ can be treated as a parameter of I from the vector # (see Definition 7).

Let us consider two examples. Any decision table $DT = (U, A, d)$ [11] defines an approximation space $AS_{DT} = (U, I_A, v_{SRI})$, where, as we will see, $I_A(x) = \{y \in U : a(y) = a(x) \text{ for all } a \in A\}$. Any sensory formula is a descriptor, i.e., a formula of the form $a = v$, where $a \in A$ and $v \in V_a$ with the standard semantics $\|a = v\|_U = \{x \in U : a(x) = v\}$. Then, for any $x \in U$, its signature $Inf_{AS_{DT}}(x)$ is equal to $\{a = a(x) : a \in A\}$ and the neighborhood $I_A(x)$ is equal to $\bigcap \mathcal{N}_{AS_{DT}}(x)$. Another example can be obtained assuming that for any $a \in A$, there is given a tolerance relation $\tau_a \subseteq V_a \times V_a$ (see, e.g., [38]). Let $\tau = \{\tau_a\}_{a \in A}$. Then, one can consider a tolerance decision table $DT_\tau = (U, A, d, \tau)$, with tolerance descriptors $a =_{\tau_a} v$ and their semantics $\|a =_{\tau_a} v\|_U = \{x \in U : v\tau_a a(x)\}$. Any such tolerance decision table $DT_\tau = (U, A, d, \tau)$ defines the approximation space $AS_{DT_\tau} = (U, I_A, v_{SRI})$, with the signature $Inf_{AS_{DT_\tau}}(x) = \{a =_{\tau_a} a(x) : a \in A\}$ and the neighborhood $I_A(x) = \bigcap \mathcal{N}_{AS_{DT_\tau}}(x)$ for any $x \in U$.

The fusion of $\mathcal{N}_{AS_{DT_\tau}}(x)$ for computing the neighborhood of x can have many different forms; the intersection is only an example. One can also consider some more general uncertainty functions, e.g., with values in $P^2(U)$ [49]. For example, to compute the value of $I(x)$ first some subfamilies of $\mathcal{N}_{AS}(x)$ can be selected and next the family consisting of intersection of each such a subfamily is taken as the value of $I(x)$.

Note that any sensory environment $(L, \|\cdot\|_U)$ defines an information system with the universe U of objects. Any row of such an information system for an object x consists of information if $x \in \|\alpha\|_U$ holds for any sensory formula α . Let us also observe that in our examples we have used a simple sensory language defined by descriptors of the form $a = v$. One can consider a more general approach by taking, instead of the simple structure $(V_a, =)$, some other relational structures R_a with the carrier V_a for $a \in A$ and a signature. Then any formula (with one free variable) from a sensory language with the signature that is interpreted in R_a defines a subset $V \subseteq V_a$ and induces on the universe of objects a neighborhood consisting of all objects having values of the attribute a in the set V . Note that this is the basic step in hierarchical modeling [50].

Rough Inclusion Function

One can consider general constraints which the rough inclusion functions should satisfy. Searching for such constraints initiated investigations resulting in creation and development of rough mereology (see, e.g., [12, 51] and the bibliography in [51]). In this subsection, we present only some examples of rough inclusion functions.

The rough inclusion function $v_S : P(U) \times P(U) \rightarrow [0, 1]$ defines the degree of inclusion of X in Y , where $X, Y \subseteq U$.

In the simplest case it can be defined by (see, e.g., [11, 38]):

$$v_{SRI}(X, Y) = \begin{cases} \frac{\text{card}(X \cap Y)}{\text{card}(X)} & \text{if } X \neq \emptyset \\ 1 & \text{if } X = \emptyset. \end{cases}$$

This measure is widely used by the data mining and rough set communities. It is worth mentioning that Jan Łukasiewicz [34] was the first one who used this idea to estimate the probability of implications. However, rough inclusion can have a much more general form than inclusion of sets to a degree (see, e.g., [12, 49, 51]).

Another example of rough inclusion function v_t can be defined using the standard rough inclusion and a threshold $t \in (0, 0.5)$ using the following formula:

$$v_t(X, Y) = \begin{cases} 1 & \text{if } v_{SRI}(X, Y) \geq 1 - t \\ \frac{v_{SRI}(X, Y) - t}{1 - 2t} & \text{if } t \leq v_{SRI}(X, Y) < 1 - t \\ 0 & \text{if } v_{SRI}(X, Y) \leq t. \end{cases}$$

The rough inclusion function v_t is used in the variable-precision rough set approach [52].

Another example of rough inclusion is used for function approximation [49] and relation approximation [53].

Then the inclusion function v^* for subsets $X, Y \subseteq U \times U$, where $X, Y \subseteq \mathcal{R}$ and \mathcal{R} is the set of reals, is defined by

$$v^*(X, Y) = \begin{cases} \frac{\text{card}(\pi_1(X \cap Y))}{\text{card}(\pi_1(X))} & \text{if } \pi_1(X) \neq \emptyset \\ 1 & \text{if } \pi_1(X) = \emptyset, \end{cases} \quad (6)$$

where π_1 is the projection operation on the first coordinate. Assume now that X is a cube and Y is the graph $G(f)$ of the function $f : \mathcal{R} \rightarrow \mathcal{R}$. Then, e.g., X ; is in the lower approximation of f if the projection on the first coordinate, of the intersection $X \cap G(f)$ is equal to the projection of X on the first coordinate. This means that the part of the graph $G(f)$ is ‘well’ included in the box X ; i.e., for all arguments that belong to the box projection on the first coordinate, the value of f is included in the box X projection on the second coordinate.

Usually, there are several parameters that are tuned in searching for a relevant rough inclusion function. Such parameters are listed in the vector $\#$. An example of such parameters is the threshold mentioned for the rough inclusion function used in the variable-precision rough set model. We would like to mention some other important parameters. Among them are pairs $(L^*, \|\cdot\|_U^*)$, where, L^* is an extension of L and $\|\cdot\|_U^*$ is an extension of $\|\cdot\|_U$ and where $(L, \|\cdot\|_U)$ is a sensory environment. For example, if L consists of sensory formulas $a = v$ for $a \in A$ and $v \in V_a$, then one can take as L^* the set of descriptor conjunctions. For rule-based classifiers we search in such a set of formulas for patterns relevant for decision classes. We present more details in the following section.

19.3.2 Lower and Upper Approximations

The lower and the upper approximations of subsets of U are defined as follows.

Definition 8. For any approximation space $AS_{\#, \$} = (U, I_{\#}, v_{\$})$ and any subset $X \subseteq U$, the lower and upper approximations are defined by

$$LOW(AS_{\#, \$}, X) = \{x \in U : v_{\$}(I_{\#}(x), X) = 1\},$$

$$UPP(AS_{\#, \$}, X) = \{x \in U : v_{\$}(I_{\#}(x), X) > 0\}, \text{ respectively.}$$

The lower approximation of a set X with respect to the approximation space $AS_{\#, \$}$ is the set of all objects which can be classified with certainty as objects of X with respect to $AS_{\#, \$}$. The upper approximation of a set X with respect to the approximation space $AS_{\#, \$}$ is the set of all objects which can be possibly classified as objects of X with respect to $AS_{\#, \$}$.

Several known approaches to concept approximations can be covered using the discussed approximation spaces here, e.g., the approach given in [11], approximations based on the variable-precision rough set model [52], or tolerance (similarity) rough set approximations (see, e.g., [38] and references therein).

The classification methods for concept approximation developed in machine learning and pattern recognition make it possible to decide if a given object belongs to the approximated concept or not [1]. The classification methods yield the decisions using only partial information about approximated concepts. This fact is reflected in the rough set approach by assumption that concept approximations should be defined using only partial information about approximation spaces. To decide if a given object belongs to the (lower or upper) approximation of a given concept, the rough inclusion function values are

needed. In the next section, we show how such values, so needed in classification making, are estimated on the basis of available partial information about approximation spaces.

19.3.3 Extension of Approximation Spaces: Concept Approximation

In this section we consider the problem of approximation of concepts over a universe U^∞ (concepts that are subsets of U^∞) [45]. We assume that the concepts are perceived only through some subsets of U^∞ , called samples. This is a typical situation in the machine learning, pattern recognition, or data mining approaches [1, 2]. We explain the rough set approach to induction of concept approximations using the generalized approximation spaces of the form $AS = (U, I, v)$ and an extension operation of such approximation spaces.

Let $U \subseteq U^\infty$ be a finite sample. By Π_U , we denote a perception function from $P(U^\infty)$ into $P(U)$ defined by $\Pi_U(C) = C \cap U$ for any concept $C \subseteq U^\infty$. Let $AS = (U, I, v)$ be an approximation space over the sample U .

The problem we consider is how to extend the approximations of $\Pi_U(C)$ defined by AS to approximation of C over U^∞ . We show that the problem can be described as searching for an extension $AS_C = (U^\infty, I_C, v_C)$ of the approximation space AS , relevant for approximation of C . This requires to show how to extend the inclusion function v from subsets of U to subsets of U^∞ that are relevant for the approximation of C . Observe that for the approximation of C it is enough to induce the necessary values of the inclusion function v_C without knowing the exact value of $I_C(x) \subseteq U^\infty$ for $x \in U^\infty$.

Let AS be a given approximation space for $\Pi_U(C)$ and let us consider a language L in which the neighborhood $I(x) \subseteq U$ is expressible by a formula $pat(x)$, for any $x \in U$. It means that $I(x) = \|pat(x)\|_U \subseteq U$, where $\|pat(x)\|_U$ denotes the meaning of $pat(x)$ restricted to the sample U . In case of rule-based classifiers, patterns of the form $pat(x)$ are defined by feature value vectors (or conjunction of descriptors).

We assume that for any new object $x \in U^\infty \setminus U$, we can obtain (e.g., as a result of sensor measurement) a pattern $pat(x) \in L$ with semantics $\|pat(x)\|_{U^\infty} \subseteq U^\infty$. However, the relationships between granules over U^∞ like sets, $\|pat(x)\|_{U^\infty}$ and $\|pat(y)\|_{U^\infty}$, for different $x, y \in U^\infty$, are, in general, known only if they can be expressed by relationships between the restrictions of these sets to the sample U , i.e., between sets $\Pi_U(\|pat(x)\|_{U^\infty})$ and $\Pi_U(\|pat(y)\|_{U^\infty})$.

The set of patterns $\{pat(x) : x \in U\}$ is usually not relevant for approximation of the concept $C \subseteq U^\infty$. Such patterns are too specific, or not enough general, and can directly be applied only to a very limited number of new objects. However, by using some generalization strategies, one can search, in a family of patterns definable from $\{pat(x) : x \in U\}$ in L , for such new patterns that are relevant for approximation of concepts over U^∞ . Let us consider a subset $PATTERNS(AS, L, C) \subseteq L$ chosen as a set of pattern candidates for relevant approximation of a given concept C . For example, in case of rule-based classifier one can search for such candidate patterns among sets definable by subsequences of feature value vectors corresponding to objects from the sample U (or by drooping descriptors from the orginal conjunction). The set $PATTERNS(AS, L, C)$ can be selected by using some quality measures checked on meanings (semantics) of its elements restricted to the sample U (like the number of examples from the concept $\Pi_U(C)$ and its complement that support a given pattern). Then, on the basis of properties of sets definable by these patterns over U , we induce approximate values of the inclusion function v_C on subsets of U^∞ definable by any of such pattern and the concept C .

Next, we induce the value of v_C on pairs (X, Y) , where $X \subseteq U^\infty$ is definable by a pattern from $\{pat(x) : x \in U^\infty\}$ and $Y \subseteq U^\infty$ is definable by a pattern from $PATTERNS(AS, L, C)$.

Finally, for any object $x \in U^\infty \setminus U$, we induce the approximation of the degree $v_C(\|pat(x)\|_{U^\infty}, C)$, applying a conflict resolution strategy *Conflict.Res* (a voting strategy, in case of rule-based classifiers) to two families of degrees:

$$\{v_C(\|pat(x)\|_{U^\infty}, \|pat\|_{U^\infty}) : pat \in PATTERNS(AS, L, C)\}, \quad (7)$$

$$\{v_C(\|pat\|_{U^\infty}, C) : pat \in PATTERNS(AS, L, C)\}. \quad (8)$$

Values of the inclusion function for the remaining subsets of U^∞ can be chosen in any way – they do not have any impact on the approximations of C . Moreover, observe that for the approximation of C we do not need to know the exact values of the uncertainty function I_C – it is enough to induce the values of the inclusion function ν_C . Observe that the defined extension ν_C of ν to some subsets of U^∞ makes it possible to define an approximation of the concept C in a new approximation space AS_C .

Moreover, one can also follow principles of Bayesian reasoning and use degrees of ν_C to approximate C .

In this way, the rough set approach to induction of concept approximations can be explained as a process of inducing a relevant approximation space.

Any approximation space can be treated as a compound granule labeled by many parameters, such as attribute sets defining the neighborhoods, rough inclusions, neighborhood size measures, and parameters of patterns used for estimation of the extension of the rough inclusion. One can define the quality of a given approximation space relative to an approximated concept, e.g., by means of the boundary region size and also the approximation space size measured by an aggregation of the sizes of the approximation space components. In the process of searching for the (sub)optimal granule, i.e., in this case a classifier, all these parameters are tuned using the minimal length principle. In searching for relevant components of approximation spaces, employing various kinds of reducts plays an important role [31].

19.3.4 Compound Concept Approximation

The strategies for data models inducing developed so far are often not satisfactory for approximation of compound concepts that occur in the perception process.

One of the possible approaches in developing methods for compound concept approximations can be based on the layered (hierarchical) learning [9]. Inducing concept approximation should be developed hierarchically, starting from concepts that can be directly approximated using sensor measurements toward compound target concepts related to perception. This general idea can be realized using additional domain knowledge represented in natural language. For example, one can use some rules of behavior on the roads, expressed in natural language, to assess from recordings (made, e.g., by camera and other sensors) of actual traffic situations, if a particular situation is safe or not (see, e.g., [10, 23, 24]). The hierarchical learning has also been used for identification of risk patterns in medical data and extended for therapy planning (see, e.g. [54]). To deal with such problems one should develop methods for concept approximations together with methods aiming at approximation of reasoning schemes (over such concepts) expressed in natural language. The foundations of such an approach, creating a core of perception logic, are based on rough set theory [11]. The (approximate) Boolean reasoning methods can be scaled to the case of compound concept approximation.

Let us consider more examples.

The prediction of behavioral patterns of a compound object evaluated over time is usually based on some historical knowledge representation used to store information about changes in relevant features or parameters. This information is usually represented as a data set and has to be collected during long-term observation of a complex dynamic system. For example, in case of road traffic, we associate the object-vehicle parameters with the readouts of different measuring devices or technical equipment placed inside the vehicle or in the outside environment (e.g., alongside the road, in a helicopter observing the situation on the road, or in a traffic patrol vehicle). Many monitoring devices serve as informative sensors, such as GPS, laser scanners, thermometers, range finders, digital cameras, radar, and image and sound converters (see, e.g., [58]). Hence, many vehicle features serve as models of physical sensors. Here are some exemplary sensors: location, speed, current acceleration or deceleration, visibility, and humidity (slipperiness) of the road. By analogy to this example, many features of compound objects

are often dubbed sensors. We discuss (see also [24]) some rough set tools for perception modeling that make it possible to recognize behavioral patterns of objects and their parts changing over time. More complex behavior of compound objects or groups of compound objects can be presented in the form of *behavioral graphs*. Any behavioral graph can be interpreted as a *behavioral pattern* and can be used as a complex classifier for recognition of complex behaviors. The complete approach to the perception of behavioral patterns, based on behavioral graphs and the dynamic elimination of behavioral patterns, is presented in [24]. The tools for dynamic elimination of behavioral patterns are used for switching off in the *system attention* procedures searching for identification of some behavioral patterns. The developed rough set tools for perception modeling are used to model networks of classifiers. Such networks make it possible to recognize behavioral patterns of objects changing over time. They are constructed using an ontology of concepts provided by experts that engage in approximate reasoning on concepts embedded in such an ontology. Experiments on data from a vehicular traffic simulator [56] are showing that the developed methods are useful in the identification of behavioral patterns.

The following example concerns human computer interfaces that allow for a dialog with experts to transfer to the system their knowledge about structurally compound objects. For pattern recognition systems [57], e.g., for optical character recognition systems it will be helpful to transfer to the system a certain knowledge about the expert's view on borderline cases. The central issue in such pattern recognition systems is the construction of classifiers within vast and poorly understood search spaces, which is a very difficult task. Nonetheless, this process can be greatly enhanced with knowledge about the investigated objects provided by a human expert. We developed a framework for the transfer of such knowledge from the expert and for incorporating it into the learning process of a recognition system using methods based on rough mereology. It is also demonstrated how this knowledge acquisition can be conducted in an interactive manner, with a large data set of handwritten digits as an example.

The next two examples are related to approximation of compound concepts in reinforcement learning and planning.

In reinforcement learning [58–60], the main task is to learn the approximation of the function $Q(s, a)$, where s, a denote a global state of the system and an action performed by an agent ag , respectively, and the real value of $Q(s, a)$ describes the reward for executing the action a in the state s . In approximation of the function $Q(s, a)$ probabilistic models are used. However, for compound real-life problems it may be hard to build such models for such a compound concept as $Q(s, a)$ [4]. We propose another approach to approximation of $Q(s, a)$ based on ontology approximation. The approach is based on the assumption that in a dialog with experts an additional knowledge can be acquired, making it possible to create a ranking of values $Q(s, a)$ for different actions a in a given state s . In the explanation given by expert about possible values of $Q(s, a)$ are used concepts from a special ontology of concepts. Next, using this ontology one can follow hierarchical learning methods to learn approximations of concepts from ontology. Such concepts can have temporal character too. This means that the ranking of actions may depend not only on the actual action and the state but also on actions performed in the past and changes caused by these actions.

In [54], a computer tool based on rough sets for supporting automated planning of the medical treatment is discussed. In this approach, a given patient is treated as an investigated complex dynamical system, while diseases of this patient (RDS, PDA, sepsis, ureaplasma, and respiratory failure) are treated as compound objects changing and interacting over time. As a measure of planning success (or failure) in experiments, a special hierarchical classifier that can predict the similarity between two plans as a number between 0.0 and 1.0 is used. This classifier has been constructed on the basis of the special ontology specified by human experts and data sets. It is important to mention that besides the ontology, experts provided the exemplary data (values of attributes) for the purpose of concepts approximation from the ontology. The methods of construction of such classifiers are based on AR schemes and are described, e.g., in [10, 23, 24, 54]. This method was used for approximation of similarity between plans generated in automated planning and plans proposed by human experts during the realistic clinical treatment.

19.4 Modeling of Compound Granules

Methods based on information systems are crucial in modeling of compound pattern granules.

Let us first recall a generalization of information systems (see, e.g., [30, 51]). For any attribute $a \in A$ of an information system (U, A) , we consider together with the value set V_a of a a relational structure \mathcal{R}_a over the universe V_a . We also consider a language \mathcal{L}_a of formulas (of the same relational signature as \mathcal{R}_a). Such formulas interpreted over \mathcal{R}_a define subsets of the Cartesian products of V_a . For example, any formula α with one free variable defines a subset $\|\alpha\|_{\mathcal{R}_a}$ of V_a . Let us observe that the relational structure \mathcal{R}_a (without functions) induces a relational structure over U . Indeed, for any k -ary relation r from \mathcal{R}_a , one can define a k -ary relation $g_a \subseteq V_a^k$ by $(x_1, \dots, x_k) \in g_a$ if and only if $(a(x_1), \dots, a(x_k)) \in r$ for any $(x_1, \dots, x_k) \in U^k$. Hence, one can consider any formula from \mathcal{L}_a as a constructive method for defining a subset of the universe U with a structure induced by \mathcal{R}_a . Any such a structure is a new information granule. On the next level of hierarchical modeling, i.e., in constructing new information systems we use such structures as objects and attributes as properties of such structures. Next, one can consider the similarity between newly constructed objects and then their similarity neighborhoods will correspond to clusters of relational structures. This process is usually more complex. This is because instead of relational structure \mathcal{R}_a we usually consider a fusion of relational structures corresponding to different attributes from A . The fusion makes it possible to describe constraints that should hold between parts obtained by composition from less compound parts. Examples of relational structures can be defined by indiscernibility, similarity, intervals obtained in discretization or symbolic value grouping, and preference or spatiotemporal relations (see, e.g., [2, 38]). One can see that parameters to be tuned in searching for relevant¹ patterns over new information systems are, among others, relational structures over value sets, the language of formulas defining parts, and constraints.

The main basic steps in hierarchical modeling are the following:

1. The structures of granules on a higher level are constructed from structures of granules on the lower level.
2. A language for expressing properties of structures on a higher level is selected.
3. Some formulas (features) of the structures on a higher level are selected as relevant for pattern granule construction.
4. Indiscernibility (or tolerance) classes defined by a newly constructed information system are used as pattern granules on the higher level.

In the following sections, we discuss in more detail some issues related to the outlined modeling (for more information, see also [61–64]).

19.4.1 Constrained Sums of Granules

One of the main task in GC is to develop calculi of granules [26, 40]. Information systems used in rough set theory are a particular kind of granules. In the section, we study operations on such granules, basic for reasoning in distributed systems of granules. The operations are called constrained sums. They are developed by interpreting infomorphisms between classifications [44]. In [65] we have shown that classifications [44] and information systems [11] are, in a sense, equivalent. Operations, called constrained sums, seem to be very important in searching for patterns in data mining (e.g., in spatiotemporal reasoning) or, in more general sense, in generating relevant granules for approximate reasoning using calculi on granules [65].

¹ For target concept approximation.

First, we recall the definition of infomorphism for two information systems [65]. The infomorphisms for classifications are introduced and studied in [44].

For all formulas $\alpha \in L_{IS}$ and for all objects $x \in U$, we will denote $x \models_{IS} \alpha$ if and only if $x \in Sem_{IS}(\alpha)$, where $Sem_{IS}(\alpha)$ denotes the semantic of (α) in IS .

Definition 9 [65]. If $IS_1 = (U_1, A_1)$ and $IS_2 = (U_2, A_2)$ are information systems then an infomorphism between IS_1 and IS_2 is a pair (f^\wedge, f^\vee) of functions $f^\wedge : L_{IS_1} \rightarrow L_{IS_2}$, $f^\vee : U_2 \rightarrow U_1$ satisfying the following equivalence:

$$f^\vee(x) \models_{IS_1} \alpha \text{ if and only if } x \models_{IS_2} f^\wedge(\alpha) \quad (9)$$

for all objects $x \in U_2$ and for all formulas $\alpha \in L_{IS_1}$.

The infomorphism will be denoted shortly by

$$(f^\wedge, f^\vee) : IS_1 \rightleftarrows IS_2.$$

Sum of Information Systems

In this section we discuss a sum of two information systems (for more information, see, e.g., [66]).

Definition 10. Let $IS_1 = (U_1, A_1)$ and $IS_2 = (U_2, A_2)$ be information systems. These information systems can be combined into a single information system, denoted by $+(IS_1, IS_2)$, with the following properties:

- The objects of $+(IS_1, IS_2)$ consist of pairs (x_1, x_2) of objects from IS_1 and IS_2 ; i.e., $U = U_1 \times U_2$.
- The attributes of $+(IS_1, IS_2)$ consist of the attributes of IS_1 and IS_2 , except that if there are any attributes in common, then we make distinct copies so as not to confuse them.

Proposition 11. There are infomorphisms $(f_k^\wedge, f_k^\vee) : IS_k \rightleftarrows (IS_1, IS_2)$ for $k = 1, 2$ defined as follows:

- $f_k^\wedge(\alpha) = \alpha_{IS_k}$ (the IS_k -copy of α) for each $\alpha \in \Sigma(IS_k)$;
- for each pair $(x_1, x_2) \in U$, $f_k^\vee((x_1, x_2)) = x_k$.

Given any information system IS_3 and infomorphisms $(f_{k,3}^\wedge, f_{k,3}^\vee) : IS_k \rightleftarrows IS_3$, there is a unique infomorphism $(f_{1+2,3}^\wedge, f_{1+2,3}^\vee) : +(IS_1, IS_2) \rightleftarrows IS_3$, such that in Figure 19.2 one can go either way around the triangles and get the same result.

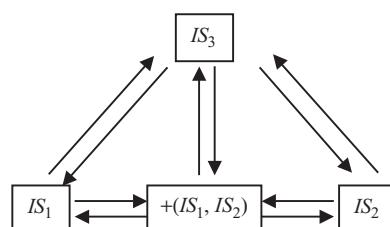


Figure 19.2 Sum of granules (information systems) IS_1 and IS_2

Table 19.1 Information system $IS_{rectangle}$ with uncertainty functions

$U_{rectangle}$	a	b	$I_a(\cdot)$	$I_b(\cdot)$	$I_{A_1}(\cdot)$
x_1	165	Yes	$\{x_1, x_3, x_5, x_6\}$	$\{x_1, x_3\}$	$\{x_1, x_3\}$
x_2	175	No	$\{x_2, x_4, x_6\}$	$\{x_2, x_4, x_5, x_6\}$	$\{x_2, x_4, x_6\}$
x_3	160	Yes	$\{x_1, x_3, x_5\}$	$\{x_1, x_3\}$	$\{x_1, x_3\}$
x_4	180	No	$\{x_2, x_4\}$	$\{x_2, x_4, x_5, x_6\}$	$\{x_2, x_4\}$
x_5	160	No	$\{x_1, x_3, x_5\}$	$\{x_2, x_4, x_5, x_6\}$	$\{x_5\}$
x_6	170	No	$\{x_1, x_2, x_6\}$	$\{x_2, x_4, x_5, x_6\}$	$\{x_2, x_6\}$

Example 12. Let us consider a diagnostic agent testing failures of the space robotic arm. Such an agent should observe the arm and detect a failure if, e.g., some of its parts are in abnormal relative position. Let us assume, in our simple example, that projections of some parts on a plane are observed and a failure is detected if the projections of some triangular or rectangular parts are in some relation; e.g., the triangle is not included sufficiently inside the rectangle. Hence, any considered object consists of parts: a triangle and a rectangle. Objects are perceived by some attributes expressing properties of parts and a relation (constraint) between them.

First, we construct an information system, called the sum of given information systems. Such system represents objects composed from parts without any constraint. It means that we consider as the universe of objects the Cartesian product of the universes of parts (Tables 19.1–19.3).

Let us consider three information systems $IS_{rectangle} = (U_{rectangle}, A_{rectangle})$, $IS_{triangle} = (U_{triangle}, A_{triangle})$, and $+(IS_{rectangle}, IS_{triangle}) = (U_{rectangle} \times U_{triangle}, \{(a, 1), (b, 1), (c, 2)\})$ presented in Table 19.1, Table 19.2, and Table 19.3, respectively. Let $U_{rectangle}$ be a set of rectangles and $A_{rectangle} = \{a, b\}$, $V_a = [0, 300]$, and $V_b = \{\text{yes, no}\}$, where the value of a means a length in millimeters of horizontal side of rectangle and for any object $x \in U_{rectangle}$ $b(x) = \text{yes}$ if and only if x is a square.

Let $U_{triangle}$ be a set of triangles and $A_{triangle} = \{c\}$ and $V_c = \{t_1, t_2\}$, where $c(x) = t_1$ if and only if x is an acute-angled triangle and $c(x) = t_2$ if and only if x is a right-angled triangle.

We assume that all values of attributes are made on a given projection plane. The results of measurements are represented in information systems. Tables 19.1 and 19.2 include only illustrative examples of the results of such measurements. We define uncertainty functions as follows:

$$\begin{aligned} y \in I_a(x) &\quad \text{if and only if } |a(x) - a(y)| \leq 5. \\ y \in I_b(x) &\quad \text{if and only if } b(x) = b(y). \\ y \in I_{A_1}(x) &\quad \text{if and only if } (y \in I_a(x) \text{ and } y \in I_b(x)). \end{aligned}$$

We assume that $(a, 1)((x_i, y_j)) = a(x_i)$, $(b, 1)((x_i, y_j)) = b(x_i)$ and $(c, 2)((x_i, y_j)) = c(y_j)$, where $i = 1, \dots, 6$ and $j = 1, 2$.

Sum of Approximation Spaces

In this section we present a simple construction of approximation space for the sum of given approximation spaces.

Table 19.2 Information system $IS_{triangle}$ with uncertainty function I_{A_2}

$U_{triangle}$	c	$I_{A_2}(\cdot)$
y_1	t_1	$\{y_1, y_3\}$
y_2	t_2	$\{y_2\}$
y_3	t_1	$\{y_1, y_3\}$

Table 19.3 An information system $+(IS_{rectangle}, IS_{triangle})$ with uncertainty function I_{A_1, A_2}

$U_{rectangle} \times U_{triangle}$	(a, 1)	(b, 1)	(c, 2)	$I_{A_1, A_2}((\cdot, \cdot))$
(x_1, y_1)	165	Yes	t_1	$\{x_1, x_3\} \times \{y_1, y_3\}$
(x_1, y_2)	165	Yes	t_2	$\{x_1, x_3\} \times \{y_2\}$
(x_1, y_3)	165	Yes	t_1	$\{x_1, x_3\} \times \{y_1, y_3\}$
(x_2, y_1)	175	No	t_1	$\{x_2, x_4, x_6\} \times \{y_1, y_3\}$
(x_2, y_2)	175	No	t_2	$\{x_2, x_4, x_6\} \times \{y_2\}$
(x_2, y_3)	175	No	t_1	$\{x_2, x_4, x_6\} \times \{y_1, y_3\}$
(x_3, y_1)	160	Yes	t_1	$\{x_1, x_3\} \times \{y_1, y_3\}$
(x_3, y_2)	160	Yes	t_2	$\{x_1, x_3\} \times \{y_2\}$
(x_3, y_3)	160	Yes	t_1	$\{x_1, x_3\} \times \{y_1, y_3\}$
(x_4, y_1)	180	No	t_1	$\{x_2, x_4\} \times \{y_1, y_3\}$
(x_4, y_2)	180	No	t_2	$\{x_2, x_4\} \times \{y_2\}$
(x_4, y_3)	180	No	t_1	$\{x_2, x_4\} \times \{y_1, y_3\}$
(x_5, y_1)	160	No	t_1	$\{x_5\} \times \{y_1, y_3\}$
(x_5, y_2)	160	No	t_2	$\{x_5\} \times \{y_2\}$
(x_5, y_3)	160	No	t_1	$\{x_5\} \times \{y_1, y_3\}$
(x_6, y_1)	170	No	t_1	$\{x_2, x_6\} \times \{y_1, y_3\}$
(x_6, y_2)	170	No	t_2	$\{x_2, x_6\} \times \{y_2\}$
(x_6, y_3)	170	No	t_1	$\{x_2, x_6\} \times \{y_1, y_3\}$

Let $AS_{\#_k} = (U_k, I_{\#_k}, v_{SRI})$ be an approximation space for information system IS_k , where $k = 1, 2$. We define an approximation space $+(AS_{\#_1}, AS_{\#_2})$ for information system $+(IS_1, IS_2)$ as follows:

- The universe is equal to $U_1 \times U_2$.
- $I_{\#_1, \#_2}((x_1, x_2)) = I_{\#_1}(x_1) \times I_{\#_2}(x_2)$.
- The inclusion relation v_{SRI} in $+(AS_{\#_1}, AS_{\#_2})$ is the standard inclusion function.

We have the following property:

Proposition 13.

$$LOW(+AS_{\#_1}, X \times Y) = LOW(AS_{\#_1}, X) \times LOW(AS_{\#_2}, Y) \quad (10)$$

$$UPP(+AS_{\#_1}, X \times Y) = UPP(AS_{\#_1}, X) \times UPP(AS_{\#_2}, Y). \quad (11)$$

Proof. We have $I_{\#_1, \#_2}((x_1, x_2)) \subseteq X \times Y$ iff $I_{\#_1}(x_1) \subseteq X$ and $I_{\#_2}(x_2) \subseteq Y$. Moreover, $I_{\#_1, \#_2}((x_1, x_2)) \cap (X \times Y) \neq \emptyset$ iff $I_{\#_1}(x_1) \cap X \neq \emptyset$ and $I_{\#_2}(x_2) \cap Y \neq \emptyset$. ■

Example 14. For information system $IS_{rectangle}$, we define an approximation space $AS_{A_1} = (U_{rectangle}, I_{A_1}, v_{SRI})$ such that

$$y \in I_a^5(x) \text{ if and only if } |a(x) - a(y)| \leq 5.$$

This means that rectangles x and y are similar with respect to the length of horizontal sides if and only if the difference in lengths is not greater than 5 mm.

Let $y \in I_b(x)$ if and only if $b(x) = b(y)$ and $y \in I_{A_1}(x)$ if and only if $\forall c \in A_1 y \in I_c(x)$.

Thus, we obtain uncertainty functions represented in the last three columns of Table 19.1.

For the information system $IS_{triangle}$, we define an approximation space as follows:

$$y \in I_{A_2}(x) \text{ if and only if } c(x) = c(y)$$

(see the last column of Table 19.1).

For $+_{(IS_{rectangle}, IS_{triangle})}$, we obtain $I_{A_1, A_2}((x, y)) = I_{A_1}(x) \times I_{A_2}(y)$ (see the last column of Table 19.3).

Sum with Constraints of Information Systems

In this section, we consider a new operation on information systems often used in searching, e.g., for relevant patterns. We start from the definition in which the constraints are given explicitly.

Definition 15. Let $IS_i = (U_i, A_i)$, $i = 1, \dots, k$, be information systems and let R be a k -ary constraint relation in $U_1 \times \dots \times U_k$, i.e., $R \subseteq U_1 \times \dots \times U_k$. These information systems can be combined into a single information system relatively to R , denoted by $+_R(IS_1, \dots, IS_k)$, with the following properties:

- The objects of $+_R(IS_1, \dots, IS_k)$ consist of k -tuples (x_1, \dots, x_k) of objects from R , i.e., all objects from $U_1 \times \dots \times U_k$ satisfying the constraint R .
- The attributes of $+_R(IS_1, \dots, IS_k)$ consist of the attributes from the sets A_1, \dots, A_k , except that if there are any attributes in common, then we make distinct copies so as not to confuse them.

Usually the constraints are defined by conditions expressed by Boolean combination of descriptors of attributes. It means that the constraints are built from expressions $a = v$, where a is an attribute and v is its value, using propositional connectives \wedge , \vee , and \neg . Observe that in constraint definition we use not only attributes of parts (i.e., from information systems IS_1, \dots, IS_k) but also some other attributes specifying relation between parts. In our example (see Table 19.4), the constraint R_1 is defined as follows: *the triangle is sufficiently included in the rectangle*. Any row of this table represents an object (x_i, y_j) composed of the triangle y_j included sufficiently into the rectangle x_i .

Let us also note that constraints are defined using primitive (measurable) attributes different than those from information systems describing parts. This makes the sum with constraint operation different from theta join in databases. On the other hand one can consider that the constraints are defined in two steps. In the first step we extend the attributes for parts and in the second step we define the constraints using some relations on these new attributes.

Let us observe that the information system $+_R(IS_1, \dots, IS_k)$ can also be described using an extension of the sum $+_{(IS_1, \dots, IS_k)}$ by adding a new binary attribute that is the characteristic function of the relation R and by taking a subsystem of the received system consisting of all objects having value 1 for this new attribute.

Table 19.4 Information system $+_{R_1}(IS_{rectangle}, IS_{triangle})$

$(U_{rectangle} \times U_{triangle}) \cap R_1$	a'	b'	c'
(x_1, y_1)	165	Yes	t_1
(x_1, y_2)	165	Yes	t_2
(x_2, y_1)	175	No	t_1
(x_2, y_2)	175	No	t_2
(x_3, y_1)	160	Yes	t_1
(x_3, y_2)	160	Yes	t_2
(x_4, y_1)	180	No	t_1
(x_4, y_2)	180	No	t_2
(x_5, y_1)	160	No	t_1
(x_5, y_2)	160	No	t_2
(x_6, y_1)	170	No	t_1
(x_6, y_2)	170	No	t_2

The constraints used to define the sum (with constraints) can often be specified by information systems. The objects of such systems are tuples consisting of objects of information systems that are arguments of the sum. The attributes describe relations between elements of tuples. One of the attribute is a characteristic function of the constraint relation (restricted to the universe of the information system). In this way, we obtain a decision system with the decision attribute defined by the characteristic function of the constraint and conditional attributes are the remaining attributes of this system. From such decision table one can induce classifier for the constraint relation. Next, the classifier can be used to select tuples in the construction of the sum with constraints.

Example 16. Let us consider three information systems

$$\begin{aligned} IS_{rectangle} &= (U_{rectangle}, A_{rectangle}), IS_{triangle} = (U_{triangle}, A_{triangle}), \\ &+_{R_1}(IS_{rectangle}, IS_{triangle}), \end{aligned}$$

presented in Table 19.1, Table 19.2, and Table 19.4, respectively. We assume that $R_1 = \{(x_i, y_j) \in U_{rectangle} \times U_{triangle} : i = 1, \dots, 6 \text{ and } j = 1, 2\}$. We also assume that $a'((x_i, y_j)) = a(x_i)$, $b'((x_i, y_j)) = b(x_i)$ and $c'((x_i, y_j)) = c(y_j)$, where $i = 1, \dots, 6$ and $j = 1, 2$.

The above examples are illustrating an idea of specifying constraints by examples. Table 19.4 can be used to construct a decision table partially specifying characteristic functions of the constraint. Such a decision table should be extended by adding relevant attributes related to the object parts, which allows to induce the high-quality classifiers for the constraint relation. The classifier can then be used to filter composed pairs of objects that satisfy the constraint. This is an important construction because the constraint specification usually cannot be defined directly in terms of measurable attributes. It can be specified, e.g., in natural language. This is the reason that the process of inducing of the relevant classifiers for constraints can require hierarchical classifier construction [13].

The constructed constraint sum of information systems can contain some incorrect objects. This is due to improper filtering of objects by the constraint classifier induced from data (with accuracy usually less than 100%). One should take this issue into account in constructing nets of information systems.

Constraint Sum of Approximation Spaces

Let $AS_{\#_i} = (U_i, I_{\#_i}, v_{SRI})$ be an approximation space for information system IS_i , where $i = 1, \dots, k$, and let $R \subseteq U_1 \times \dots \times U_k$ be a constraint relation. We define an approximation space $+_R(AS_{\#_1}, \dots, AS_{\#_k})$ for $+_R(IS_1, \dots, IS_k)$ as follows:

- The universe is equal to R .
- $I_{\#_1, \dots, \#_k}((x_1, \dots, x_k)) = (I_{\#_1}(x_1) \times \dots \times I_{\#_k}(x_k)) \cap R$.
- The inclusion relation v_{SRI} in $+_R(AS_{\#_1}, \dots, AS_{\#_k})$ is the standard inclusion function.

We have the following properties of approximations:

Proposition 17.

$$\begin{aligned} LOW(+_R(AS_{\#_1}, \dots, AS_{\#_k}), X_1 \times \dots \times X_k) \\ = R \cap (LOW(AS_{\#_1}, X_1) \times \dots \times LOW(AS_{\#_k}, X_k)) \end{aligned} \quad (12)$$

$$\begin{aligned} UPP(+_R(AS_{\#_1}, \dots, AS_{\#_k}), X_1 \times \dots \times X_k) \\ = R \cap (UPP(AS_{\#_1}, X_1) \times \dots \times UPP(AS_{\#_k}, X_k)). \end{aligned} \quad (13)$$

19.5 Rough-Fuzzy Granules

In this section, we discuss rough-fuzzy granules. Such granules are important for many applications. To explain the main idea behind such granules let us consider the problem of construction of a classifier

for a vague concept specified by a sample of positive and negative examples. Quite often, the induced boundary region of the concept can be too large and later the information that a given object falls into the boundary region may not be that meaningful in applications. In such cases, one can try to distinguish different parts in the boundary region representing different *shades* of the concept. Next, with these parts being treated as new concepts, their approximations are constructed. For applications, it is very important to have linearly ordered parts, called layers. The boundary regions of layers should satisfy the constraint that each of them has non-empty intersection with two neighboring layers only. Next, approximations of layers are extended from a given sample to the whole space of objects. The induced membership functions for the parts can be treated as rough-fuzzy membership functions of linguistic variables [67] corresponding to parts (e.g., *low*, *medium*, and *high*). In this way, we obtain a family of classifiers as an approximation of a given concept. We call such a family a rough-fuzzy granule.

Below we present a more formal description of the above idea.

Let $DT = (U, A, d)$ be a decision table, where the decision d is the fuzzy membership function ν restriction to the objects from U . Consider reals $0 < c_1 < \dots < c_k$, where $c_i \in (0, 1]$ for $i = 1, \dots, k$. Any c_i defines c_i -cut by $X_i = \{x \in U : \nu(x) \geq c_i\}$. Assume that $X_0 = U$ and $X_{k+1} = X_{k+2} = \emptyset$. A *rough-fuzzy granule* (*rf-granule*, for short) corresponding to (DT, c_1, \dots, c_k) is any granule $g = (g_0, \dots, g_k)$, such that for some $B \subseteq A$,

$$Sem_B(g_i) = [LOW(AS_B, (X_i - X_{i+1})), UPP(AS_B, (X_i - X_{i+1}))], \quad \text{for } i = 0, \dots, k, \quad (14)$$

and

$$UPP(AS_B, (X_i - X_{i+1})) \subseteq (X_{i-1} - X_{i+2}), \quad \text{for } i = 1, \dots, k,$$

where $Sem_B(g_i)$ denotes the semantics of g_i .

Any function $\nu^* : U \rightarrow [0, 1]$ satisfying the conditions

$$\begin{aligned} \nu^*(x) &= 0, & \text{for } x \in U - UPP(AS_B, X_1), \\ \nu^*(x) &= 1, & \text{for } x \in LOW(AS_B, X_k), \\ \nu^*(x) &= c_{i-1}, & \text{for } x \in LOW(AS_B, (X_{i-1} - X_i)) \text{ and } i = 2, \dots, k-1, \\ c_{i-1} < \nu^*(x) < c_i, & \text{for } x \in (UPP(AS_B, X_i) - LOW(AS_B, X_i)), \\ &\text{where } i = 1, \dots, k \text{ and } c_0 = 0, \end{aligned} \quad (15)$$

is called a B -approximation of ν .

For applications, it is necessary to develop heuristics searching for relevant attributes and parts as well as their approximations. The constructed rough-fuzzy granules are used, e.g., in approximation of other concepts.

19.6 Conclusion

We have discussed some issues for intelligent systems based on GC. The most important are applications of granules for compound concept approximation.

The approach can be extended to adaptive approximation of concepts in multiagent systems, e.g., used in control of complex adaptive systems.

Acknowledgments

The research was supported by the grant N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland and by the grant Innovative Economy Operational Programme 2007–2013 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland.

References

- [1] J. Fredman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*. Springer-Verlag, Heidelberg, 2001.
- [2] W. Kloesgen and J. Źytkow (eds). *Handbook of Knowledge Discovery and Data Mining*. Oxford University Press, Oxford, 2002.
- [3] L. Breiman. Statistical modeling: The two cultures. *Stat. Sci.* 16(3) (2001) 199–231.
- [4] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
- [5] T. Poggio and S. Smale. The mathematics of learning: Dealing with data. *Not. AMS* 50(5) (2003) 537–544.
- [6] M. Fahle and T. Poggio (eds). *Perceptual Learning*. MIT Press, Cambridge, 2002.
- [7] M. Luck, P. McBurney, and Ch. Preist. *Agent Technology: Enabling Next Generation Computing: A Roadmap for Agent Based Computing*, AgentLink, 2003.
- [8] M. Gell-Mann. *The Quark and the Jaguar – Adventures in the Simple and the Complex*. Little, Brown and Co., London, 1994.
- [9] P. Stone. *Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer*. MIT Press, Cambridge, 2000.
- [10] S.H. Nguyen, J. Bazan, A. Skowron, and H.S. Nguyen. Layered learning for concept synthesis. In: *Transactions on Rough Sets I, LNCS3100*. Springer, Heidelberg, 2004, pp. 187–208.
- [11] Z. Pawlak. *Rough Sets. Theoretical Aspects of Reasoning about Data*. Kluwer, Dordrecht, 1991.
- [12] L. Polkowski and A. Skowron. Rough mereology: A new paradigm for approximate reasoning. *J. Approx. Reason.* 15(4) (1996) 333–365.
- [13] S.K. Pal, L. Polkowski, and A. Skowron (eds). *Rough-Neural Computing: Techniques for Computing with Words*. Springer-Verlag, Berlin, 2004.
- [14] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer, Dordrecht, 2003.
- [15] W. Pedrycz (ed). *Granular Computing*. Physica-Verlag, Heidelberg, 2001.
- [16] L.A. Zadeh. A new direction in AI: Toward a computational theory of perceptions. *AI Mag.* 22(1) (2001) 73–84.
- [17] J.R. Hobbs. Granularity. In: *Proceedings of Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985, pp. 432–435. Also in *Readings in Qualitative Reasoning about Physical Systems*, D.S. Weld and J. de Kleer (eds), Morgan Kaufmann Publishers, San Mateo, CA, 1989, pp. 542–545.
- [18] J.R. Hobbs. Half orders of magnitude. In: *KR-2000 Workshop on Semantic Approximation, Granularity, and Vagueness*, Breckenridge, CO, April 2000.
- [19] J.R. Hobbs and V. Kreinovich. Optimal choice of granularity in commonsense estimation: Why half orders of magnitude. In: *Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, Vancouver, British Columbia, July 2001, pp. 1343–1348.
- [20] L.A. Zadeh. Outline of a new approach to the analysis of complex system and decision processes. *IEEE Trans. Syst. Man Cybern. SMC3* (1973) 28–44.
- [21] L.A. Zadeh. Fuzzy sets and information granularity. In: M. Gupta, R. Ragade, and R. Yager (eds), *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, 1979, pp. 3–18.
- [22] L.A. Zadeh. From computing with numbers to computing with words – From manipulation of measurements to manipulation of perceptions. *IEEE Trans. Circuits Syst.* 45 (1999) 105–119.
- [23] J. Bazan and A. Skowron. Classifiers based on approximate reasoning schemes. In: B. Dunin-Keplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds), *Monitoring, Security, and Rescue Tasks in Multiagent Systems MSRAS, Advances in Soft Computing*. Springer, Heidelberg, 2005, pp. 191–202.
- [24] J. Bazan, J.F. Peters, and A. Skowron. Behavioral pattern identification through rough set modeling. In: *Proceedings of RSFDGrC'2005*, LNAI 3641. Springer, Heidelberg, 2005, pp. 688–697.
- [25] J. Bazan, H.S. Nguyen, S.H. Nguyen, and A. Skowron. Rough set methods in approximation of hierarchical concepts. In: *Proceedings of RSCTC'2004*, LNAI 3066. Springer, Heidelberg, 2004, pp. 346–355.
- [26] A. Skowron and J. Stepaniuk. Information granules and rough-neural computing. In: S.K. Pal, L. Polkowski, and A. Skowron (eds), *Rough-Neural Computing: Techniques for Computing with Words*. Springer-Verlag, Berlin, 2004, pp. 43–84.
- [27] J. Rissanen. Modeling by shortest data description. *Automatica* 14 (1978) 465–471.
- [28] J. Rissanen. Minimum-description-length principle. In: S. Kotz and N. Johnson (eds), *Encyclopedia of Statistical Sciences*. John Wiley & Sons, New York, 1985, pp. 523–527.
- [29] Z. Pawlak and A. Skowron. Rudiments of rough sets. *Inf. Sci. Int. J.* 177(1) (2007) 3–27.
- [30] Z. Pawlak and A. Skowron. Rough sets: Some extensions. *Inf. Sci. Int. J.* 177(1) (2007) 28–40.
- [31] Z. Pawlak and A. Skowron. Rough sets and Boolean reasoning. *Inf. Sci. Int. J.* 177(1) (2007) 41–73.
- [32] G. Frege. *Grundlagen der Arithmetik 2*. Verlag von Herman Pohle, Jena, 1893.

- [33] S. Leśniewski. Grunzüge eines neuen Systems der Grundlagen der Mathematik. *Fundam. Mate.* XIV (1929) 1–81.
- [34] J. Łukasiewicz. Die logischen Grundlagen der Wahrscheinlichkeitsrechnung, Kraków 1913. In: L. Borkowski (ed), *Jan Łukasiewicz—Selected Works*. North-Holland, Amstardam, Polish Scientific Publishers, Warsaw, 1970.
- [35] A. Skowron. Rough sets in KDD (plenary lecture). In: Z. Shi, B. Faltings, and M. Musen (eds), *16-th World Computer Congress (IFIP'2000): Proceedings of Conference on Intelligent Information Processing (IIP'2000)*. Publishing House of Electronic Industry, Beijing, 2000, pp. 1–17.
- [36] H.S. Nguyen. Approximate boolean reasoning: Foundations and applications in data mining. In: J.F. Peters and A. Skowron (eds), *Transactions on Rough Sets V: Journal Subline, Lecture Notes in Computer Science 4100*. Springer, Heidelberg, 2006, pp. 344–523.
- [37] RSES. logic.mimuw.edu.pl/~rses/, accessed January 24, 2008.
- [38] A. Skowron and J. Stepaniuk. Tolerance approximation spaces. *Fundam. Inf.* 27 (1996) 245–253.
- [39] J. Stepaniuk. Tolerance information granules. In: B. Dunin-Keplicz, A. Jankowski, A. Skowron, and M. Szczęska (eds), *Monitoring, Security and Rescue Techniques in Multiagent Systems*. Springer, Berlin, 2005, pp. 305–316.
- [40] A. Skowron and J. Stepaniuk. Information granules: Towards foundations of granular computing. *Int. J. Intell. Syst.* 16(1) (2001) 57–86.
- [41] A. Skowron. Approximate reasoning in distributed environments. In: N. Zhong and J. Liu (eds), *Intelligent Technologies for Information Analysis*. Springer, Heidelberg, 2004, pp. 433–474.
- [42] S. Staab and R. Studer (eds). *Handbook on Ontologies. International Handbooks on Information Systems*. Springer, Heidelberg, 2004.
- [43] A. Kandel and M. Last (eds). *Advances in fuzzy logic*. *Inf. Sci. Int. J.* 177(2) (2007) 329–331.
- [44] J. Barwise and J. Seligman. *Information Flow: The Logic of Distributed Systems*. Cambridge University Press Tracts in Theoretical Computer Science 44, Cambridge, UK, 1997.
- [45] J. Bazan, A. Skowron, and R. Swiniarski. Rough sets and vague concept approximation: From sample approximation to adaptive learning. In: *Transactions on Rough Sets V: LNCS Journal Subline*, LNCS 4100. Springer, Heidelberg, 2006, pp. 39–62.
- [46] J. Stepaniuk, J. Bazan, and A. Skowron. Modelling complex patterns by information systems. *Fundam. Inf.* 67(1–3) (2005) 203–217.
- [47] J. Stepaniuk, A. Skowron, J.F. Peters, and R. Swiniarski. Calculi of approximation spaces. *Fundam. Inf.* 72(1–3) (2006) 363–378.
- [48] J. Stepaniuk. Knowledge discovery by application of rough set models. In: L. Polkowski, S. Tsumoto, and T.Y. Lin (eds), *Rough Set Methods and Applications. New Developments in Knowledge Discovery in Information Systems*. Physica-Verlag, Heidelberg, 2000, pp. 137–233.
- [49] A. Skowron, R. Swiniarski, and P. Synak. Approximation spaces and information granulation. In: *Transactions on Rough Sets III: LNCS Journal Subline*, LNCS 3400. Springer, Heidelberg, 2005, pp. 175–189.
- [50] A. Skowron and P. Synak. Complex patterns. *Fundam. Inf.* 60(1–4) (2004) 351–366.
- [51] L. Polkowski. *Rough Sets: Mathematical Foundations. Advances in Soft Computing*. Physica-Verlag, Heidelberg, 2002.
- [52] W. Ziarko. Variable precision rough set model. *J. Comput. Syst. Sci.* 46 (1993) 39–59.
- [53] J. Stepaniuk. Rough relations and logics. In: L. Polkowski and A. Skowron (eds), *Rough Sets in Knowledge Discovery I. Methodology and Applications*. Physica-Verlag, Heidelberg, 1998, pp. 248–260.
- [54] J. Bazan, P. Kruczak, S. Bazan-Socha, A. Skowron, and J.J. Pietrzyk. Risk pattern identification in the treatment of infants with respiratory failure through rough set modeling. In: *Proceedings of IPMU'2006, Paris, France, July 2–7, 2006*. Éditions E.D.K., Paris, 2006, pp. 2650–2657.
- [55] C. Urmson, J. Anhalt, M. Clark, et al. *High Speed Navigation of Unrehearsed Terrain: Red Team Technology for Grand Challenge 2004*. Technical Report CMU-RI-TR-04-37. Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, June 2004.
- [56] The Road simulator. <http://duch.mimuw.edu.pl/~bazan/simulator>, accessed January 24, 2008.
- [57] R. Duda, P. Hart, and R. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2002.
- [58] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
- [59] T.G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Artif. Intell.* 13(5) (2000) 227–303.
- [60] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *J. Artif. Intell. Res.* 4 (1996) 227–303.
- [61] H.S. Nguyen, A. Skowron, and J. Stepaniuk. Granular computing: A rough set approach. *Comput. Intell. Int. J.* 17(3) (2001) 514–544.
- [62] A. Skowron and J. Stepaniuk. Information granule decomposition. *Fundam. Inf.* 47 (3–4) (2001) 337–350.

- [63] A. Skowron and J. Stepaniuk. Information granules: Towards foundations for spatial and temporal reasoning. *Proc. Indian Nat. Sci. Acad.* 67A(2) (2001) 315–325.
- [64] A. Skowron, J. Stepaniuk, and J.F. Peters. Towards discovery of relevant patterns from parameterized schemes of information granule construction. In: M. Inuiguchi, S. Hirano, and S. Tsumoto (eds), *Rough Set Theory and Granular Computing*. Springer, Berlin, 2003, pp. 97–108.
- [65] A. Skowron, J. Stepaniuk, and J.F. Peters. Rough sets and infomorphisms: Towards approximation of relations in distributed environments. *Fundam. Inf.* 54(1–2) (2003) 263–277.
- [66] A. Skowron and J. Stepaniuk. Constrained sums of information systems. In: *Proceedings of the RSCTC 2004*, LNCS 3066. Springer, Heidelberg, 2004, pp. 300–309.
- [67] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Part I Inf. Sci.* 8, (1975) 199–249; *Part II Inf. Sci.* 8, (1975) 301–357; *Part III Inf. Sci.* 9 (1975) 43–80.

20

Construction of Rough Information Granules

Anna Gomolińska

20.1 Introduction

According to Zadeh, who introduced the term ‘information granule’ into a fuzzy-set analysis of complex systems [1–3], an *information granule* (*infogranule* in short) is a clump of objects of some sort, drawn together on the basis of indistinguishability, similarity, or functionality. In this definition, general enough to comprise a large number of special cases is the stress laid on the reasons for clustering objects into clumps, and three such motives are suggested: the forementioned indistinguishability, similarity, and functionality.¹ Only symbolic objects are allowed to form infogranules in our approach. For example, they may be items in a database, points and sets of points of a space, data tables and rows in such tables, various mathematical structures including information systems, text documents, single instructions and algorithms, and identifiers/names of abstract objects as well as physical entities as human agents and their coalitions, cars, apples, trees in a forest, and so forth.

The notion of an infogranule is related to such notions as cluster, concept, (distributive or collective) class, and complex object, to mention a few. It directly follows from the definition that infogranules are clusters. Obviously, not every cluster is an infogranule, and to see this consider an economic cluster.² Concept is understood as a distributive class (in particular, set) of entities sharing some properties. For instance, the concept of being blue consists of all entities perceived as blue. From this perspective, concepts consisting of symbolic objects are similarity-based infogranules. Infogranules can be distributive classes like sets of similar items in a data table or collective classes like the class of procedures constituting an algorithm. In general, the notion of a class is broader than that of an infogranule. A geographical region is a collective class but not an infogranule. The case of complex object is analogous. Every infogranule

¹ Indistinguishability, viewed as a special case of similarity of objects, will be omitted unless necessary. Moreover, the term ‘indiscernability’ will be preferred to ‘indistinguishability.’

² Economic cluster is a vague concept related to social and economic phenomenon that in spite of globalization of market and excellent opportunities for making businesses via Internet, geographical location is still fundamental to competition of companies. According to Porter, who introduced and studied economic clusters, ‘[c]lusters are geographic concentrations of interconnected companies and institutions in a particular field’ [4–6]. The world well-known clusters are the California wine cluster, Silicon Valley, Hollywood, the Italian leather fashion cluster, the Dutch transportation and flower clusters, and the cluster of built-in kitchens and appliances in Germany, to name a few.

may be viewed as a complex object but the opposite is not true. A human being is a complex object but definitely not an infogranule.

Apart from similarity (subsuming indiscernability) and functionality, other relationships among objects constituting an infogranule may explicitly be taken into account. Thus, from the mathematical perspective, infogranules can be ordered sets (in particular, tuples of objects), relations, graphs, data structures (stacks, tables, etc.), abstract algebras and various algebraic structures, topological spaces, and so on. For instance, an algorithm, being a functionality-based infogranule as a set of instructions drawn together for solving a particular problem, is usually a partially ordered set of instructions and procedures and not merely a flat collection of instructions. In spite of the fact that the notion of an infogranule is very broad, granular computing is a promising, useful, and sometimes the only feasible approach to complex problem solving, especially when approximate solutions suffice. It should be emphasized that – given a problem to be solved – only such infogranules are taken into account which satisfy some requirements or, in other words, some specification.

The phenomenon of granularity of a space of entities is often observed in the real life. By nature, our knowledge about objects is usually incomplete, imprecise, and uncertain. Moreover, although the language used is rich and flexible, words often have vague meanings. For these and other reasons as, e.g., the need of effectiveness and efficiency of (inter)action, objects have finite, often vague and erroneous, descriptions. Clearly, different objects can be described in a similar or even the same way. From an observer's perspective, they are similar or indiscernible, respectively. Assuming rationality, indistinguishable objects should be treated equally, and similar objects should be dealt with in a similar way. Therefore, not only we perceive objects together with 'clouds' of objects indiscernible or similar to them but we also take into account these 'clouds' when operating on the objects. The well-known examples of indiscernability-based infogranules are equivalence classes of some equivalence relations, e.g., a set of indiscernible rows in a Pawlak's information system, an indefinite integral being a family of all functions having the same derivative, and a direction as the set of all straight lines parallel to one another. For any reflexive relation of similarity ϱ on a universe U and any object $u \in U$, the set of objects similar to u and the set of objects to which u is similar are instances of similarity-based infogranules. Also functionality-based infogranules widely occur in theory and the real life. These are, for example, operating procedures, codes of conduction, algorithms, computer programs, (rough) classifiers, models of computational grids,³ plans and schedules, rule complexes representing agents and games [7–10], and symbolic representations of groups of people, coalitions and teams of cooperating agents, swarms, institutions, economic clusters, and so forth.

A fundamental idea of granular computing [2, 3, 11–23] is that instead of or apart from performing actions on single objects, we operate on infogranules. At first glance, computing with infogranules may seem to be simpler [24], and it is really so in some cases. For example, instead of considering any real value from a given interval $[a, b]$, it can be easier to take into account only several values a_0, \dots, a_n obtained by discretization of the interval (see, e.g., [25, 26] for the discretization problem within the rough-set framework). In general – however – one should not expect computations with infogranules to be much easier than computations with objects constituting these granules. As a matter of fact, new problems arise, e.g., nearness of infogranules [27–29]. Nonetheless, taking into account granularity of the universe of objects investigated is a necessity when (inter)acting under vague, incomplete, and uncertain information.

In this chapter we discuss the problem of construction of an infogranule satisfying a given specification. A specification is understood as a non-empty finite set of formulas of a knowledge representation language, describing some desired properties of the infogranule being constructed. Such a description can specify what constituents of the infogranule in question should or should not be, how the elements should be related to one another, how the architecture of the infogranule should look like, and general properties concerning the constructed infogranule and the process of construction. Hence, two major tasks can be identified: (i) searching for objects satisfying certain requirements, including the relationships among the objects, and (ii) composition of an infogranule from such objects in compliance with the specification.

³ That is, collections of distributed resources used to execute large-scale computations.

Both (i) and (ii) should be accomplished in a way consistent with general postulates regarding who may do what and how, i.e., obeying constraints on the size of the infogranule, the time and the terms of realization of the task, the acceptable level of risk of error, the quality of approximate solutions, the norms to be complied with, and so on. Satisfiability of formulas and sets of formulas by objects is the key issue in both (i) and (ii) and will be of a particular interest in our study. It should be emphasized that concepts involved in satisfiability of a specification are usually known only partially by a number of positive and negative examples. Therefore, reasoning about such concepts and, in particular, judgment of membership in these concepts have to be based on approximate, soft methods. Our perspective is the rough-set one, yet many observations seem to be general enough to hold true within other soft-computing frameworks.

The rest of this chapter is organized as follows. Pawlak's information systems are recalled in Section 20.2. In the same section are several basic instances of infogranules described. Rough approximation spaces are overviewed in Section 20.3. Section 20.4 is devoted to labeling infogranules by formulas and rules of a knowledge representation language. Among other things, extensions of formulas and sets of formulas are viewed as infogranules. In Section 20.5 are various rough forms of satisfiability of formulas and sets of formulas by objects recalled. In Section 20.6 is a search for objects satisfying a given specification discussed. Construction of infogranules from such objects is addressed in Section 20.7. The last section contains final remarks.

20.2 Pawlak's Information Systems

Systems for representation of information, referred to as *information systems* (or simply *infosystems*) and briefly recalled in this section, were introduced by Zdzisław Pawlak in the early eighties of the last century [30–33]. To tell the truth, only the case of a non-deterministic infosystem, possibly with missing values, is considered here. Such a system may be viewed as a triple $\mathcal{A} = (U, A, \tau)$ where U – the universe of \mathcal{A} – is a non-empty finite set of objects, A is a non-empty finite set of attributes, and τ is an *information mapping* assigning to every pair $(u, a) \in U \times A$ a set of values. For every attribute a , let $V_a = \{\tau(u, a) \mid u \in U\}$ and $V_A = \{V_a \mid a \in A\}$. Objects, attributes, attribute values, and sets of values will be denoted by u, a, v , and V , possibly with sub/superscripts. For simplicity, we will not make a distinction between entities and their names unless necessary. $\tau(u, a) = V$ informally reads as ‘the value of a at u belongs to V .’ In particular, $\tau(u, a) = \emptyset$ means that no information about the value of a at u is available in the system; i.e., this value is missing.

Where for every object u and every attribute a , $\tau(u, a)$ contains at most one element, \mathcal{A} is *deterministic*; otherwise the infosystem is *non-deterministic*. Each attribute a is, in fact, a partial mapping on U . If all attribute values are given, $a : U \mapsto \bigcup V_a$ will be a mapping. When \mathcal{A} is deterministic and all attributes are mappings, τ will become useless. Such infosystems, referred to as *basic* here, are often defined as pairs of the form (U, A) . Apart from basic infosystems, Pawlak introduced himself multivalued infosystems⁴ and probabilistic infosystems [32]. Non-deterministic infosystems, also described by Pawlak in [32], were studied by Lipski in [34, 35]. Infosystems with missing values were investigated, e.g., in [36–39]. Generally, various forms of non-determinism in infosystems were discussed in [40].

A *decision system* is an infosystem $\mathcal{A} = (U, A, \tau)$ where the set of attributes A is divided into two non-empty disjoint sets of *condition* and *decision* attributes, C and D , respectively. Observe that \mathcal{A} is the result of merging of two infosystems: $(U, C, \tau_{|U \times C})$ and $(U, D, \tau_{|U \times D})$. Decision attributes are customarily denoted by d , possibly with sub/superscripts. For simplicity, assume that $(U, D, \tau_{|U \times D})$ is deterministic. In this way, for every decision attribute d , objects of U are partially classified into a finite number of classes determined by values of d .

An infosystem is often represented in the form of a table called an *information table*, and a similar representation of a decision system is known as a *decision table*. In infosystems is the whole information about an object u given by the set $\{(a, \tau(u, a)) \mid a \in A\}$. On the other hand, attributes are ordered in

⁴ An attribute a may merely be a binary relation on U and $\bigcup V_a$ in multivalued infosystems.

information tables, so information about u provided by such a table is a sequence of values $\tau(u, a)$, for every $a \in A$.

In an infosystem \mathcal{A} induces each non-empty⁵ $B \subseteq A$ an equivalence relation, ind_B , such that for any objects u, u' ,

$$(u, u') \in \text{ind}_B \stackrel{\text{def}}{\Leftrightarrow} \forall a \in B. \tau(u, a) = \tau(u', a). \quad (1)$$

ind_B , called the *B-indiscernability* relation, assigns to every object u its equivalence class $[u]_B$ consisting of all objects indiscernible from u by means of attributes of B . Equivalence classes of ind_B are referred to as the *elementary B-infogranules*. Where $\wp U$ denotes the power set of U , the mapping $[\cdot]_B : U \mapsto \wp U$ producing elementary *B-infogranules* from objects of U is an instance of an *uncertainty* mapping [41].

Now we briefly recall the *descriptor language* for an infosystem \mathcal{A} , $L_{\mathcal{A}}$, which is a simple language to reason about properties of objects and *concepts* (i.e., sets of objects) of \mathcal{A} [30, 33]. Let $\text{FOR}_{\mathcal{A}}$ denote the set of all formulas of $L_{\mathcal{A}}$. Two sorts of constant symbols – being the only terms – occur in the language: attribute names and names of sets of values. For simplicity, let \emptyset be the constant symbol for the empty set. \wedge (conjunction) and \neg (negation) are taken as primitive propositional connectives, whereas \vee (disjunction), \rightarrow (material implication), and \leftrightarrow (double implication) are defined by means of \wedge, \neg along the classical lines. By assumption, the logical and metalogical connectives of conjunction and disjunction will take the precedence of implication and double implication. Formulas are denoted by α and β with sub/superscripts if needed. Atomic formulas are pairs of terms (a, V) called (*generalized*) *descriptors* where a is an attribute name and V is a name of a set of values occurring in \mathcal{A} . Compound formulas are formed from descriptors and propositional connectives as usual.

Crisp (or *c-*) *satisfiability* of formulas and sets of formulas by objects, \models_c , is defined as follows, for any descriptor (a, V) where $V \neq \emptyset$, any formulas α, β , any set of formulas X , and an arbitrary object u :

$$\begin{aligned} u \models_c (a, V) &\stackrel{\text{def}}{\Leftrightarrow} \emptyset \neq \tau(u, a) \subseteq V, & u \models_c (a, \emptyset) &\stackrel{\text{def}}{\Leftrightarrow} \tau(u, a) = \emptyset, \\ u \models_c \alpha \wedge \beta &\stackrel{\text{def}}{\Leftrightarrow} u \models_c \alpha \ \& \ u \models_c \beta, \\ u \models_c \neg \alpha &\stackrel{\text{def}}{\Leftrightarrow} u \not\models_c \alpha, \\ u \models_c X &\stackrel{\text{def}}{\Leftrightarrow} \forall \alpha \in X. u \models_c \alpha. \end{aligned} \quad (2)$$

Thus, in particular, u will *c-satisfy* (a, V) if the values of τ at u and a belong to V . With every formula or a set of formulas Φ , there is an infogranule of all objects *c-satisfying* Φ associated, called the *crisp* (or *c-*) *extension* of Φ and denoted by $||\Phi||_c$. Hence,

$$\begin{aligned} ||(a, V)||_c &= \{u \mid \emptyset \neq \tau(u, a) \subseteq V\}, \\ ||(a, \emptyset)||_c &= \{u \mid \tau(u, a) = \emptyset\}, \\ ||\alpha \wedge \beta||_c &= ||\alpha||_c \cap ||\beta||_c, \\ ||\neg \alpha||_c &= U - ||\alpha||_c, \\ ||X||_c &= \bigcap \{||\alpha||_c \mid \alpha \in X\}. \end{aligned} \quad (3)$$

Note that in the case of a non-empty finite set of formulas X , $||X||_c = ||\bigwedge X||_c$, where $\bigwedge X$ denotes a conjunction of all formulas of X .

A concept X is referred to as *B-definable* if it is a set-theoretical union of elementary *B-infogranules*. Even if a concept X is not *B-definable* (so is *B-undefinable*), X can be approximated by its *lower* and *upper rough B-approximations* [31, 33], $\text{low}_B^{\cup} X$ and $\text{upp}_B^{\cup} X$, respectively, given by

$$\text{low}_B^{\cup} X \stackrel{\text{def}}{=} \bigcup \{[u]_B \mid [u]_B \subseteq X\} \quad \text{and} \quad \text{upp}_B^{\cup} X \stackrel{\text{def}}{=} \bigcup \{[u]_B \mid [u]_B \cap X \neq \emptyset\}. \quad (4)$$

⁵ Non-emptiness is assumed for simplicity.

$\text{low}_B^U X$ is the largest B -definable concept included in X , whereas $\text{upp}_B^U X$ is the least B -definable concept containing X . If the difference $\text{upp}_B^U X - \text{low}_B^U X$, called the B -boundary region of X , is empty, X will be referred to as B -exact; otherwise it will be B -rough. It turns out that exactness and definability are two faces of the same coin in this case. Furthermore, the lower and upper rough B -approximations of X may equally be defined as

$$\text{low}_B X = \{u \mid [u]_B \subseteq X\} \quad \text{and} \quad \text{upp}_B X = \{u \mid [u]_B \cap X \neq \emptyset\}, \quad (5)$$

respectively, since $\text{low}_B^U X = \text{low}_B X$ and $\text{upp}_B^U X = \text{upp}_B X$ in the case considered. The concepts $\text{low}_B X$ and $\text{upp}_B X$, interpreted as the set of all objects which certainly belong to X and the set of all objects which possibly belong to X , respectively, are instances of functionality-based infogranules. Thus, the lower and upper rough B -approximations of concepts and – more generally – arbitrary B -definable concepts are infogranules in \mathcal{A} .

20.3 Rough Approximation Spaces

Briefly speaking, rough approximation spaces are mathematical structures providing tools for approximation and reasoning about vague concepts. By a *rough approximation space* we mean a triple $M = (U, \varrho, \kappa)$, where U is a non-empty set of objects, ϱ is a reflexive relation on U , and κ is a *rough inclusion function* (RIF in short) on U . Although infinite sets of objects U are allowed, we will mainly deal with finite universes in practice. ϱ , interpreted as a *similarity* relation, assigns to every object u two elementary infogranules: the image of $\{u\}$, $\varrho^\rightarrow\{u\}$, consisting of all objects to which u is similar and the counterimage of $\{u\}$, $\varrho^\leftarrow\{u\}$, being the set of all objects similar to u .⁶ A concept X is referred to as ϱ -definable in the case it is a union of elementary infogranules of the form $\varrho^\leftarrow\{u\}$; otherwise X is ϱ -undefinable. The relation ϱ induces uncertainty mappings $\Gamma_\varrho, \Gamma_\varrho^*$ on U (i.e., such mappings $f : U \mapsto \wp U$ that for every $u \in U, u \in f(u)$ [41]) defined by

$$\Gamma_\varrho u \stackrel{\text{def}}{=} \varrho^\leftarrow\{u\} \quad \text{and} \quad \Gamma_\varrho^* u \stackrel{\text{def}}{=} \varrho^\rightarrow\{u\}. \quad (6)$$

Observe that $\Gamma_\varrho^* = \Gamma_\varrho^{-1}$ and, moreover, $\Gamma_\varrho^* = \Gamma_\varrho$ if ϱ is symmetric. On the other hand, every uncertainty mapping Γ induces a reflexive relation ϱ_Γ such that for any objects u, u' ,

$$(u, u') \in \varrho_\Gamma \stackrel{\text{def}}{\Leftrightarrow} u \in \Gamma u'. \quad (7)$$

Summarizing, a rough approximation space may equally be defined as a structure (U, Γ, κ) as in Skowron–Stepaniuk's approach [41, 42]. Furthermore, if ϱ is derived from attribute values of objects of an infosystem $\mathcal{A} = (U, A, \tau)$, we will say that M is based on \mathcal{A} .

RIFs measure the degree of inclusion of concepts in concepts. The *standard RIF*, going back to Lukasiewicz [43] and denoted by $\kappa^\mathcal{L}$ here, is the best known among RIFs. Let $\#X$ denote the cardinality of a set X . For a non-empty finite U and arbitrary concepts $X, Y \subseteq U$, $\kappa^\mathcal{L}(X, Y)$ is given by

$$\kappa^\mathcal{L}(X, Y) \stackrel{\text{def}}{=} \begin{cases} \frac{\#(X \cap Y)}{\#X} & \text{if } X \neq \emptyset \\ 1 & \text{otherwise.} \end{cases} \quad (8)$$

A formal notion of a rough inclusion was introduced by Polkowski and Skowron and characterized axiomatically within the theory of being part to degree called *rough mereology* [16, 44–46]. In line with rough mereology, by a RIF on U we understand every mapping $\kappa : \wp U \times \wp U \mapsto [0, 1]$ satisfying rif_1 and rif_2 below:

$$\begin{aligned} \text{rif}_1(\kappa) &\stackrel{\text{def}}{\Leftrightarrow} \forall X, Y, \kappa(X, Y) = 1 \Leftrightarrow X \subseteq Y, \\ \text{rif}_2(\kappa) &\stackrel{\text{def}}{\Leftrightarrow} \forall X, Y, Z, Y \subseteq Z \Rightarrow \kappa(X, Y) \leq \kappa(X, Z). \end{aligned} \quad (9)$$

⁶ It is worth noticing that $\varrho^\rightarrow = \varrho^{-1\leftarrow}$, so any infogranule of the form $\varrho^\rightarrow\{u\}$ is, in fact, the counterimage of $\{u\}$ given by the relation converse to ϱ .

One may also consider other postulates, e.g., rif₃–rif₅:

$$\begin{aligned}\text{rif}_3(\kappa) &\stackrel{\text{def}}{\Leftrightarrow} \forall X \neq \emptyset; \kappa(X, \emptyset) = 0, \\ \text{rif}_4(\kappa) &\stackrel{\text{def}}{\Leftrightarrow} \forall X \neq \emptyset; \forall Y, \kappa(X, Y) = 0 \Leftrightarrow X \cap Y = \emptyset, \\ \text{rif}_5(\kappa) &\stackrel{\text{def}}{\Leftrightarrow} \forall X \neq \emptyset; \forall Y, \kappa(X, Y) + \kappa(X, U - Y) = 1.\end{aligned}$$

In the classical Pawlak rough-set model is a rough approximation space defined as a pair (U, ϱ) , where U is a non-empty finite set and ϱ is an equivalence relation of indiscernability of objects [31, 33]. The lower and upper rough approximations of concepts are defined in line with (4) or, equivalently, as in (5). Therefore, RIFs may be omitted in this case.

Given an approximation space $M = (U, \varrho, \kappa)$, every concept X is, in general, assigned a pair of infogranules $(\underline{X}, \overline{X})$, where \underline{X} , a lower approximation of X , is an infogranule approximating X from inside, whereas \overline{X} , an upper approximation of X , is an infogranule approximating X from outside. The lower and upper approximations of concepts can be defined in many ways. Keeping with Pawlak's idea, the following two pairs of approximations of X are obtained, where low and upp stand for 'lower' and 'upper,' respectively:

$$\begin{aligned}\text{low } X &\stackrel{\text{def}}{=} \{u \mid \varrho^{\leftarrow}\{u\} \subseteq X\}, \\ \text{upp } X &\stackrel{\text{def}}{=} \{u \mid \varrho^{\leftarrow}\{u\} \cap X \neq \emptyset\}, \\ \text{low}^{\cup} X &\stackrel{\text{def}}{=} \bigcup \{\varrho^{\leftarrow}\{u\} \mid \varrho^{\leftarrow}\{u\} \subseteq X\}, \\ \text{upp}^{\cup} X &\stackrel{\text{def}}{=} \bigcup \{\varrho^{\leftarrow}\{u\} \mid \varrho^{\leftarrow}\{u\} \cap X \neq \emptyset\}.\end{aligned}\tag{10}$$

The lower (respectively, upper) approximation $\text{low } X$ (respectively, $\text{upp } X$) consists of all objects u that their elementary infogranules $\varrho^{\leftarrow}\{u\}$ are included in (overlap with) X , and $\text{low}^{\cup} X$ ($\text{upp}^{\cup} X$) is the union of such infogranules.

Following Skowron and Stepaniuk [41, 42], the lower and upper approximations of X may be defined by

$$\begin{aligned}\text{low}^S X &\stackrel{\text{def}}{=} \{u \mid \kappa(\varrho^{\leftarrow}\{u\}, X) = 1\}, \\ \text{upp}^S X &\stackrel{\text{def}}{=} \{u \mid \kappa(\varrho^{\leftarrow}\{u\}, X) > 0\}.\end{aligned}\tag{11}$$

We can also obtain their ϱ -definable versions, $\text{low}^{\cup} X$ and $\text{upp}^{\cup} X$, by putting

$$\begin{aligned}\text{low}^{\cup} X &\stackrel{\text{def}}{=} \bigcup \{\varrho^{\leftarrow}\{u\} \mid \kappa(\varrho^{\leftarrow}\{u\}, X) = 1\}, \\ \text{upp}^{\cup} X &\stackrel{\text{def}}{=} \bigcup \{\varrho^{\leftarrow}\{u\} \mid \kappa(\varrho^{\leftarrow}\{u\}, X) > 0\}.\end{aligned}\tag{12}$$

Thus, the lower (respectively, upper) approximation $\text{low}^S X$ (respectively, $\text{upp}^S X$) is defined as the set of all objects u that infogranules $\varrho^{\leftarrow}\{u\}$ are included in X to the highest (to some positive) degree, and $\text{low}^{\cup} X$ ($\text{upp}^{\cup} X$) is the union of such infogranules.

Lower (respectively, upper) rough approximations are also known as *positive (non-negative) regions*. When the difference between a non-negative region and a positive region of X (i.e., a *boundary* region of X) is empty, the concept X will be called *exact*; otherwise X will be *rough*.

In Ziarko's *variable-precision rough-set model* – an extension of the Pawlak rough-set model – are boundary regions of concepts determined to a varying degree of precision [47, 48]. Slightly adapting the original definitions of variable-precision negative and positive regions of concepts, we can define the *s-negative* and *t-positive* regions of X ($0 \leq s < t \leq 1$) as the sets

$$\text{neg}_s X \stackrel{\text{def}}{=} \{u \mid \kappa(\varrho^{\leftarrow}\{u\}, X) \leq s\} \quad \text{and} \quad \text{pos}_t X \stackrel{\text{def}}{=} \{u \mid \kappa(\varrho^{\leftarrow}\{u\}, X) \geq t\},\tag{13}$$

respectively. Their ϱ -definable counterparts are given by

$$\begin{aligned}\text{neg}_s^{\cup} X &\stackrel{\text{def}}{=} \bigcup\{\varrho^{\leftarrow}\{u\} \mid \kappa(\varrho^{\leftarrow}\{u\}, X) \leq s\}, \\ \text{pos}_t^{\cup} X &\stackrel{\text{def}}{=} \bigcup\{\varrho^{\leftarrow}\{u\} \mid \kappa(\varrho^{\leftarrow}\{u\}, X) \geq t\}.\end{aligned}\quad (14)$$

The s -negative (respectively, t -positive) region $\text{neg}_s X$ (respectively, $\text{pos}_t X$) is the set of all objects u that infogranules $\varrho^{\leftarrow}\{u\}$ are included to the degree at most s (at least t) in X , and $\text{neg}_s^{\cup} X$ ($\text{pos}_t^{\cup} X$) is defined as the union of such infogranules. The difference between the t -positive region and the s -negative region of X is called the (s, t) -boundary region of X .

Starting with ϱ^{\rightarrow} instead of ϱ^{\leftarrow} , we arrive at analogous notions of lower and upper approximations as well as variable-precision negative and positive regions, distinguished from the above ones by means of $*$. For instance,

$$\text{upp}^* X \stackrel{\text{def}}{=} \{u \mid \varrho^{\rightarrow}\{u\} \cap X \neq \emptyset\}. \quad (15)$$

When discussing properties of various forms of concept approximation, it can be useful to deal with approximation mappings $f : \wp U \mapsto \wp U$, where $f \in \{\text{low}, \text{upp}, \text{low}^S, \text{upp}^S, \text{neg}_s, \text{pos}_t\}$, instead of approximations of concepts. For any such mapping f , let

$$f^{\cup} X \stackrel{\text{def}}{=} \bigcup\{\varrho^{\leftarrow}\{u\} \mid u \in fX\}. \quad (16)$$

Let \circ denote the composition of mappings.⁷ It turns out that for an arbitrary concept X ,

- (a) $f^{\cup} = \text{upp}^* \circ f$,
 - (b) $\text{pos}_1 = \text{low}^S = \text{low}$,
 - (c) $\text{upp}^S X = U - \text{neg}_0 X$,
 - (d) $\text{upp}^S = \text{upp}$ if $\text{rif}_4(\kappa)$,
 - (e) $f^* = f$ if ϱ is symmetric,
 - (f) $f^{\cup} = f$ if ϱ is an equivalence relation.
- (17)

20.4 Formulas and Rules as Labels of Infogranules

As pointed out by Zadeh [1, 49], words of a natural language⁸ serve as linguistic labels for infogranules understood as fuzzy sets of objects of some sort drawn together on the basis of indiscernability, similarity, or functionality. In computing with words (CW in short), an approach to computing and reasoning proposed by Zadeh [1, 49, 50] (see also [51]), we manipulate with labels of infogranules instead of numbers. It is worth mentioning that

[t]here are two major imperatives for computing with words. First, computing with words is a necessity when the available information is too imprecise to justify the use of numbers. And second, when there is a tolerance for imprecision which can be exploited to achieve tractability, robustness, low solution costs and better rapport with reality. Exploitation of the tolerance for imprecision is an issue of central importance in CW.

(Lotfi A. Zadeh [49])

There are two different views on using words in computation and reasoning [52]. According to the first reasoning schema, a problem to be solved is specified in a user-friendly language. Then, words occurring in this specification are appropriately interpreted as certain infogranules. Next, a computation with these infogranules is performed, and the infogranules obtained as a result are named using words of the language. Finally, the user is provided with the results in the form of linguistic expressions. Granular

⁷ That is, given mappings $g, h : \wp U \mapsto \wp U$, $(h \circ g)X = h(gX)$.

⁸ For the purpose of computation, canonical forms of sentences of a natural language are of use.

computing obviously plays an important role in this paradigm. The second view on CW assumes that the input information for solving a problem is given in the form of a collection of infogranules. In the next step, these infogranules are labeled by words of a language. Then, a computation with words is performed. Finally, the words obtained as a result of the computation are interpreted as certain infogranules which constitute the output information.

In this section we generalize Zadeh's idea on usage of words as labels of infogranules by treating also formulas and rules of a knowledge representation language as such labels. Given a language L interpreted on an approximation space $M = (U, \varrho, \kappa)$, its set of formulas FOR, and a non-empty relation of satisfiability of formulas or sets of formulas of L by objects of U , \models_S . For any object u and any formula or set of formulas Φ , $u \models_S \Phi$ reads as ' Φ is S -satisfied by u ' or, equivalently, ' u S -satisfies Φ '. The counterimage of $\{\Phi\}$ given by \models_S is called the S -extension of Φ and denoted by $||\Phi||_S$. It is a functionality-based infogranule, labeled just by Φ , which consists of all objects S -satisfying Φ . Here denotes $||\cdot||_S$ the mapping assigning to every such Φ its extension $||\Phi||_S$. More generally, Φ will be called S -satisfiable if it is S -satisfied by at least one object, i.e., $||\Phi||_S \neq \emptyset$; otherwise Φ will be S -unsatisfiable.

By a rule r over L we mean a pair (P_r, C_r) of finite sets of formulas called the sets of *premises* and *conclusions* of r , respectively. It is assumed that every rule has at least one conclusion.⁹ A rule r , derived from a decision system, will be called a *decision rule* if P_r, C_r are sets of descriptors beginning with different condition-attribute names and different decision-attributes names, respectively. Decision rules are examples of *association rules* where both premises and conclusions are descriptors, each of which begins with a different attribute name (for methods of induction of association rules and, in particular, decision rules see, e.g., [53–59]).

Given a mapping $||\cdot||_S$ corresponding to a certain relation of satisfiability of sets of formulas \models_S , a rule r may be viewed as a label for the infogranule $(||P_r||_S, ||C_r||_S)$ composed of the S -extensions of the sets of premises and conclusions of r .¹⁰ Furthermore, we can say that r is S -applicable to u , $r \in \text{apl}_S(u)$, if and only if P_r is S -satisfied by u (i.e., $u \models_S P_r$ or, equivalently, $u \in ||P_r||_S$). More generally, r is called S -applicable in the case it is S -applicable to some object (i.e., $||P_r||_S \neq \emptyset$).

Thus, formulas and rules play double roles, namely, as constructive elements and labels of infogranules. Examples of functionality-based infogranules which consist of infogranule labels are the image of $\{u\}$ given by \models_S , consisting of formulas or sets of formulas S -satisfied by u and denoted by $|u|_S$ here, the set $\text{apl}_S(u)$ of all rules S -applicable to u , and a *rough classifier* composed of decision rules together with metarules for resolving conflicts among rules.¹¹

20.5 Rough Satisfiability of Formulas and Their Sets

When reasoning under incomplete, vague, or uncertain information, an agent relying on crisp satisfiability of formulas and sets of formulas will sooner or later encounter difficulties with effectiveness, efficiency, and consistency. Indeed, there can be no object in a database having precisely all properties specified by a formula or a set of formulas Φ . At the same time, there can be a number of objects which almost satisfy Φ , and the agent would take the advantage of this fact if he/she were more flexible. Similarly, the classifiers available to an agent can contain no single rule applicable to a given object in the strict, precise sense. On the other hand, dropping some premises of a rule and treating the remaining ones less strictly can improve the effectiveness and the efficiency of decision making. For instance, suppose that none of candidates can entirely fulfil an employer's requirements. While a 'soft-satisfiability'-oriented employer might accept the

⁹ Usually, it is exactly one conclusion.

¹⁰ A more general case may also take place where the form of satisfiability considered for P_r is different from that one for C_r . For instance, premises of a rule may be tested for satisfiability by one agent, whereas conclusions by another one.

¹¹ Rough classifiers serve the purpose of rough classification of objects (see, e.g., [60, 61]). Rule-based rough classifiers are particular cases of *rule complexes* where the latter notion (i.e., rule complex) is a key mathematical concept of the socially embedded game theory [7–10].

best candidate among those approximately matching the conditions, it is likely that a ‘crisp-satisfiability’-oriented agent will continue his/her search. Additionally, crisp satisfiability is sensitive to noise in the sense that nearly indiscernible objects which, nevertheless, have different descriptions only because of noise can be treated in completely different ways. For such reasons approximate forms of satisfiability are of interest.

Within the rough-set framework we can derive a great deal of approximate satisfiability relations, notions of extension, and forms of rule applicability. It can be a matter of taste, intention, emotion, or optimization which notion of satisfiability an agent is inclined to use. The same agent can prefer a form of satisfiability in one situation and a completely different form in another situation. Apart from the case that an agent has some notions of satisfiability already at his/her disposal, the agent can arrive at new forms of satisfiability by adaptive multilayer learning (see, e.g., [61–68] for learning of complex concepts in the rough-set theory). As we will see, rough notions of satisfiability and extension are often equipped with parameters which may be tuned to obtain a satisfactory quality of results.

In this section we discuss a somewhat simplified view on rough satisfiability of formulas and sets of formulas where notions of rough satisfiability and extension are already given to an agent (see [69–72] for a more detailed elaboration of rough satisfiability and rule applicability). To start with, let us consider an approximation space $M = (U, \varrho, \kappa)$ based on an infosystem $\mathcal{A} = (U, A, \tau)$. We take the descriptor language $L_{\mathcal{A}}$ as the knowledge representation language here.

20.5.1 The Case of Single Formulas

The first form of rough satisfiability of formulas by objects presented here is the *t-satisfiability* where $t \in [0, 1]$. Informally speaking, a formula α is *t-satisfied* by u (or, equivalently, u *t-satisfies* α), $u \models_t \alpha$, if and only if sufficiently many objects similar to u *c-satisfy* α where sufficiency is determined by a threshold value t . More precisely,

$$u \models_t \alpha \stackrel{\text{def}}{\Leftrightarrow} \kappa(\varrho^{-1}\{u\}, ||\alpha||_c) \geq t. \quad (18)$$

Where κ is standard, α is *t-satisfied* by u in the case the ratio of objects similar to u which *c-satisfy* α is not less than t . For example, a formula α saying that a patient has appendicitis is 0.95-satisfied by a patient named u if and only if at least 95% of patients, having similar symptoms as u does, really have appendicitis. Clearly, the more cautious the agent is, the higher is the sufficiency threshold. In the limit case of $t = 1$, all objects similar to u should *c-satisfy* α .¹² The corresponding *t-extension* of α , defined as the set $||\alpha||_t = \{u \in U \mid u \models_t \alpha\}$, is equal to the *t-positive region* of the *c-extension* of α ; i.e., $||\alpha||_t = \text{pos}_t ||\alpha||_c$. In particular, $||\alpha||_1 = \text{low} ||\alpha||_c$.

The above form of satisfiability can be enhanced to guarantee that not only sufficiently many objects similar to u *c-satisfy* a formula α but also u itself *c-satisfies* α . Namely, we say that α is *t⁺-satisfied* by u , $u \models_t^+ \alpha$, if and only if α is both *c-satisfied* and *t-satisfied* by u . Hence, the *t⁺-extension* of α is given by

$$||\alpha||_t^+ = ||\alpha||_c \cap \text{pos}_t ||\alpha||_c. \quad (19)$$

To illustrate this case let objects represent agents clustered into infogranules of witnesses of the same event. According to the above definition, a formula α describing an event will be satisfied by a witness named u if α agrees with u ’s evidence and with evidences of sufficiently many agents having seen the same event. Indeed, an evidence of a witness is more reliable when confirmed by others.

Two more rough forms of satisfiability of a formula can be defined as follows. α will be referred to as *P-satisfied* (respectively, *P*-satisfied*) by u , written $u \models_P \alpha$ (respectively, $u \models_{P^*} \alpha$), if it is *c-satisfied*

¹² A peculiarity of this approach is that u may or may not itself *t-satisfy* α if $t < 1$. Apparently, this can be an advantage when checking whether or not u *c-satisfies* α is problematic as it can be in infosystems with missing values.

by some object similar to u (respectively, to which u is similar).¹³ Formally,

$$\begin{aligned} u \models_P \alpha &\stackrel{\text{def}}{\Leftrightarrow} \exists u' \in U. ((u', u) \in \varrho \ \& \ u' \models_c \alpha), \\ u \models_{P^*} \alpha &\stackrel{\text{def}}{\Leftrightarrow} \exists u' \in U. ((u, u') \in \varrho \ \& \ u' \models_c \alpha). \end{aligned} \quad (20)$$

The corresponding P - and P^* -extensions of α , $||\alpha||_P$ and $||\alpha||_{P^*}$, respectively, are the upper rough approximations in the sense of upp and upp* of the c -extension of α , namely,

$$||\alpha||_P = \text{upp}||\alpha||_c \quad \text{and} \quad ||\alpha||_{P^*} = \text{upp}^*||\alpha||_c. \quad (21)$$

The both forms of satisfiability will coincide if ϱ is symmetric, and analogously for the extensions. Satisfiability of these kinds can be useful where the tolerance for error is high, the time for decision making is strictly limited, or the agent relying on these forms of satisfiability is very brave or, just opposite, very cautious. At first glance, the latter seems to be strange. However, consider an agent who turns down an e-offer just because he/she has learnt about a similar case where a client was cheated.

20.5.2 The Case of Sets of Formulas

Like in the case of single formulas, rough satisfiability of a set of formulas can be defined in a number of alternative ways. As earlier, decision making which form of satisfiability to choose is context dependent and an agent's choice is often motivated by searching for the best solution possible. Clearly, no form of rough satisfiability is universal enough as to fit every situation of satisfiability judgment. One fairly general procedure of judgment of satisfiability of a set of formulas is the following. In order to judge whether or not an object u satisfies a non-empty set of formulas X , one has to decide how many and/or which formulas of X to take into account and what kind of satisfiability of single formulas to choose. First consider the case where all formulas of X are treated equally with respect to their satisfiability. Let \models_S be a form of satisfiability of single formulas, $\kappa' : \wp\text{FOR} \times \wp\text{FOR} \mapsto [0, 1]$ be a RIF measuring the degree of inclusion of sets of formulas in sets of formulas, $t \in [0, 1]$ be a sufficiency threshold value, and $|u|_S$ be the set of formulas S -satisfied by u . Define

$$u \models_{S,t} X \stackrel{\text{def}}{\Leftrightarrow} \kappa'(X, |u|_S) \geq t; \quad (22)$$

i.e., X will be (S, t) -satisfied by u , $u \models_{S,t} X$, if u S -satisfies sufficiently many formulas of X where sufficiency is determined by t . The (S, t) -extension of X is defined as usual; i.e., $||X||_{S,t} \stackrel{\text{def}}{=} \{u \in U \mid u \models_{S,t} X\}$. If X is finite and κ' is standard when restricted to finite first arguments, then $u \models_{S,t} X$ if and only if the ratio of formulas of X which are S -satisfied by u is greater than or equal to t . For example, $(c, 0.9)$ -satisfiability (respectively, $(0.7, 0.9)$ -satisfiability, $(0.7^+, 0.9)$ -satisfiability, $(P, 0.9)$ -satisfiability, and $(P^*, 0.9)$ -satisfiability) of X by u means that no less than 90% of formulas of X are c -satisfied (respectively, 0.7-satisfied, 0.7⁺-satisfied, P-satisfied, and P^* -satisfied) by u . Recall that for $\kappa = \kappa^\mathcal{L}$, a formula α will be 0.7-satisfied by u if α is c -satisfied by at least 70% of objects similar to u ; α will be 0.7⁺-satisfied by u if u itself and at least 70% of objects similar to u c -satisfy α ; it will be P -satisfied by u if at least one object similar to u c -satisfies α ; and, finally, α will be P^* -satisfied by u if it is c -satisfied by at least one object to which u is similar. Let us note that $(c, 1)$ -satisfiability of X is equivalent to its c -satisfiability.

Now we describe a more general case. First, divide X into n non-empty sets X_i , where $i = 0, \dots, n - 1$ and $n > 0$, so $\{X_0, \dots, X_{n-1}\}$ is a partition of X . Both the case with one class only and the case with one-element classes are allowed. For each i , choose a relation of satisfiability of single formulas \models_{S_i} , a RIF $\kappa_i : \wp\text{FOR} \times \wp\text{FOR} \mapsto [0, 1]$ and a sufficiency threshold value $t_i \in [0, 1]$, indicating how many

¹³ ‘P’ refers to Pawlak’s idea on the rough truth of formulas according to which α is *roughly true* in M if $\text{upp}||\alpha||_c = U$. Needless to say, the logical notions of truth and satisfiability are closely related.

elements of X_i are to be S_i -satisfied by u . In the sequel, let $\bar{t} = (t_0, \dots, t_{n-1})$ and $\bar{S} = (S_0, \dots, S_{n-1})$. Then,

$$u \models_{\bar{S}, \bar{t}} X \stackrel{\text{def}}{\Leftrightarrow} \forall i = 0, \dots, n-1. \kappa_i(X_i, |u|_{S_i}) \geq t_i, \quad (23)$$

which means that X is (\bar{S}, \bar{t}) -satisfied by u if and only if for each $i = 0, \dots, n-1$, sufficiently many formulas of X_i are S_i -satisfied by u where sufficiency is determined by t_i . In this case,

$$||X||_{\bar{S}, \bar{t}} = \bigcap_{i=0}^{n-1} ||X_i||_{S_i, t_i}. \quad (24)$$

By a suitable selection and adjustment of forms of satisfiability S_i and threshold parameters t_i , a situation can be modeled where formulas of X , representing conditions to be fulfilled, are important to a various extent. While some can be indispensable and have to be treated very seriously, others may even be omitted. For instance, let $n = 2$, \models_{S_0} be \models_1 , \models_{S_1} be \models_p , and $\bar{t} = (1, 1)$. Then,

$$u \models_{\bar{S}, \bar{t}} X \Leftrightarrow u \in \bigcap_{\alpha \in X_0} \text{low}||\alpha||_c \cap \bigcap_{\alpha \in X_1} \text{upp}||\alpha||_c; \quad (25)$$

i.e., u belongs to the lower rough approximation in the sense of low of the c -extension of each formula in X_0 and to the upper rough approximation in the sense of upp of the c -extension of each formula in X_1 . Thus, all conditions in X are necessary for its fulfillment, but yet satisfiability of some formulas (namely, those constituting X_0) is more important than satisfiability of others (i.e., those in X_1).

Observe that instead of numerical sufficiency threshold values, one may use qualitative thresholds by means of words like ‘many,’ ‘few,’ ‘large number,’ ‘small,’ ‘most of,’ and so on. In line with Zadeh, these words are labels for infogranules of numbers. For instance, ‘almost all’ may mean about 100% of objects, ‘most of’ may be interpreted as more than 50% of objects, and so forth. Thus, instead of saying that ‘ X is (0.6, 0.98)-satisfied by u ,’ one may say that ‘ X is (most-of, almost-all)-satisfied by u ,’ interpreted as almost all formulas of X are satisfied in the crisp sense by most of the objects similar to u .

20.6 In Search of Objects Satisfying a Specification

In this section we discuss searching for objects which satisfy a specification, excluding cases where such objects are to be constructed. Here by a specification we understand a non-empty finite set of formulas of a certain knowledge representation language. These formulas are intended to describe features of an object or, more accurately, a class (infogranule) of objects.

An exemplary specification in a natural language may state that ‘[a] candidate for a position P is high educated in computer science or related disciplines (Ph.D. would be an advantage), has several years of practice at such or similar posts, is fluent in English, French and Spanish, and has good organizational skills.’ Other examples of specifications are a set of mathematical equations and/or inequalities (solutions to these (in)equalities are just the objects satisfying the specification), a mathematical theory (models of this theory are the objects searched for), a set of objectives to be realized (appropriate action or interaction strategies are the objects we are looking for), a negotiation protocol (negotiation processes complying with this protocol are the objects sought for), a set of constraints on a schedule,¹⁴ a specification of an algorithm, a recipe for making a cake, an order for a new computer placed with a high-tech shop, a set of formulas of the descriptor language of an infosystem, and so forth.

Consider a language L and a specification $X \subseteq \text{FOR}$. Let U^∞ denote a non-empty set of all objects conceived (the universe). The aim is to find sufficiently many objects satisfying X where both ‘sufficiency’ and ‘satisfiability’ are understood in a certain, possibly vague sense.

¹⁴ For instance, constraints on a train timetable.

20.6.1 The Single-Agent Case

Suppose that an (individual) agent Ag is charged with the aforementioned task. For simplicity, let ‘sufficiently many’ be synonymous to ‘as many as possible.’ Assume also that Ag may decide him/herself which kind of satisfiability to choose. Suppose that it is S -satisfiability, so Ag’s task is to find as many objects S -satisfying X as possible. All such objects constitute the S -extension of X ; i.e., $\|X\|_S = \{u \in U^\infty \mid u \models_S X\}$. However, only a finite subset of the universe U^∞ is usually available in practically oriented domains. Thus, assume for the time being that Ag only has an access to U being a non-empty finite subset of U^∞ .

Given an object u , judgments of $u \models_S X$ and $u \in \|X\|_S$ are equivalent. While Ag judges whether or not X is S -satisfied by u in the former case, in the latter one he/she judges whether or not u is subsumed under the concept $\|X\|_S$ (or u is one of $\|X\|_S$ ’s). That is, searching for objects S -satisfying X is equivalent to seeking objects classified as members of the class $\|X\|_S$.

Clearly, searching for objects S -satisfying X is an optimization problem the solution of which is a subset of $\|X\|_S$. A searching mechanism and a rough or another ‘soft’ classifier for $\|X\|_S$ is needed to solve the problem (for rough classifiers see, e.g., [60, 61, 67]). If Ag is given a partial classifier for $\|X\|_S$, i.e., a set of rules or a function which for any object considered, returns ‘yes,’ ‘no,’ or ‘undecided,’ then all objects of U for which the decision is ‘yes’ will plausibly constitute a subset of $\|X\|_S \cap U$ being a solution to the search problem. In the most optimistic case the whole set $\|X\|_S \cap U$ can be obtained, but yet the case where no object is classified as S -satisfying X is possible as well.

20.6.2 When Satisfiability of a Specification Is Problematic

Ag, who – for the sake of simplicity – is assumed to have a partial classifier R for $\|X\|_S$, is obviously interested in finding at least one object S -satisfying X . However, it can happen that no object of U will be classified by R to $\|X\|_S$. The difficulties with satisfiability of X can be caused by several things.

First, X can be logically inconsistent in the sense it can contain mutually contradictory formulas. Then, it can be difficult to find a single object satisfying X even if S -satisfiability is non-crisp. Secondly, X can be S -unsatisfied by any object available to Ag in spite of the fact that such objects do exist. In this case Ag would possibly find appropriate objects if U consisted of sufficiently many objects. For example, an employer E cannot decide whom to offer a job because no actual candidate satisfies the requirements. Maybe, if there were more candidates, E would make a decision.

It can also happen that the requirements described by X are consistent, but unrealistic, or that S -satisfiability is too restrictive.¹⁵ As a result, relatively few objects of U^∞ can S -satisfy X at all, so the chance that none of them belongs to U is high. A good (and sad) illustrative example is the case of the prophylactic program toward an early diagnosis of TB in a certain country C . The specification of a participant of the program contained many, sometimes almost excluding, conditions. Consequently, very few candidates were accepted.

Another source of failure can be limited search-and-test abilities of Ag (e.g., physical constraints on time, space, and resources). By way of example, suppose that U is so huge that practically infinite, similarity is not transitive, and X consists of a non-tautological formula α . One may stipulate α to be satisfied by an object u if it is c -satisfied by all objects similar to u and all objects similar to those similar to u , and so forth. Under strictly limited time, an effective checking-up of satisfiability of α is hardly possible.

Last but not the least, the classifier R can be unsuitable to the particular task. It can be too weak (too many ‘undecided’ returned) or erroneous (‘no’ answers instead of ‘yes’ answers). The sources of undecidability of R can be the language L itself,¹⁶ the fact that the classifier was trained and tested on

¹⁵ The situation resembles approximation of concepts by their lower rough approximations which can be empty even for concepts consisting of considerably many objects.

¹⁶ The satisfiability problem is known to be undecidable for too expressive languages like, e.g., the full first-order language.

data very different from the given ones, and incompleteness and/or noisiness of the data available to Ag, to name a few.

20.6.3 How To Overcome Possible Difficulties

In the case of failure, Ag may stop searching and may report that no object has been found, or he/she may continue, adjusting the procedure appropriately.

When the problem seems to lie in X , Ag may request that X be more realistic or free of contradictions. Often, however, Ag is not in power to shape the specification. Nonetheless, the agent may try to enlarge the search space U by new objects and to modify or to change the form of satisfiability and/or the classifier. It is like in the real-life example where tenders for a piece of work are reinvited (and sometimes offered more attractive conditions) because no candidate has fulfilled given requirements so far.

When the failure is caused by the constraints put on Ag and the procedures used, the agent may try to renegotiate the contract or to choose more efficient procedures, e.g., a less demanding, yet still acceptable, form of satisfiability or a better classifier.

As regards changes in S -satisfiability, parameters in \models_S (e.g., threshold values) are tuned so that some objects will eventually satisfy X . Needless to say that there exist such rough forms of satisfiability that even logically inconsistent specifications can be satisfied. By way of illustration, suppose that X consists of a descriptor (a, V) and its negation. Clearly, X is c -unsatisfiable. However, it is, e.g., $(c, 0.5)$ -satisfiable, which means that at least 50% of formulas of X are c -satisfiable. In this case, the use of $(c, 0.5)$ -satisfiability is equivalent with partition of X into two one-formula sets and application of c -satisfiability to them.

If Ag's classifier R is undecisive or returns 'no' for an object u only because of the lack of information, S -satisfiability may be replaced by a more flexible form which, e.g., will use information about objects (dis)similar to u . As a matter of fact, many rough notions of satisfiability enable us to reason by analogy. For instance, Ag can draw a conclusion that u satisfies X because sufficiently many formulas of X are satisfied in the required sense by sufficiently many objects similar to u even if u does not c -satisfy any formula of X . Clearly, analogy-based reasoning is risky but sometimes it is the only effective and/or efficient method of reasoning in a given situation (for analogy-based reasoning in classifier construction see, e.g., [73]). For example, in the course of diagnosing a patient, it can be impossible or risky to perform some tests in a short time. Then, on the basis of similar cases and taking into account the available information, a physician may, nevertheless, decide to operate on a patient or to start a treatment if the level of risk is not too high and the expected results are promising.

Decision makers often take into account not only objects similar to a given object but also objects dissimilar to it.¹⁷ In other words, an agent makes judgments of $u \models_S X$ by analyzing arguments for and against this statement. In [19, 63] are arguments for and against the membership of an object in a concept used to approximate complex concepts. Such an *argument* is a triple (s, α, t) , where α is a *pattern* describing a class of objects (e.g., a conjunction of descriptors of a descriptor language for an infosystem) and $s, t \in [0, 1]$ are threshold values for rough inclusion of concepts in concepts.

Changes in the form of satisfiability often influence what a classifier is used. Conversely, it can be the classifier to blame for the failure. Then, keeping the satisfiability form unchanged, the classifier is improved or rebuilt to be more correct, robust, and decisive.

20.6.4 Searching for Objects as a Multilayer Process

In many cases of satisfiability relations, the cases from Section 20.5 included, satisfiability of a set of formulas resolves itself – one way or another – into satisfiability of single formulas. Let \models_S be of that

¹⁷ However, most classifiers in data mining are based on either similarity or dissimilarity [74]. In [75] we extend the notion of an approximation space by a relation of dissimilarity of objects which need not be complementary to the relation of similarity. In such a space are concepts approximated by means of similarity- and dissimilarity-based infogranules.

kind. Then, a classifier for $\|X\|_S$, R , is built on some classifiers R_0, \dots, R_n for $\|\alpha_0\|_{S_0}, \dots, \|\alpha_n\|_{S_n}$, respectively, where $\alpha_0, \dots, \alpha_n$ are some formulas of X , \models_{S_i} is a satisfiability relation selected for α_i , and $i = 0, \dots, n$. In the case R is a rule-based classifier, it contains rules for composition of answers from R_0, \dots, R_n . For example, such a rule could say that ‘if most of the answers returned by R_0, \dots, R_m are positive and at least one answer returned by R_{m+1}, \dots, R_n is positive (for some $m < n$), then the decision will be yes.’

Consider a formula $\alpha \in X$, indispensable for S -satisfiability of X , to be tested for S' -satisfiability. In the case where S' -satisfiability is the crisp one and α is a descriptor (a, V) of the descriptor language L_A of an infosystem $A = (U, A, \tau)$, the classifier for $\|\alpha\|_{S'}$ is very simple: It will return ‘yes’ (i.e., $u \in \|\alpha\|_{S'}$) if both $\tau(u, a), V$ are non-empty and $\tau(u, a) \subseteq V$, or both $\tau(u, a), V$ are empty; the answer will be ‘no’ otherwise. Where S' -satisfiability is one of the rough forms of satisfiability described in Section 20.5, making a judgment whether or not $u \in \|\alpha\|_{S'}$ resolves itself into a number of judgments whether or not $u' \in \|\alpha\|_c$, performed for some objects $u' \in U$, and a comparison of the number of positive (or negative) answers with a threshold value. Also when α is an arbitrary formula of L_A , testing its S' -satisfiability will be reduced to testing c -satisfiability of some descriptors.

Now suppose that α cannot be expressed as a formula of the descriptor language L_A . This means that Ag cannot say whether or not an object u s' -satisfies α only on the basis of information contained in the infosystem A . In many cases, α can successfully be rewritten to a formula β of a descriptor language. However, the problem will remain open¹⁸ unless Ag is given new, relevant information. For example, α may state that ‘a situation on a road is safe’ or ‘a situation in a forest is an emergency.’ Formally, both the expressions can easily be transformed into a descriptor form, namely, (safe-situation-on-road, {yes}) and (emergency-in-forest, {yes}), respectively.¹⁹ Unfortunately, this does not help much because the above descriptors denote complex concepts which cannot be learned solely on the basis of sensor data.

A rule-based classifier for $\|\alpha\|_S$ should contain at least one decision rule with α (or with a formula β in some sense equivalent to α) as the conclusion.²⁰ If there exist more than one rule with this property, Ag will have to select one of such rules, say r , to apply in this case. Then, an object u will plausibly S' -satisfy α from Ag’s perspective if the set of premises of r , P_r , is S'' -satisfied by u , where $\models_{S''}$ is a relation of satisfiability of sets of formulas by objects.²¹ The degree to which the fact of S' -satisfiability of α is conditioned by S'' -satisfiability of P_r depends on the quality of r , measured by various indices as support, confidence, and coverage, to name a few (see, e.g., [76] for discussion of rough forms of these indices). In this way, the judgment of S' -satisfiability of α by u is reduced to the judgment of S'' -satisfiability of P_r by u . In some cases of approximate satisfiability, *production* rules are used instead of typical decision rules [62, 66–68, 77]. An example of a production rule is the following: ‘If the degree of membership in concept C_1 is medium and the degree of membership in C_2 is high, then the degree of membership in C will be medium.’ Briefly speaking, such rules enable us to draw conclusions on the degree of membership of an object in a concept on the right-hand side of a rule on the basis of the degrees of membership of the object in concepts on the left-hand side of the rule.

Next, as in the case of S -satisfiability of X , a classifier for $\|P_r\|_{S''}$ is needed, based on classifiers for $\|\beta_0\|_{S'_0}, \dots, \|\beta_m\|_{S'_m}$, where β_0, \dots, β_m are some formulas of P_r indispensable for its S'' -satisfiability, $\models_{S'_i}$ is a satisfiability relation selected for β_i , and $i = 0, \dots, m$. By principle, premises of a decision rule refer to concepts simpler than the one described by the conclusion. Therefore, assuming ‘reasonable’ forms of satisfiability, Ag will need classifiers for less and less complex concepts. Eventually, he/she will arrive at the level where the formulas under investigation are formulas of the descriptor language L_A .

Thus, searching for objects S -satisfying X is a multilayer process where, at each level, classifiers for concepts occurring at this level are needed together with mechanisms for tuning parameters and switching from one to another form of satisfiability (or membership) whenever necessary or convenient.

¹⁸ Instead of a classifier for $\|\alpha\|_{S'}$, a classifier for $\|\beta\|_{S'}$ is needed.

¹⁹ (safe-situation-on-road, yes), (emergency-in-forest, yes) are even simpler forms.

²⁰ For simplicity, let satisfiability of $\{\alpha\}$ be reduced to S' -satisfiability of α and, moreover, let decision rules have one conclusion only.

²¹ In particular, it can be equal to \models_S .

Such classifiers may be partly given and partly learned from the sensory data and the domain knowledge available.²² In the case of rule-based classifiers, the rules given to an agent can be rules of logical inference (*modus ponens*, a rule for decomposition of conjunction into conjuncts, and so on), common-sense rules, or rules provided by experts. For instance, safety on road is partly described by rules of the traffic regulations and partly by common-sense rules. Similarly, emergency in forest is to high extent described by operating procedures for forest guards, yet some rules like ‘do not make an open fire’ or ‘do not leave glass in forest’ are (or, rather, should be) common knowledge. Other rules have to be discovered by Ag in the course of a process of (un)supervised learning of concepts from sensory data and/or domain knowledge [61–66, 68, 78, 79, 81–83]. As a result, a network of approximate concepts can be obtained, serving as an approximate ontology of concepts [84].

20.6.5 The Multiagent Case

The problem of searching for objects satisfying a specification X was, so far, considered from the standpoint of a single agent. Now we briefly discuss this issue, assuming that a collective agent (i.e., a group of agents), Ag, is charged with such a task.

To start with, suppose that $\text{Ag} = \{\text{Ag}_0, \dots, \text{Ag}_k\}$ ($k > 0$). Every agent Ag_i has an access to his/her own sources of information and knowledge as well as to the sources shared by the group. Among other things, Ag_i is given a non-empty finite set of objects U_i serving as Ag_i 's search space. Let \models_S be chosen by Ag as the satisfiability relation as earlier, and let Ag be given a classifier for $\|X\|_S$, R , for the sake of simplicity. Assume also that $\alpha_0, \dots, \alpha_n$ are some formulas of X , important for S -satisfiability of X in the sense that R is based on classifiers R_0, \dots, R_n for $\|\alpha_0\|_{S_0}, \dots, \|\alpha_n\|_{S_n}$, respectively, where \models_{S_i} is a satisfiability relation assigned to α_i for $i = 0, \dots, n$.

The agents make a plan who will do what; i.e., they decompose the main task into subtasks to be performed by particular members of the group. For simplicity, assume that $k > n$ and Ag_i is charged with searching for objects S_i -satisfying α_i where $i = 0, \dots, n$. Then, the remaining agents may be asked to help $\text{Ag}_0, \dots, \text{Ag}_n$ in their tasks, or they may only take part in the final decision making whether or not a particular object S -satisfies X . By assumption, the agents are cooperative and positively motivated to realize their parts of the plan and to achieve the final goal.

Now suppose that the agents have already accomplished their tasks, which means that for each α_i , a set of objects S_i -satisfying this formula is returned as the result of Ag_i 's search. In the next step, Ag collectively makes a judgment which of these objects S -satisfy the whole specification X . Such a procedure should take into account the results obtained by particular agents and combine them into a set of objects S -satisfying X . In order to be effective and efficient, Ag needs to have developed a procedure, e.g., in the form of a set of metarules, to resolve conflicts among agents who may have contradictory opinions.

As an illustrative example consider a conference program committee whose task is to evaluate the quality of contributed papers in order to select a collection of accepted articles. Each agent, a committee member, is supposed to judge upon several papers, whereas each paper is reviewed by three independent agents. A single agent's overall recommendation is based on the results of judgments of such issues as quality of content, significance for theory/practice, originality, relevance for the conference, quality of presentation, and so forth. Next, for every paper, the agents' opinions are combined into a final statement of acceptance or rejection. Since the opinions may contradict one another, a procedure for resolving possible conflicts is badly needed. Although the example is described in terms other than ‘specification,’ ‘satisfiability,’ ‘formula,’ ‘object,’ and ‘classifier,’ it can be given a more formal shape relatively easy.

20.7 Construction of Infogranules

Construction of an infogranule is usually not a matter of an ordinary application of mathematical operations to some objects. Indeed, the result of construction should possess intended properties, at least to a

²² A classifier for (safe-situation-on-road, {yes}) was subject to an intensive research [61, 62, 68, 78–80]. The case of ‘emergency-in-forest’ might be treated analogously.

satisfactory extent. When constructing an infogranule, two major stages can be identified. First, appropriate objects serving as constructive elements should be found or given. Next, these elements are composed into an infogranule according to a given specification. The first step has already been discussed, so we focus on the latter now.

In the course of a granule construction, an individual or collective agent charged with this task may need to perform operations on infogranules and/or their labels. Such operations are constrained by conditions specifying numerical or symbolic values of variables. For instance, a constraint can say that the result of aggregation of infogranules has to be an infogranule near to the degree at least t to a given infogranule X where $t \in [0, 1]$ or that the resulting infogranule has to be similar to X as much as possible.

Operations on infogranules (including approximation, decomposition, and aggregation), nearness, and similarity of infogranules are major issues in granular computing – a soft approach to computation developed also within the rough-set theory [11, 13, 15–18, 20–22, 27, 77, 85, 86]. On the other hand, operating on infogranule labels may be placed among research problems within computing with words. As mentioned in Section 20.4, both granular computing and computing with words are closely related. Computing with words (or labels) may be viewed as a step in computation with infogranules and vice versa.

20.7.1 Distributive vs. Collective Infogranules

In the case of infogranules being distributive classes,²³ the second step in infogranule construction plays a minor role. Indeed, when constructive elements are already found or given, making a granule will consist in a straightforward selection of a sufficient number of them. By way of illustration, given an infosystem $\mathcal{A} = (U, A, \tau)$, the construction of the infogranule of objects similar to a given object $u \in U$ and 0.6-satisfying a formula α of the descriptor language $L_{\mathcal{A}}$ resolves itself into a selection from all objects similar to u , such objects u' that 60% of objects similar to u' c -satisfy α .

Unlike in the above case, the second stage can be difficult when infogranules are collective classes.²⁴ An example is creation of a winning strategy by an agent for a successful interaction with other agents in a group. Although the elementary actions (moves) available to the agent can be roughly known, guessing how to compose them into a winning strategy is not an easy matter.

20.7.2 Set-Theoretical Operations on Infogranules

Consider infogranules $G, G_1, G_2 \subseteq U$ and a non-empty family of infogranules $\mathcal{G} \subseteq \wp U$.

Set-theoretical operations can yield (but not necessarily do) infogranules when applied to infogranules. It can be the case that the union $\bigcup \mathcal{G}$, the intersection $\bigcap \mathcal{G}$, the difference $G_1 - G_2$, the complement $U - G$, the Cartesian product $G_1 \times G_2$, the product $\Pi \mathcal{G}$, or the power set $\wp G$ are infogranules. For instance, according to the specification of an infogranule G , let an object u be an element of G if and only if it is an element of G_1 or G_2 . This means, in fact, that $G = G_1 \cup G_2$. By way of illustration, given an approximation space $M = (U, \varrho, \kappa)$, suppose that $Y \subseteq U$ is an infogranule in M if and only if Y is ϱ -definable. Then, any union of infogranules in M is an infogranule in M as opposite to the case of intersection where the intersection of two different ϱ -elementary infogranules $\varrho^{\leftarrow}\{u\}$ and $\varrho^{\leftarrow}\{u'\}$ may or may not be an infogranule in M .

Furthermore, it is possible but not necessary that subsets and supersets of infogranules are infogranules. In particular, binary relations on G_1 and G_2 may be infogranules as subsets of $G_1 \times G_2$. For example, let U_1 and U_2 be non-empty sets, $U = U_1 \times U_2$, and an infogranule be any partial mapping²⁵ on U_1 . Hence, subsets of infogranules are also infogranules. On the other hand, only such supersets of infogranules are infogranules which are partial mappings on U_1 .

²³ Similarity-based infogranules are often of this kind, e.g., a concept in an approximation space or a set of objects similar to a given object. An example of a distributive functionality-based infogranule is a set of decision rules.

²⁴ Examples of such infogranules are complex objects like texts, plans for (inter)actions, strategies, schedules, codes, regions on a plane, computer programs, and so on.

²⁵ For the sake of simplicity, mappings are viewed as partial mappings too.

Since additional constraints are often imposed on the result of application of the set-theoretical operations, these operations alone are usually insufficient to produce infogranules satisfying a given specification. In [87] introduced Skowron and Stepaniuk the notion of a *constrained sum* of infosystems (and the corresponding operation on approximation spaces), being an example of a constrained operation on infogranules.

20.7.3 Decomposition and Aggregation

In general, two fundamental types of operations on infogranules can be distinguished: decomposition and aggregation. Operations of the former kind decompose infogranules into families of infogranules [13, 77, 86, 88]. Where ϱ is an equivalence relation on U , the partition of a ϱ -definable concept Y into a family of ϱ -elementary infogranules, being equivalence classes of ϱ , is a simple example of decomposition. A more sophisticated case is a decomposition of an infosystem into a family of infosystems satisfying some postulates. Other examples are the operation of difference and that of taking a subcomplex (subset) of a complex (set) of objects [10], provided that some other requirements put on the infogranule construction are fulfilled.

As an operation, aggregation is converse to decomposition. Namely, a family of infogranules \mathcal{G} is transformed by such an operation into an infogranule G . Aggregation of infogranules can formally be analyzed within (rough) mereology by means of a class operator. An easy example of aggregation of infogranules is the operation of union applied to families of ϱ -definable concepts. Another interesting example is the operation of a constrained sum [87]. Assuming that some additional conditions are satisfied, the operations of union, intersection, Cartesian product, product, and composition of relations may be viewed as instances of aggregation too.

Not all elements of infogranules constituting the family \mathcal{G} have to be used to build G . For instance, consider logical theories (or, to put it another way, sets of formulas), consistent in some sense, as infogranules. Assume that the result of aggregation of a family of such theories \mathcal{T} should be a consistent, \subseteq -maximal theory T included in $\bigcup \mathcal{T}$. Clearly, it can happen that some formulas of $\bigcup \mathcal{T}$ have to be dropped to retain consistency. In general, an aggregation of \mathcal{G} into G proceeds in two steps: First, some objects of $\bigcup \mathcal{G}$ are selected as the constructive elements of G and, then, these elements are composed into G in accordance with given requirements. That is, aggregation of \mathcal{G} into G is an instance of construction of G where the search space for constructive elements is restricted to $\bigcup \mathcal{G}$.

As a set of formulas, an infogranule specification is itself an infogranule of labels. Therefore, decomposition and aggregation of specifications are particular cases of decomposition and aggregation of infogranules, respectively. In [13, 88] is a method of construction of an infogranule G satisfying a specification X proposed where X is first decomposed into a family of specifications of parts of G and, next, these parts – being objects of lower level with respect to G – are constructed. Finally, G is aggregated from the constructed parts in a way securing satisfiability of X by G .

20.7.4 Remarks on Operations on Labels

In the case of infogranules being concepts of U , labeled by formulas of a knowledge representation language L , we may perform logical operations on the labels instead of operating directly on infogranules. For instance, assuming crisp satisfiability of formulas, conjunction (respectively, disjunction) of formulas corresponds to intersection (union) of their c -extensions, whereas negation of a formula corresponds to taking the complement of the c -extension of this formula. Intensional operators²⁶ applied to a formula will provide us with various approximate regions of the c -extension of the formula. That is, such intensional operators are logical counterparts of certain approximation mappings.

²⁶ For example, operators of knowledge and belief.

Where infogranules are pairs of concepts of U , labeled by rules over L , we may also try to operate on the labels. Given $Y, Y' \subseteq U$, r may be viewed as a label for (Y, Y') if Y and Y' are extensions of the sets of premises and conclusions of r , P_r and C_r , respectively. Examples of operations on rules are operations of adding and deleting premises, add_Z , del_Z , respectively, parameterized by finite sets of formulas Z and defined as follows, for any rule r :

$$\begin{aligned}\text{add}_Z(r) &\stackrel{\text{def}}{=} r' \quad \text{where } P_{r'} = P_r \cup Z \text{ and } C_{r'} = C_r, \\ \text{del}_Z(r) &\stackrel{\text{def}}{=} r'' \quad \text{where } P_{r''} = P_r - Z \text{ and } C_{r''} = C_r.\end{aligned}\tag{26}$$

In the crisp case, the augmentation of the set of premises of r by the formulas of Z makes r more specific as opposite to the removal of these formulas from P_r . Indeed,

$$||P_{r'}||_c \subseteq ||P_r||_c \subseteq ||P_{r''}||_c.\tag{27}$$

The method of dropping premises proves useful in search of robust decision rules, correctly classifying new objects.

20.7.5 A Multilayer Process of Infogranule Construction

In general, construction of an infogranule G satisfying a specification X can be a very complex, multi-layered process. Apart from requirements regarding objects which constitute G , X can contain postulates concerning the architecture of G (in particular, the relationships among components of G) and/or the right way of achieving the goal, i.e., various norms saying how to construct G properly. Therefore, apart from searching for objects to be composed into G , an agent Ag may have to search for relationships among these objects and, moreover, to judge whether or not the way in which G is built up is correct.

Since some or even all objects constituting G can themselves be infogranules, Ag may need to construct these components of G , say G_0, \dots, G_n , first. Clearly, the same can apply to G_0, \dots, G_n . In this way, a hierarchical structure of infogranules to be constructed is obtained where the initial infogranule G is placed at the highest level, the components of G are put one level beneath, and so forth. The very bottom layer consists of ‘atomic’ objects such that no further construction is needed in order to find them. It is worth recalling that apart from obtaining all these components of G , one has to link them accordingly. Indeed, ‘[j]ust as a 1.3-kg pile of neurons is not a brain unless the neurons are wired together properly’ (Jeffrey O. Kephart [89]), a loose collection of objects each of which satisfies a given specification need not be the infogranule one is looking for. Linking the infogranules constituting G with one another according to the specification X is, in fact, a construction of appropriate relations on the set of all such infogranules. These relations are specified partly and approximately only, so their construction is not an easy matter in general.

Suppose that X specifies a certain goal to be realized, e.g., to sail by boat from A to B and, moreover, to achieve this safely, as fast as possible, and keeping the total costs at a low level. The infogranule to be constructed is a plan for a sailing-boat trip from A to B, taking the specified requirements into account. The proper part of the plan can be represented by a path in a graph where nodes are labeled by states (or situations) and edges by actions to be performed to move from a state to a state. A realization of such a plan can be visualized as moving from a node labeled by the initial state to a node labeled by the terminal state along a selected path of the graph representing the plan. Before any arrangements are made, an agent planning the trip will have to understand several complex concepts like ‘safe sailing,’ ‘as fast as possible,’ and ‘low costs.’ To put it another way, classifiers are needed to judge whether or not a sailing-boat trip is safe, fast, and economic. Then it can turn out that a number of other complex concepts need to be learned. For instance, to judge upon safety such questions should be addressed as the technical condition of the boat, training, experience, and health conditions of the captain and the crew, and weather conditions, to name a few. Clearly, each of these concepts is also complex and subject to a further analysis. For example, a classifier is needed to decide whether or not the sailing boat under consideration is in a good order and can be rented for such a trip. Due to high complexity of concepts used as well as incompleteness and uncertainty of information available, reasoning and judging upon

such questions as what a boat to rent, whom to invite to the crew, when to start, what a route to choose, what to do in the case of a failure in the plan realization, what to do in the case of emergency, how much money is needed, whom to ask to sponsor the trip, and so on are approximate only. Every state and every action of the plan should be carefully thought over, however, making modifications and changes possible. Depending on the level of granularity, the plan should be less or more detailed. Nevertheless, only some aspects can be decided precisely. For example, although general tendencies in weather conditions can be known for a region of interest, the weather is usually highly unpredictable over a longer period of time, so one has to be prepared for changing weather conditions. Construction of such a compound action plan satisfying a specification and enabling an adaptation to varying conditions is an example of a challenging task for granular computing.

20.8 Summary

In this chapter we addressed the problem of construction of information granules satisfying a given specification. Such a specification is understood as a non-empty finite set of formulas of a knowledge representation language, describing the desired properties of the infogranule to be constructed. Although our perspective was the rough-set one, many observations are general enough to apply to other soft-computing paradigms. Construction of an infogranule was presented as a multilayer process with two major steps at each layer: (i) searching for objects to be constructive elements of infogranules at a particular level and (ii) composition of these objects into infogranules according to the specification.

The central notion in an infogranule construction is that of satisfiability of formulas and sets of formulas by objects. Since judgment of satisfiability can be transformed into judgment of membership of objects in concepts, both the problems (i.e., formula satisfiability and concept membership) were discussed interchangeably.

In general, construction of an infogranule satisfying a given specification is a complex task. It involves searching for and/or construction of objects which are supposed to be members of concepts described by the specification. Unfortunately, these concepts are usually known to the agent constructing the infogranule only partially. Therefore, many particular tasks of an infogranule construction remain a challenge despite the fact that rough (and other) classifiers for concept membership as well as other granular computing techniques can be employed. Examples of interesting, yet difficult, cases are construction of computing systems (particularly the autonomic ones [89]), compound action plans, models of agents' coalitions, models of economic clusters, and networks of classifiers for complex concepts.

Acknowledgment

The author expresses her gratitude to Professor Andrzej Skowron for many insightful remarks and helpful discussions which influenced the shape of this chapter. Thanks also go to the anonymous referees for useful comments which helped improve the paper. All errors left are the author's sole responsibility. The research has been supported by the grant N N516 368334 from Ministry of Science and by the grant Innovative Economy Operational Programme 2007–2013 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland.

References

- [1] L.A. Zadeh. Outline of a new approach to the analysis of complex system and decision processes. *IEEE Trans. Syst. Man Cybern.* 3 (1973) 28–44.
- [2] L.A. Zadeh. Fuzzy sets and information granularity. In: M. Gupta, R. Ragade, and R. Yager (eds), *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, 1979, pp. 3–18.
- [3] L.A. Zadeh. Toward a theory of fuzzy information granulation and its certainty in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.

- [4] R. Breault. The evolution of structured clusters, 2000. <http://www.ptbmagazine.com/July00/ptb700.clusters2.html>, accessed January 2008.
- [5] M.E. Porter. Clusters and the new economics of competition. *Harv. Bus. Rev.* 76(6) (1998) 77–90.
- [6] M.E. Porter. *On Competition*, Harvard Business School Press, Boston, MA, 1998.
- [7] T.R. Burns and A. Gomolińska. The theory of socially embedded games: The mathematics of social relationships, rule complexes, and action modalities. *Qual. Quant.* 34(4) (2000) 379–406.
- [8] T.R. Burns, A. Gomolińska, and L.D. Meeker. The theory of socially embedded games: Applications and extensions to open and closed games. *Qual. Quant.* 35(1) (2001) 1–32.
- [9] T.R. Burns and E. Roszkowska. Generalized game theory: Assumptions, principles, and elaborations in social theory. *Stud. Log. Grammar and Rhetoric* 8(21) (2005) 7–40.
- [10] A. Gomolińska. Fundamental mathematical notions of the theory of socially embedded games: A granular computing perspective. In: S.K. Pal, L. Polkowski, and A. Skowron (eds), *Rough-Neural Computing: Techniques for Computing with Words*. Springer-Verlag, Berlin, Heidelberg, 2004, pp. 411–434.
- [11] M. Inuiguchi, S. Hirano, and S. Tsumoto (eds). *Rough Set Theory and Granular Computing*, Vol. 125 *Studies in Fuzziness and Soft Computing*. Springer-Verlag, Berlin, Heidelberg, 2003.
- [12] T.Y. Lin. Granular computing on binary relations I. Data mining and neighborhood systems. In: L. Polkowski and A. Skowron (eds), *Rough Sets in Knowledge Discovery*, Vol. 1. Physica-Verlag, Heidelberg, 1998, pp. 107–121.
- [13] H.S. Nguyen, A. Skowron, and J. Stepaniuk. Granular computing: A rough set approach. *Comput. Intell.* 17(3) (2001) 514–544.
- [14] W. Pedrycz (ed). *Granular Computing. An Emerging Paradigm*, Vol. 70 *Studies in Fuzziness and Soft Computing*. Physica-Verlag, Heidelberg, 2001.
- [15] L. Polkowski and A. Skowron. Towards adaptive calculus of granules. In: L.A. Zadeh and J. Kacprzyk (eds), *Computing with Words in Information/Intelligent Systems*, Vol. 1. Physica-Verlag, Heidelberg, 1999, pp. 201–228.
- [16] L. Polkowski and A. Skowron. Rough mereological calculi of granules: A rough set approach to computation. *Comput. Intell.* 17 (3) (2001) 472–492.
- [17] A. Skowron and J. Stepaniuk. Information granules in distributed environment. *Lect. Notes Artif. Intell.* 1711 (1999) 357–365.
- [18] A. Skowron and J. Stepaniuk. Towards discovery of information granules. *Lect. Notes Artif. Intell.* 1704 (1999) 542–547.
- [19] A. Skowron and J. Stepaniuk. Information granules and rough-neural computing. In: S.K. Pal, L. Polkowski, and A. Skowron (eds), *Rough-Neural Computing: Techniques for Computing with Words*. Springer-Verlag, Berlin, Heidelberg, 2004, pp. 43–84.
- [20] A. Skowron, J. Stepaniuk, and S. Tsumoto. Information granules for spatial reasoning. *Bull. Int. Rough Set Soc.* 3(4) (1999) 147–154.
- [21] A. Skowron, R. Swiniarski, and P. Synak. Approximation spaces and information granulation. *Trans. Rough Sets III Lect. Notes Comput. Sci. J. Subline* 3400 (2005) 175–189.
- [22] J. Stepaniuk. Tolerance information granules. In: B. Dunin-Keplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds), *Monitoring, Security, and Rescue Techniques in Multiagent Systems*. Springer-Verlag, Berlin, Heidelberg, 2005, pp. 305–316.
- [23] Y.Y. Yao. Granular computing. *Comput. Sci. (Ji Suan Ji Ke Xue)* 31 (2004) 1–5.
- [24] J.R. Hobbs. Granularity. In: D.S. Weld and J. de Kleer (eds), *Readings in Qualitative Reasoning About Physical Systems*. Morgan Kaufman, San Mateo, CA, 1989, pp. 542–545. <http://www.isi.edu/hobbs/granularity-web.pdf>.
- [25] H.S. Nguyen. *Discretization of Real Value Attributes, Boolean Reasoning Approach*. Ph.D. Thesis. Warsaw University, Warsaw, 1997.
- [26] H.S. Nguyen and S.H. Nguyen. Discretization methods for data mining. In: L. Polkowski and A. Skowron (eds), *Rough Sets in Knowledge Discovery*, Vol. 1. Physica-Verlag, Heidelberg, 1998, pp. 451–482.
- [27] J.F. Peters, A. Skowron, and J. Stepaniuk. Nearness of objects: Extension of approximation space model. *Fundam. Inf.* 79 (3–4) (2007) 497–512.
- [28] J. Stepaniuk. Knowledge discovery by application of rough set models. In: L. Polkowski, S. Tsumoto, and T.Y. Lin (eds), *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems*. Physica-Verlag, Heidelberg, 2001, pp. 137–233.
- [29] M. Wolski. Approximation spaces and nearness type structures. *Fundam. Inf.* 79 (3–4) (2007) 567–577.
- [30] Z. Pawlak. Information systems – theoretical foundations. *Inf. Syst.* 6(3) (1981) 205–218.
- [31] Z. Pawlak. Rough Sets. *Comput. Inf. Sci.* 11 (1982) 341–356.
- [32] Z. Pawlak. *Information Systems. Theoretical Foundations (in Polish)*. Wydawnictwo Naukowo-Techniczne, Warsaw, 1983.
- [33] Z. Pawlak. *Rough Sets. Theoretical Aspects of Reasoning About Data*, Kluwer, Dordrecht, 1991.

- [34] W. Lipski. Informational systems with incomplete information. In: *Proceedings of the 3rd International Symposium on Automata, Languages and Programming*. Edinburgh University Press, Edinburgh, 1976, pp. 120–130.
- [35] W. Lipski. On semantic issues connected with incomplete information databases. *ACM Trans. Database Syst.* 4(3) (1979) 262–296.
- [36] S. Greco, B. Matarazzo, and R. Słowiński. Handling missing values in rough set analysis of multi-attribute and multi-criteria decision problems. *Lect. Notes Artif. Intell.* 1711 (1999) 146–157.
- [37] M. Kryszkiewicz. Rough set approach to incomplete information system. *Inf. Sci.* 112 (1998) 39–49.
- [38] J. Stefanowski and A. Tsoukiàs. On the extension of rough sets under incomplete information. *Lect. Notes Artif. Intell.* 1711 (1999) 73–81.
- [39] J. Stefanowski and A. Tsoukiàs. Incomplete information tables and rough classification. *Comput. Intell.* 17(3) (2001) 545–566.
- [40] I. Düntsch, G. Gediga, and E. Orłowska. Relational attribute systems. *Int. J. Hum.–Comput. Stud.* 55 (2001) 293–309.
- [41] A. Skowron and J. Stepaniuk. Tolerance approximation spaces. *Fundam. Inf.* 27(2–3) (1996) 245–253.
- [42] A. Skowron and J. Stepaniuk. Generalized approximation spaces. In: *Proceedings of the 3rd Int. Workshop on Rough Sets and Soft Computing, San Jose, CA, November 1994*, The Society for Computer Simulation, San Diego, CA, 1994, pp. 156–163.
- [43] J. Łukasiewicz. Die logischen Grundlagen der Wahrscheinlichkeitsrechnung. In: L. Borkowski (ed), *Jan Łukasiewicz—Selected Works*. North-Holland, Polish Scientific, Amsterdam, London, Warsaw, 1970, pp. 16–63. First published in Kraków in 1913.
- [44] L. Polkowski and A. Skowron. Rough mereology. *Lect. Notes Artif. Intell.* 869 (1994) 85–94.
- [45] L. Polkowski and A. Skowron. Rough mereology: A new paradigm for approximate reasoning. *Int. J. Approx. Reason.* 15(4) (1996) 333–365.
- [46] L. Polkowski and A. Skowron. Rough mereology in information systems. A case study: Qualitative spatial reasoning. In: L. Polkowski, S. Tsumoto, and T.Y. Lin (eds), *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems*. Physica-Verlag, Heidelberg, 2001, pp. 89–135.
- [47] W. Ziarko. Variable precision rough set model. *J. Comput. Syst. Sci.* 46(1) (1993) 39–59.
- [48] W. Ziarko. Probabilistic decision tables in the variable precision rough set model. *Comput. Intell.* 17(3) (2001) 593–603.
- [49] L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Trans. Fuzzy Syst.* 4(2) (1996) 103–111.
- [50] L.A. Zadeh. A theory of approximate reasoning. In: J. Hayes, D. Michie, and L.I. Mikulich (eds), *Machine Intelligence*, Vol. 9. Halstead Press, New York, 1979, pp. 149–194.
- [51] L.A. Zadeh and J. Kacprzyk (eds). *Computing with Words in Information/Intelligent Systems*, Vol. 1. Physica-Verlag, Heidelberg, 1999.
- [52] D. Dubois, L. Foulloy, S. Galichet, and H. Prade. Performing approximate reasoning with words? In: L.A. Zadeh and J. Kacprzyk (eds), *Computing with Words in Information/Intelligent Systems*, Vol. 1. Physica-Verlag, Heidelberg, 1999, pp. 24–49.
- [53] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, D.C.*, May 1993, ACM, New York, 1993, pp. 207–216.
- [54] J.G. Bazan. A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables. In: L. Polkowski and A. Skowron (eds), *Rough Sets in Knowledge Discovery*, Vol. 1. Physica-Verlag, Heidelberg, 1998, pp. 321–325.
- [55] J.W. Grzymała-Busse. LERS – A data mining system. In: O. Maimon and L. Rokach (eds), *The Data Mining and Knowledge Discovery Handbook*. Springer-Verlag, Berlin, Heidelberg, 2005, pp. 1347–1351.
- [56] J.W. Grzymała-Busse. Rule induction. In: O. Maimon and L. Rokach (eds), *The Data Mining and Knowledge Discovery Handbook*. Springer-Verlag, Berlin, Heidelberg, 2005, pp. 255–267.
- [57] M. Kryszkiewicz. Fast discovery of representative association rules. *Lect. Notes Artif. Intell.* 1424 (1998) 214–221.
- [58] H.S. Nguyen and S.H. Nguyen. Rough sets and association rule generation. *Fundam. Inf.* 40(4) (1999) 383–405.
- [59] J. Stefanowski. *Algorithms of Decision Rule Induction in Knowledge Discovery (in Polish)*, Vol. 361 *Rozprawy Wydawnictwo Politechniki Poznańskiej*, Poznań, 2001.
- [60] J.G. Bazan. Classifiers based on two-layered learning. *Lect. Notes Artif. Intell.* 3066 (2004) 356–361.
- [61] J.G. Bazan and A. Skowron. Classifiers based on approximate reasoning schemes. In: B. Dunin-Kęplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds), *Monitoring, Security, and Rescue Techniques in Multiagent Systems*. Springer-Verlag, Heidelberg, Berlin, 2005, pp. 191–202.

- [62] J.G. Bazan, S.H. Nguyen, H.S. Nguyen, and A. Skowron. Rough set methods in approximation of hierarchical concepts. *Lect. Notes Artif. Intell.* 3066 (2004) 346–355.
- [63] J.G. Bazan, A. Skowron, and R. Swiniarski. Rough sets and vague concept approximation: From sample approximation to adaptive learning. *Trans. Rough Sets V Lect. Notes Comput. Sci. J. Subline* 4100 (2006) 39–63.
- [64] S.H. Nguyen, J.G. Bazan, A. Skowron, and H.S. Nguyen. Layered learning for concept synthesis. *Trans. Rough Sets I Lect. Notes Comput. Sci. J. Subline* 3100 (2004) 187–208.
- [65] S.H. Nguyen and H.S. Nguyen. Learning concept approximation from uncertain decision tables. In: B. Dunin-Kęplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds), *Monitoring, Security, and Rescue Techniques in Multiagent Systems*. Springer-Verlag, Berlin, Heidelberg, 2005, pp. 247–260.
- [66] A. Skowron and P. Synak. Complex patterns. *Fundam. Inf.* 60(1–4) (2004) 351–366.
- [67] D. Ślezak, M.S. Szczuka, and J. Wróblewski. Harnessing classifier networks – towards hierarchical concept construction. *Lect. Notes Artif. Intell.* 3066 (2004) 554–560.
- [68] P. Synak, J.G. Bazan, A. Skowron, and J.F. Peters. Spatio-temporal approximate reasoning over complex objects. *Fundam. Inf.* 67(1–3) (2005) 249–269.
- [69] A. Gomolińska. A graded applicability of rules. *Lect. Notes Artif. Intell.* 3066 (2004) 213–218.
- [70] A. Gomolińska. A graded meaning of formulas in approximation spaces. *Fundam. Inf.* 60(1–4) (2004) 159–172.
- [71] A. Gomolińska. Satisfiability and meaning of formulas and sets of formulas in approximation spaces. *Fundam. Inf.* 67 (1–3) (2005) 77–92.
- [72] A. Gomolińska. Towards rough applicability of rules. In: B. Dunin-Kęplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds), *Monitoring, Security, and Rescue Techniques in Multiagent Systems*. Springer-Verlag, Berlin, Heidelberg, 2005, pp. 203–214.
- [73] A. Wojna. Analogy-based reasoning in classifier construction. *Trans. Rough Sets IV Lect. Notes Comput. Sci. J. Subline* 3700 (2005) 277–374.
- [74] U. von Luxburg. *Statistical Learning with Similarity and Dissimilarity Functions*. Ph.D. Thesis. Technische Universität Berlin, Berlin, 2004.
- [75] A. Gomolińska. Approximation spaces based on relations of similarity and dissimilarity of objects. *Fundam. Inf.* 79 (3–4) (2007) 319–333.
- [76] A. Gomolińska. Rough validity, confidence, and coverage of rules in approximation spaces. *Trans. Rough Sets III Lect. Notes Comput. Sci. J. Subline* 3400 (2005) 57–81.
- [77] A. Skowron, J. Stepaniuk, and J.F. Peters. Towards discovery of relevant patterns from parameterized schemes of information granule construction. In: M. Inuiguchi, S. Hirano, and S. Tsumoto (eds), *Rough Set Theory and Granular Computing*. Springer-Verlag, Berlin, Heidelberg, 2003, pp. 97–108.
- [78] J.G. Bazan. Behavioral pattern identification through rough set modelling. *Fundam. Inf.* 72(1–3) (2006) 37–50.
- [79] J. Stepaniuk, J.G. Bazan, and A. Skowron. Modelling complex patterns by information systems, *Fundam. Inf.* 67(1–3) (2005) 203–217.
- [80] The WITAS project homepage. <http://www.ida.liu.se/ext/witas/>, accessed January 2008.
- [81] J.F. Peters. Approximation spaces for hierarchical intelligent behavioral system models. In: B. Dunin-Kęplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds), *Monitoring, Security, and Rescue Techniques in Multiagent Systems*. Springer-Verlag, Berlin, Heidelberg, 2005, pp. 13–30.
- [82] A. Skowron, J. Stepaniuk, J.F. Peters, and R. Swiniarski. Calculi of approximation spaces. *Fundam. Inf.* 72(1–3) (2006) 363–378.
- [83] P. Stone. *Layered Learning in Multi-agent Systems: A Winning Approach to Robotic Soccer*. MIT Press, Cambridge, MA, 2000.
- [84] A. Skowron and J. Stepaniuk. Ontological framework for approximation. *Lect. Notes Artif. Intell.* 3641 (2005) 718–727.
- [85] S.K. Pal, L. Polkowski, and A. Skowron (eds), *Rough-Neural Computing: Techniques for Computing with Words*, Springer-Verlag, Berlin, Heidelberg, 2004.
- [86] A. Skowron and J. Stepaniuk. Information granule decomposition. *Fundam. Inf.* 47(3–4) (2001) 337–350.
- [87] A. Skowron and J. Stepaniuk. Constrained sums of information systems. *Lect. Notes Artif. Intell.* 3066 (2004) 300–309.
- [88] H.S. Nguyen, S.H. Nguyen, and A. Skowron. Decomposition of task specification. *Lect. Notes Comput. Sci.* 1609 (1999) 310–318.
- [89] J.O. Kephart. Research challenges of autonomic computing. In: *Proceedings of the 27th International Conference on Software Engineering, ICSE'2005, St. Louis, Missouri, May 2005*, The ACM, New York, 2005, pp. 15–22.

21

Spatiotemporal Reasoning in Rough Sets and Granular Computing

Piotr Synak

21.1 Introduction

Spatial, temporal, and spatiotemporal data can be found in many real-life problems investigated by many researchers around the world. Spatiotemporal data mining is a hot topic for several years covered by huge amount of related literature [1, 2]. Nevertheless, it is still a big challenge to understand complex spatiotemporal processes – to model them and to draw some interesting conclusions about them. Granulation of data as well as understanding how to utilize expert's knowledge seem to be key points to discovery of solutions of those problems.

In this chapter, we focus on some general understanding of problems related to modeling of spatiotemporal objects and reasoning about them by means of information granules satisfying, to a satisfactory degree, some spatiotemporal constrains expressed by vague concepts and dependencies between them.

We investigate these problems in the context of rough sets [3, 4] and information granulation [5–8] and we consider different kinds of granules. In the rough set approach the construction of more general granules is based on soft satisfaction of some constrains.

One of the basic tasks is to approximate complex spatiotemporal concepts often specified in a natural language. In several cases, such approximation cannot be done directly from available data, e.g., sensory measurements like signals, images, or moving images. The problem is with translation of information encoded by raw numbers directly to the satisfiability degrees of complex concepts. Hence, hierarchical (layered) methods of approximation should be used, based on hierarchy of concepts where low-level concepts are in some sense close to the input data. Information granulation plays an important role in the process of such concept approximation. It allows us to construct more general and universal structures with broader scope of application and more compact description. In the hierarchical process of complex concepts approximation, granules can be created on any level of hierarchy. Temporal nature of concepts to be approximated extorts the approximation process to take into account temporal properties of objects (or granules) and also temporal relations satisfied by objects.

The basic steps in modeling complex patterns are the following: (1) the structure of objects (granules) on a higher level is derived from structures of their components defined on the lower level and some

relations between them; (2) relevant properties of the structures on a higher level are selected from a set of formulas (from a language which should be somehow discovered) expressing their properties.

In this chapter we discuss mainly two types of granules: so-called hierarchical information maps [9] and approximate reasoning networks (AR networks) [10, 11].

From the granular computing perspective the hierarchical information maps are granules being representations (or models) of another granules and their properties relatively to some context. In several cases, this context can be determined by some partial order and interpreted in a temporal domain. Different degrees of granularity correspond to different levels of hierarchical information map. Hence, on higher levels we consider, e.g., sets of partial orders and their properties. The granules being modeled by information maps correspond to complex and structured objects as well as to their parts. Properties of those granules can be expressed by formulas of some temporal logic. The domain knowledge about objects, given, e.g., in the form of ontology can be used to determine the hierarchy of an information map.

AR networks are hierarchical patterns constructed over sensory measurements mentioned above, i.e., raw low-level information. They are discovered from hierarchical information maps and experimental data. They make it possible to approximate the domain knowledge, i.e., complex spatiotemporal concepts related to structured objects represented in hierarchical information maps. Since structured objects are relational structures, there can be considered several ways of their granulation. In this case instead of objects represented by a single structure one can consider a family of structures and its properties. The reasoning process proceeds usually in a bottom-up manner – from satisfaction degrees of low-level concepts we construct classifiers to reason about concepts from higher level.

As we mentioned above, AR networks are patterns built over hierarchical information maps. Thus, they make it possible to reason about structured (spatial) objects and also about objects evaluating over time. Simple AR networks can be easily composed to form more complex networks. Because they can contain loops, i.e., satisfaction of some concept in time t may lead to its satisfaction in time $t + \Delta t$, they are feasible to model properties of dynamical systems.

21.2 Preliminaries

Let us present some basic notions used in this chapter. In most cases we are going to use the standard notation of rough set theory [3, 12]. Thus, by $\mathbb{A} = (U, A)$ we denote an *information system* [4, 13] with the universe U of *objects* and the attribute set A . Each *attribute* $a \in A$ is a function $a : U \rightarrow V_a$, where V_a is the *value set* of a . For a given set of attributes $B \subseteq A$, we define the *indiscernibility relation* $IND(B)$ on the universe U that partitions U into classes of indiscernible objects. We say that objects x and y are *indiscernible* with respect to B if and only if $a(x) = a(y)$ for each $a \in B$.

The values of attributes for a given object $x \in U$ form an elementary pattern generated by \mathbb{A} , where an *elementary pattern* (or *information signature*) $Inf_B(x)$ is a set $\{(a, a(x)) : a \in B\}$ of attribute-value pairs over $B \subseteq A$ consistent with a given object x . By

$$INF(A) = \{Inf_B(x) : x \in U, B \subseteq A\} \quad (1)$$

we denote the set of all signatures generated by A .

Decision tables are denoted by $\mathbb{A} = (U, A, d)$, where $d \notin A$ is the *decision attribute*. The decision attribute d defines partition of the universe U into *decision classes*. An object x is *inconsistent* if there exists an object y such that $x \text{ } IND(A) \text{ } y$, but x and y belong to different decision classes, i.e., $d(x) \neq d(y)$. *Positive region* of a decision table \mathbb{A} (denoted by $POS(\mathbb{A})$) is the set of all consistent objects.

21.3 Knowledge Modeling by Using Hierarchical Information Maps

In this section we are going to consider the problem of modeling knowledge relatively to some context. For this purpose we use so-called information maps where information (knowledge) together with its context constitutes a state. Then we can consider some ordering relations on states including temporal and consequence relations as well as we can investigate the information changes over states.

In the case of knowledge modeling of structured objects, consisting of some parts being also structured objects, we can use more detailed models, describing information not only related to an object itself but also related to its parts. This more detailed information may also vary relatively to yet another context. We perform such modeling by using hierarchical information maps.

21.3.1 Information Maps

One of the tools that can be used for modeling of information dependently on some (e.g., temporal) context is information map [14, 15]. Information maps are usually generated from experimental data (e.g., information systems or decision tables) and are defined by some binary (transition) relations on the set of states. In this context a state consists of an information label and the corresponding information extracted from a given data set. This kind of structure provides basic models over which one can search for relevant patterns for many data mining problems [14, 15].

An *information map* \mathcal{A} is a quadruple (E, \leq, I, f) , where E is a finite set of *information labels* defining context of modeled information, $\leq \subseteq E \times E$ is a binary *transition relation* on information labels, I is an *information set*, and $f : E \rightarrow I$ is an *information function* associating any information label with the corresponding information.

Example 1. In Figure 21.1a, we present an example of information map, where $E = \{e_1, e_2, e_3, e_4, e_5\}$, $I = \{f(e_1), f(e_2), f(e_3), f(e_4), f(e_5)\}$, and the transition relation \leq is a partial order on E .

A *state* is any pair $(e, f(e))$, where $e \in E$. The set $\{(e, f(e)) : e \in E\}$ of all states of \mathcal{A} is denoted by $S_{\mathcal{A}}$. The transition relation on the set of information labels can be extended to the relation on states, e.g., in the following way: $(e_1, i_1) \leq (e_2, i_2)$ if and only if $e_1 \leq e_2$. A *path* in \mathcal{A} is any sequence $s_0 s_1 s_2 \dots$ of states such that for every $i \geq 0$: $s_i \leq s_{i+1}$ and if $s_i \leq s \leq s_{i+1}$, then $s = s_i$ or $s = s_{i+1}$.

We say that a state s is *reachable* from a state s_0 if and only if there exists a path $s_1 s_2 \dots s_n$, such that $s_1 = s_0$ and $s_n = s$. A *property* of \mathcal{A} is any subset of $S_{\mathcal{A}}$. Let F be a set of temporal formulas. We say that the property φ is *expressible* in F if and only if $\varphi = \|\alpha\|$ for some $\alpha \in F$, where $\|\alpha\|$ is the meaning of α .

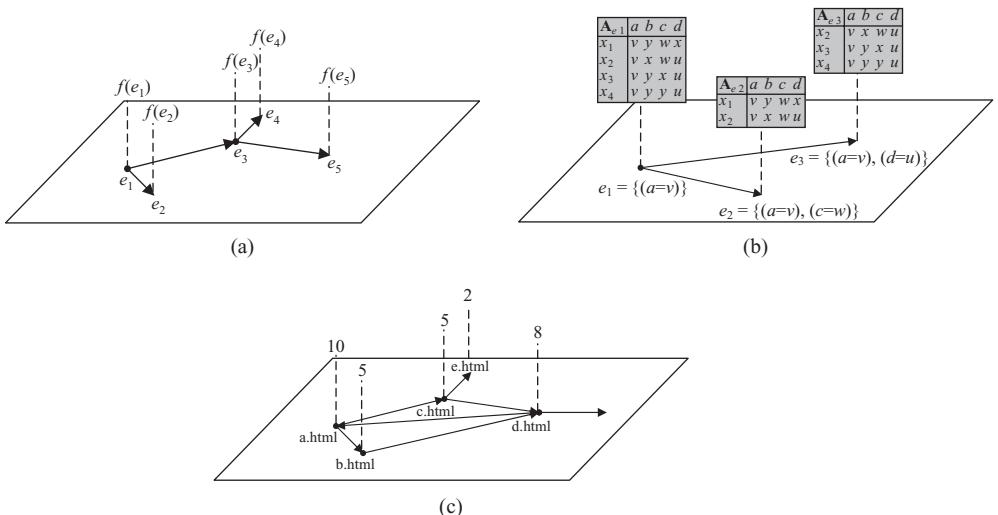


Figure 21.1 (a) An information map: labels are denoted by e_1, e_2, \dots and the corresponding information by $f(e_1), f(e_2), \dots$; (b) an information map of an information system; (c) sessions of users visiting a Web site

21.3.2 Information Maps of Data Tables

Any information system $\mathbb{A} = (U, A)$ defines its information map as a graph consisting of nodes that are elementary patterns generated by \mathbb{A} . By elementary pattern one can use particular information signature $Inf_B(x)$ (see Section 21.2) related to some set of attributes $B \subseteq A$ and consistent with a given object $x \in U$. Thus, the set of labels E is equal to the set $INF(A) = \{Inf_B(x) : x \in U, B \subseteq A\}$ of all elementary patterns of \mathbb{A} . The relation \leq is defined in a straightforward way; i.e., for $e_1, e_2 \in INF(A)$, $e_1 \leq e_2$ if and only if $e_1 \subseteq e_2$. Hence, relation \leq is a partial order on E . Finally, the information set I is equal to $\{\mathbb{A}_e : e \in INF(A)\}$, where \mathbb{A}_e is a subsystem of \mathbb{A} with the universe U_e equal to the set $\{x \in U : \forall (a, t) \in e, a(x) = t\}$. Attributes in \mathbb{A}_e are the same as in \mathbb{A} but with domain restricted to U_e . The information function f mapping $INF(A)$ into I is defined by $f(e) = \mathbb{A}_e$ for any $e \in INF(A)$ (see Figure 21.1b).

One can investigate context-dependent properties of an information system modeled by using information map, e.g., properties related to the distribution of values of some attribute.

Example 2. Let α be a formula such that $(e, \mathbb{A}_e) \models \alpha$ has the following intended meaning: ‘at least 75% of objects of system \mathbb{A}_e have value u on attribute d .’ In the example presented in Figure 21.1b, $\|\alpha\|_{\mathbb{A}} = \{(e_1, \mathbb{A}_{e_1}), (e_3, \mathbb{A}_{e_3})\}$ [15].

Moreover, other information functions for information maps over data tables are possible. Such a function can be a kind of ‘view’ of dependencies in the data table. Then, e.g., $f(e)$ can be equal to the set of all dependencies in \mathbb{A}_e that have sufficient support and confidence.

21.3.3 Information Maps for Web Traffic Data

In previous example, the transition relation of an information map was a partial order. This is not the case in general. If we consider information map defined for web traffic data, we can see that this relation is not antisymmetric and transitive.

Let us consider a web server hosting several web pages linked to each other. We can assume the set E to be the set of all URLs of this server. The transition relation \leq can then be determined by links between pages; i.e., $e_1 \leq e_2$ if and only if there is a link from page e_1 to e_2 . In this case an information map of such a web server would reflect all the possible links between pages. We can define the information function f as one associating each page of the server with some information about the given URL. For example, when investigating traffic on the server and users’ sessions, such an information can be the number of sessions going through a page (see Figure 21.1c) or all of the session identifiers.

Example 3. One can be interested in patterns describing traffic on the server in terms of common pages visited by users of the same profile. The following formula illustrates this idea: ‘50% of all sessions (of a given profile) initially go through the page e_i and next through the page e_k .’

Another example of information map can be obtained for web mining problem (see [16–18]) corresponding to one user session. Then the set of labels of such a map is a subset of some global set of labels E_0 , in this case corresponding to all URLs of a given web server. Each user session determines some transitions of a particular map and from one page a user can go to more than one next page (e.g., opening a new session track in a new window of the browser). For such a family of maps we can consider several problems related to finding patterns describing characteristic pages of the server.

Several other examples of information maps can be found, e.g., in [15].

21.3.4 Decision Tables over Information Maps and Information Granulation

One of the typical schemes of object classification is based on the analysis of decision tables. By using the information given about an object (e.g., some object pattern), we try to classify it to a proper decision class. In many cases this scheme needs to be extended because the context of the information should be

considered together with the information itself. It means that instead of a single information signature relative to the investigated object x , we also have to examine some other objects that are in some relation to x . Properties of those objects can be important in order to extend information about x by information about the context in which x occurs. In a more complex case, we can consider states of objects and relations between such states. Temporal relations between states, in the case of objects changing over time, provide another possible source of information about the context in which objects occur.

Thus, the scheme of object classification can be constructed as follows. We are given a decision table. Next, we extend it by some relations on objects (or values of attributes) to a relational decision table defining some neighborhoods of objects (possibly overlapping each other). Thus, we construct a new decision table, where objects are pairs (*object, object_neighborhood*), and attributes describe properties of the objects in the context of their neighborhoods.

In the case of information maps, the above idea is generalized to more complex information granules that are pairs (*state, state_neighborhood*), where *state* is a state of a given information map \mathcal{A} and *state_neighborhood* is the neighborhood of this state in \mathcal{A} . A state can be identified by some information about an object and it determines some set of objects (a subtable), e.g., set of objects indiscernible by means of some attributes. Thus, *state_neighborhood* is a much more complex structure than *object_neighborhood* in the previous case, because it is a set (defined by transition relation) of subtables satisfying some constraints. Also the attributes of the constructed decision table are more complex because they express properties of complex neighborhoods. The decision attribute is complex as well because it classifies a state which is a complex object (in our example – a subtable). Thus, for a given state s , we can consider, e.g., the distribution of objects corresponding to s in decision classes as the value of decision for s .

21.3.5 Structured Objects Changing over Time

One of the fundamental concepts considered in the previous sections is the notion of an object. Objects are some real entities that can be described by some physical observations or measurements. An object can be though identified with some information about it, i.e., with some vector of measurements. From this interpretation it follows that one vector of measurements can describe several objects. From the point of view of this information only, such objects are indiscernible, although in fact they can be different. This way of understanding objects is used in the rough set theory where the indiscernibility relation is utilized in many different ways, and the mentioned vector of measurements is called object signature.

In a bit more complex case we can consider structure of objects. Structured (complex) objects consist of some parts which can be constrained by some relations of different nature, e.g., spatial relations. The parts can be built from some simpler parts and therefore the structure can be hierarchical with many different levels of hierarchy. The relation object-part corresponds in most cases to some spatial relation. These problems are considered in rough-mereological approach [19, 20]. There can be considered several examples of structured objects, e.g., a car, a human system, and situation on a road.

Some structured objects are static, while some other dynamic. The latter case means that a structured object may evaluate (change) over time. The most complex case seems to be when properties of an object and its parts as well as its structure change dynamically.

Example 4. A very good and illustrative example is the mentioned above situation on a road (see, e.g., [11, 21]). One can observe cars on some street crossing. They form a very complex and dynamic object. It consists of some parts (particular groups of cars) that can consist of yet another parts (smaller groups of cars). Each part, including the whole object, may have various properties changing over time. And, moreover, the structure of the object can change dynamically.

Modeling and reasoning about such complex and dynamic objects is a very challenging and important issue.

21.3.6 Hierarchical Information Maps

One possibility of modeling structured objects evaluating over time is to use some multilevel relational structure. A hierarchical information map is an example of such a structure. It consists of several levels, each modeling temporal behavior of parts from the same level of the object's structure. Every part of a complex (structured) object defines its own space of states together with the corresponding transitions. Thus, on each level we keep several graphs – one graph for one part. The edges of these graphs are labeled with some temporal relations; however, they are defined for particular parts. The lowest level of the map corresponds to elementary (atomic) parts.

We connect the nodes of graphs from adjacent levels by some spatial relation, defining schemes of constructing a more complex object in a given state from its parts (which are also in some states).

Example 5. An example is presented in Figure 21.2. A complex object in state v_1 consists of two parts that are in states x_1 and y_1 . The same object in state v_3 consists of three parts in states x_3 , y_2 , and z_2 , respectively.

With each non-atomic part in some state x_i at any level, we can associate a decision table containing, e.g., information about historical observations of this part in x_i . The rows (objects) of such a system correspond to different observations.

The presented structure – multilevel hierarchical information maps – consists of several information maps that are linked together by some relations on the sets of states. It is important to note that in modeling of such maps we express properties of states and relations between them using the language of domain knowledge (e.g., a simplified natural language). Next, using hierarchical information maps and

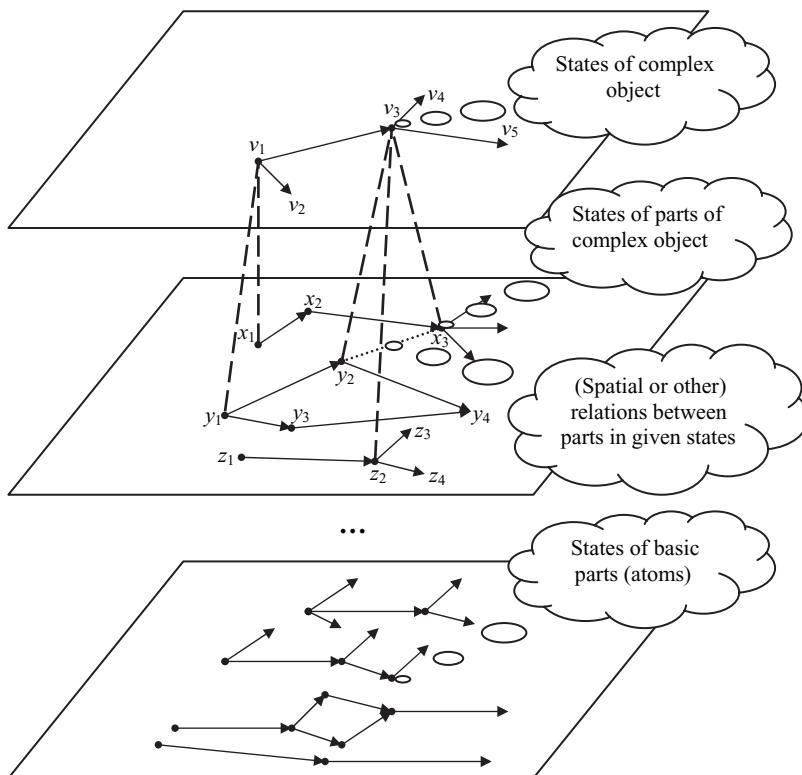


Figure 21.2 An example of hierarchical information map

experimental data we are searching for AR networks (see next sections), representing relevant patterns for approximation of complex concepts that appear on different levels of maps. Such AR networks are constructed along the derivations performed in domain knowledge using the representation in hierarchical information maps.

In the most general case, there can be also given some other relations defined between parts from the same level, e.g., spatial or temporal, reflecting some constraints which should be satisfied by parts in given states in order to reason about more complex object (see Figure 21.2). For example, the state of an object can change from safe to unsafe if its parts are in some particular states and, additionally, if they are too close to each other.

21.4 Hierarchical Reasoning about Complex Spatiotemporal Concepts

21.4.1 Reasoning about Concepts

In previous sections we were investigating the problem of modeling structured objects dynamically changing over time. Once the model is created we can consider the problem of reasoning, i.e., driving conclusions about properties of objects. The properties are often called *concepts* and when an object has some property we usually say that it *satisfies* a concept. For example, we can say that some men satisfy the concept of being tall or the observed situation on a road crossing satisfies the concept ‘safe situation.’ Let us emphasize that in the latter case we have a very complex concept (although briefly formulated), describing property of a structured object evaluating over time, so we can say that it is a complex spatiotemporal concept.

The reasoning process is thus a process leading to the conclusion whether given concept is satisfied by some object or not. However, in a real life it can be quite difficult or not feasible at all to check univocally if a concept is satisfied or not. In such a case some AR methodology has to be applied. By using approximate methods we obtain approximate answer, i.e., specification of a degree of satisfiability, and we say that a concept is satisfied to a certain degree.

21.4.2 Structured Reasoning Schemes

One of the possibilities to approximate concepts related to structured objects is to use so-called *approximate reasoning schemes* (AR schemes) [22–24]. Such schemes usually have a tree structure, with the root labeled by the satisfiability degree of some feature by a complex object and leaves labeled by the satisfiability degrees of some other features by primitive objects (i.e., the most simple parts of a complex object). An AR scheme can have many levels. Then, from properties of basic parts we conclude about properties of more complex parts and, after some levels, about properties of the complex target object.

An AR scheme is constructed from labeled approximate rules, called productions. Productions can be extracted from data using domain knowledge. We define productions as parameterized implications with premises and conclusions built from patterns sufficiently included in the approximated concept.

Example 6. In Figure 21.3, we present an example of production for some concepts C_1 , C_2 , and C_3 approximated by three linearly ordered layers small, medium, and large. This production is a collection of three simpler rules, called production rules, with the following interpretation: (1) if inclusion degree to a concept C_1 is at least medium and to a concept C_2 at least large, then the inclusion degree to a concept C_3 is at least large; (2) if the inclusion degree to a concept C_1 is at least small and to a concept C_2 at least medium, then the inclusion degree to a concept C_3 is at least medium; (3) if the inclusion degree to a concept C_1 is at least small and to a concept C_2 at least small, then the inclusion degree to a concept C_3 is at least small.

The concept from the upper level of production is called the *target concept* of production, while the concepts from the lower level of production are called the *source concepts* of production. For example, in the case of production from Figure 21.3, C_3 is the target concept and C_1 and C_2 are the source concepts.

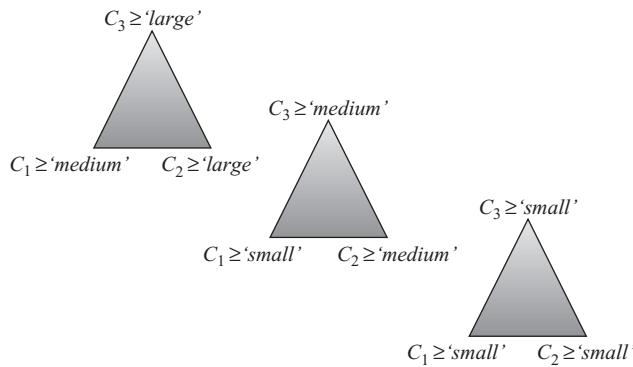


Figure 21.3 An example of production as a collection of three production rules

One can construct an AR scheme by composing single production rules chosen from different productions from a family of productions for various target concepts. In Figure 21.4, we have two productions. The target concept of the first production is C_5 and the target concept of the second production is the concept C_3 . We select one production rule from the first production and one production rule from the second production. These production rules are composed and then a simple AR scheme is obtained that can be treated as a new two-level production rule. Notice that the target pattern of lower production rule in this AR scheme is the same as one of the source patterns from the higher production rule. In this case, the common pattern is described as follows: inclusion degree (of some pattern) to a concept C_3 is at least *medium*.

In this way, we can compose AR schemes into hierarchical and multilevel structures using productions constructed for various concepts.

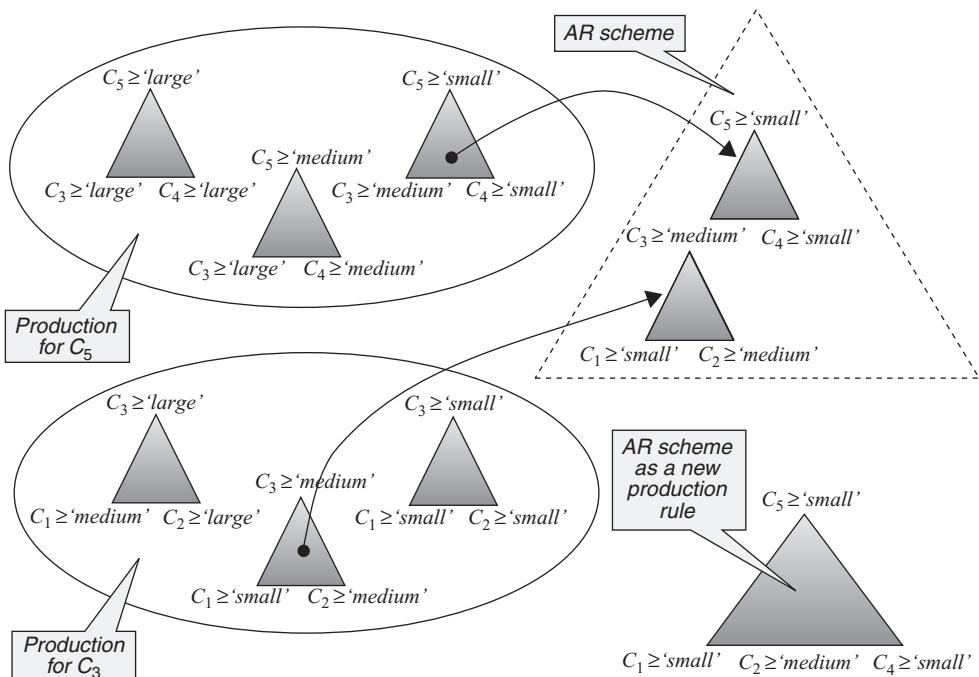


Figure 21.4 Synthesis of AR scheme

21.4.3 Related Problems

The analysis of structured objects and reasoning about related concepts is a very important topic nowadays. Let us formulate a few problems related to such an analysis.

Feature Selection and Extraction

One of the crucial point of reasoning process is to develop methods searching for relevant features for concept approximation. Such features can be selected from a given set of available features. In a more general case, one can first try to discover a set of features from which the relevant features are next selected. This is because the space of all features can be very large and from practical point of view searching for relevant features in the whole space is not feasible. These are the two basic steps in classifiers construction in machine learning, pattern recognition, and data mining known as the feature selection and extraction [25–27]. Features used for object description are expressed by formulas from some language – its choice depends on the investigated problems and, in particular, on a context in which we should consider objects. The problem of relevant language discovery is a challenging problem in data mining.

We propose to use soft domain knowledge for relevant feature extraction for complex concept approximation. Domain reasoning schemes are treated as hints in searching for such relevant features. On the basis of such schemes we propose to induce AR schemes representing patterns relevant for complex concepts. Characteristic functions of such patterns can be used as relevant features. By collecting AR schemes, relevant for a given concept, we can gradually approximate the concept. Searching for relevant features for complex concept approximation without domain knowledge is infeasible because of the very huge search space from which one should extract such features. This aspect is also related to a general discussion in statistical community about the necessity to develop a new statistical approach to deal with complex real-life problems (see, e.g., [28, 29]).

Observe that, in many cases, we can measure only selected number of features (satisfiability degrees of some concepts) what makes the given information about objects incomplete. Moreover, satisfiability degree of some features may be estimated only partially (i.e., to some degree), and therefore for expressing the satisfiability of corresponding formulas, multivalued logic should be used.

Structure Discovery

Another very important problem is related to the relevant structure discovery of complex objects, taking into account the context in which they appear. Such a structure description, usually unknown and very complex, depends on the chosen description language. The choice of another description language can change the object perception. To discover the relevant perception of the object structure is very hard because of a very huge space of possible structures from which it is necessary to search for the relevant ones. Moreover, in many cases we can observe only some features of some parts of objects and relations (or their approximations) they satisfy; the features of different parts can be expressed in different languages. All these aspects are closely related to perception problems (see, e.g., [30–33]). Note that nowadays it is a growing expectation that existing knowledge about perception in psychology and neuroscience can lead to feasible searching strategies for relevant perception of complex objects. In our approach we propose to support the object perception using soft domain knowledge. The developed AR schemes on the basis of domain knowledge can be considered as perception schemes [23, 31–33].

Thus, to reason about the structure of a complex object, we have to use composition (one- or many-level) schemes of its parts. This kind of structure should also make it possible to reason about inclusion degrees of the complex object in higher level concepts from inclusion degrees of its parts in lower level concepts, assuming that these parts satisfy some additional constraints (e.g., expressing closeness of parts in a considered space). These problems are considered in rough-mereological approach [19, 20].

21.4.4 Approximate Reasoning Networks

For our further discussion it is important to note that we describe states of objects by some of their properties. Hence, we identify states with collections of objects defined by these properties rather than with single objects. The analogous assumption about parts of objects is made.

Let us now discuss the class of problems for the case when a given object evaluates over time, which is measured by changes of some of its features over time [2]. We are going to consider different states of an object at different time points and some transitions between states. For a complex object, its structure can be perceived in each state. This structure can evaluate from state to state, which can be expressed in different ways. In the simplest case we observe changes of inclusion degrees in patterns (formulas) describing features of parts. In a more complex case, the structure itself can also change and some parts can be replaced by other ones. The language in which features are expressed can also evaluate, either in relation to the whole complex object or in relation to some of its parts. Problems related to this subject are widely studied in the literature (see, e.g., [2, 34]).

To model changes of complex objects over time, we use hierarchical information maps (see Section 21.3.6). AR networks [10, 35], i.e., spatiotemporal AR schemes, make it possible to approximate (in a language definable in terms of attributes available from data, e.g., sensor measurements) reasonings about spatiotemporal properties of objects performed by an expert in his/her language (e.g., a simple fragment of a natural language). They differ from the static case of AR schemes in that they are also constructed along the time dimension. We assume that AR networks are discovered from data and domain knowledge.

Figure 21.5 illustrates a fragment of a simple AR network. A graph on a plane represents states of a complex object by means of some concepts labeled by minimal satisfaction degrees. With each node there is associated an AR scheme – observe that in the construction of patterns the classifiers induced before can also be involved. It is used to approximate the degree in which a complex object matches a given concept by using the information about satisfaction degrees of lowerlevel concepts by elementary (sensory) objects. Nodes are linked by edges labeled by temporal relations, which can depend not only on time but also on the context in which objects appear. In the simplest case, the temporal relations can be a time-consequence relation.

Rules used in construction of AR networks link some spatiotemporal patterns describing spatiotemporal properties of objects or their parts. They can be used for prediction of inclusion degrees of objects into some patterns related to the future, from information about degrees of inclusion of objects in some other spatiotemporal patterns in the past. Patterns are expressed by formulas of some languages and some temporal operators can be used for expressing their relationships in time.

In Figure 21.6, we can see an illustration of the following spatiotemporal rule: $C_1(x_1(t)) \wedge C_2(x_2(t)) \wedge Time(t, \Delta t) \Rightarrow C_3((x_1 \oplus x_2)(t + \Delta t))$. The interpretation of the rule is as follows: if parts x_1 and x_2 match to a satisfactory degrees the concepts C_1 and C_2 at time t , respectively, and some time constraint $Time(t, \Delta t)$ is additionally satisfied, then the complex object constructed from these parts, $(x_1 \oplus x_2)$, matches the concept C_3 at time $t + \Delta t$ to a satisfactory degree. Such rules are extensions of production rules to the spatiotemporal case. Their compositions define AR networks.

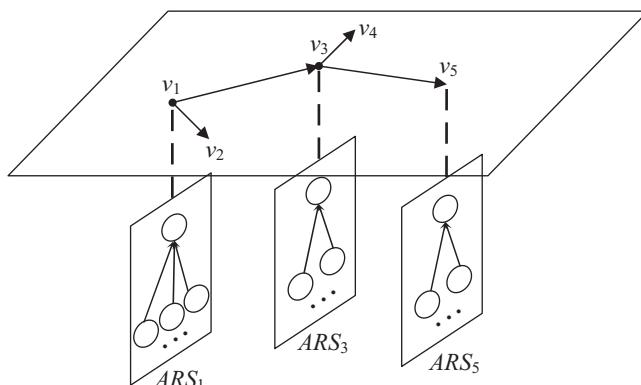


Figure 21.5 An example of a simple AR network

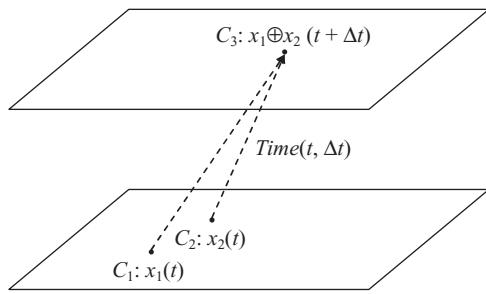


Figure 21.6 Reasoning about complex object $x_1 \oplus x_2$ at time $t + \Delta t$ based on properties of its parts x_1 and x_2 at time t

Observe that any AR network determines some pattern used for approximation of a given concept. To obtain good approximation several patterns (AR networks) ‘for’ and ‘against’ the approximated concept should be discovered and next fused to form a classifier for this concept.

One of the schemes of reasoning by using AR networks is as follows. A hierarchical information map represents the domain knowledge about different states of some complex object. Each state is identified by some property of the object, i.e., some concept. To measure degree of satisfaction of a concept the AR schemes are used. Suppose we have measurements (e.g., obtained from some sensors) of the current (observed) situation. We test each AR scheme to determine a concept matched the best in order to identify the state of the complex object. Next, from the hierarchical information map, we conclude what the possible next states are, which gives the opportunity to undertake some action.

In general, the information given about the states can be broader. It might be based on some historical knowledge (training data) [2, 26]. Each state can be additionally described by some information system, where each attribute corresponds to some of the state properties. In particular, an attribute can reflect satisfaction of some properties in the near past. Thus, a given state can be described not only by spatial patterns but also by spatiotemporal ones.

As we have already mentioned, the AR networks form a kind of patterns for approximation of complex concepts. With each concept there are associated AR networks matching the concept to a satisfactory degree. An important problem is to select all AR networks matched by the currently observed situation. The size of spatiotemporal knowledge base can be huge and, thus, the process of choosing the most relevant AR network can be complex.

Example 7. Let us consider a problem of an automatic situation tracking on the road by an unmanned aerial vehicle (UAV) [36]. Suppose a UAV monitors two vehicles going on the road close to each other, and using this observation it is supposed to evaluate their situation, e.g., in the context of safety. Different approximation schemes describe their mutual location, e.g., ‘first behind second’ and ‘first on the left of second.’ Thus, we have a collection of AR networks ARN_1, \dots, ARN_n , describing typical situations in which two vehicles may occur. The UAV, while monitoring the vehicles, tries to find an AR network matching the observed situation to the highest degree. The result of monitoring of the vehicles in time is then a sequence

$$\langle (ARN_{i_1}, d_1, t_1, Time(t_1, t_2)), \dots, (ARN_{i_{k-1}}, d_{k-1}, t_{k-1}, Time(t_{k-1}, t_k)) \rangle,$$

where for $j = 1, \dots, k - 1$, the network ARN_{i_j} is matched by the observed situation to the highest degree d_j (among all considered AR networks) at time t_j , and $Time(t_j, t_{j+1})$ are time constraints satisfied for $j = 1, \dots, k - 1$. These time constraints can have a more complex form. In particular, they can depend on properties of AR networks linked with them.

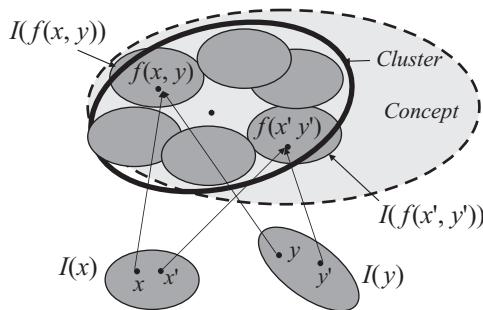


Figure 21.7 Relational structure granulation

21.4.5 Reasoning and Relational Structure Granulation

Let us discuss an important role which the relational structure granulation [23, 37] plays in searching for relevant patterns in approximate reasoning, e.g., in searching for relevant approximation patterns (see Figure 21.7).

Approximation Spaces

One of the basic concepts of rough set theory is the indiscernibility relation, defined by means of information about objects, which is used to define set approximations. There have been reported several generalizations of the rough set approach based, e.g., on approximation spaces defined by tolerance and similarity relation, or a family of indiscernibility relations [38, 39]. Rough set approximations have also been generalized for preference relations and rough-fuzzy hybridizations (see, e.g., [40]). Let us consider a generalized approximation space introduced in [41]. It is defined by $AS = (U, \mathcal{I}, v)$, where U is a set of objects (universe), \mathcal{I} is an *uncertainty function* defined on U with values in the powerset $P(U)$ of U (e.g., $\mathcal{I}(x)$ can be interpreted as a *neighborhood* of x), and v is an *inclusion function* defined on the Cartesian product $P(U) \times P(U)$ with values in the interval $[0, 1]$ (or more generally, in a partially ordered set), measuring the degree of inclusion of sets. The lower AS_* and upper AS^* approximation operations can be defined in AS by

$$AS_*(X) = \{x \in U : v(\mathcal{I}(x), X) = 1\}, \quad (2)$$

$$AS^*(X) = \{x \in U : v(\mathcal{I}(x), X) > 0\}. \quad (3)$$

The neighborhood of an object x can be defined by the indiscernibility relation IND . If IND is an equivalence relation, then we have $\mathcal{I}(x) = [x]_{IND}$. In the case where IND is a tolerance (similarity) relation $\tau \subseteq U \times U$, we take $\mathcal{I}(x) = \{y \in U : x\tau y\}$; i.e., $I(x)$ is equal to the tolerance class of τ defined by x . The standard inclusion function is defined by $v(X, Y) = \frac{|X \cap Y|}{|X|}$ if X is non-empty and by $v(X, Y) = 1$ otherwise. For applications it is important to have some constructive definitions of \mathcal{I} and v . The approach based on inclusion functions has been generalized to the *rough-mereological approach* (see, e.g., [19, 20]). The inclusion relation $x\mu_r y$ with the intended meaning x is a part of y to a degree r has been taken as the basic notion of the rough mereology, which is a generalization of the Leśniewski mereology [42].

Concept Approximation

Let $AS = (U, \mathcal{I}, v)$ be an approximation space. For any object $x \in U$, there is defined a neighborhood $I(x)$ specified by the value of the uncertainty function from AS . From these neighborhoods, by applying different kind of operations, some other, more relevant ones (e.g., for the considered concept approximation) should be determined. Such neighborhoods can be extracted by searching the space of neighborhoods generated from values of the uncertainty function. The possible classes of operations

include some generalization operations, set-theoretical operations (union and intersection), clustering, and operations on neighborhoods defined by functions and relations in the underlying relational structure. In the latter case, relations from such a structure may define relations between objects or their parts.

Figure 21.7 illustrates an exemplary scheme of searching for neighborhoods (patterns and clusters) relevant for concept approximation. In this example, f denotes a function with two arguments from the underlying relational structure. Due to the uncertainty, we cannot perceive objects exactly but only by using available neighborhoods defined by the uncertainty function from an approximation space. Hence, instead of the value $f(x, y)$ for a given pair of objects (x, y) , one should consider a family of neighborhoods $\mathcal{F} = \{I(f(x', y')) : (x', y') \in I(x) \times I(y)\}$. From this family \mathcal{F} , a subfamily \mathcal{F}' of neighborhoods can be chosen, which consists of neighborhoods with some properties relevant for approximation. Next, a subfamily \mathcal{F}' can be, e.g., generalized to clusters that are relevant for the concept approximation, i.e., clusters sufficiently included into the approximated concept (see Figure 21.7). The inclusion degrees can be measured by granulation of the inclusion function from the relational structure.

Let us emphasize that the above ideas can be analogously applied to construction of higher levels of information maps (see Section 21.3.6). Using information granulation one can construct from a given information map a new one at the higher level, which is simpler (more compact) but still sufficient for approximation of complex concepts with a satisfactory quality.

21.5 Planning Based on Hierarchical Reasoning

21.5.1 Concepts, States, and Actions in the Reasoning Process

In this section let us present some general scheme of reasoning about complex concepts that are satisfied to an unsatisfactory degree. Such a case can be a result of some changes of the situation over time and may be required to undertake appropriate actions. Let U be a universe of objects and C be a given concept. For example, we can consider a set of patients as U and a concept of having given disease as C . Let us also denote by $\neg C$ the complementary concept to C – in our example the concept of not having given disease. Now, we can consider some set $X \subseteq U$ of objects included into C to a satisfactory degree, as well as $Y \subseteq U$ – the set of objects well included into $\neg C$.

A given situation can be changing dynamically over time which we can refer to by states of an object. We can observe that in some states the concept C is satisfied, while in some other states is $\neg C$. It means that there is additionally some transition relation $R \subseteq U \times U$ responsible for the process of transformation of objects from the set X to the set Y . Thus, we can say that $Y = XR = \{y \in U : \exists x \in X \ xRy\}$.

The reasoning about concepts related to X , here C , is performed in a hierarchical manner by means of some patterns (see Figure 21.8) and classifiers constructed using language of those patterns. In a similar way, one would have to construct a hierarchical classifier for approximation of a concept related to relation R , namely, a concept satisfied by relation R to a satisfactory degree. Such a classifier, for a given pair of objects (x, y) , where $x \in X$, $y \in Y$, must take into account (1) properties of x by means of relevant patterns constructed for X , (2) properties of y by means of relevant patterns constructed for Y , and (3) properties of the pair (x, y) by means of relevant patterns constructed for R . (Note that those patterns can be defined in a language much different from those in the other two cases; e.g., we can consider a closeness between x and y .)

Let us emphasize that, in general, the situation can be much more complex. It can be impossible to approximate such a relation R that directly moves us from set X to Y . We can rather expect to be able to approximate a relation that moves us into ‘the right direction,’ i.e., to the state where the change of satisfaction degree of some concept is desired. It means that being in the state where satisfaction degree of concept C is high, we have transition that moves us to the state where this degree is lower; i.e., change of degree is negative. We iteratively use several such transition to move in a direction of low satisfiability of C and high satisfiability of $\neg C$.

The considered problem is related to the following situation: the reasoning about an investigated object leads us to the conclusion that the object does not satisfy a given concept to a satisfactory degree. We would like to impose such a change that the concept is satisfied. What comes handy is a set of available actions that we can perform to change some properties of an object. The actions are given in the form

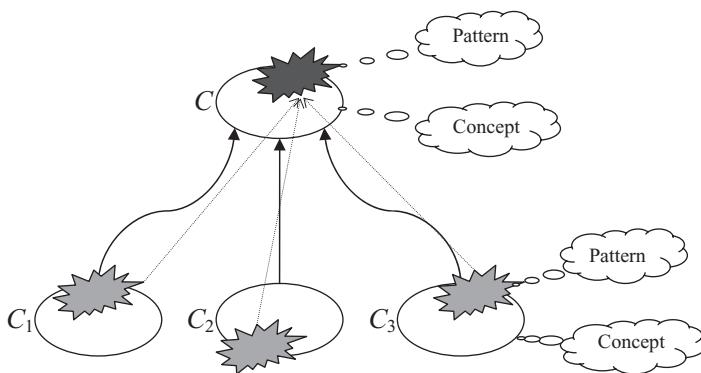


Figure 21.8 Approximation of complex concept C by using patterns constructed from patterns approximating low-level concepts C_1, C_2 , and C_3

of rules where premise describes objects to which given action can be applied, and conclusion specifies what will be changed after the action is triggered. In our example we can consider a patients having some disease (thus, satisfying C). We would like to undertake some actions to treat the patient, so it satisfies $\neg C$ into satisfactory degree. An action could correspond in this case to application of some medicine. A set of action is then a plan of therapy.

We can easily see that there can be induced several transition relations (several paths) that some of their compositions lead a given object from set X to set Y . Let us emphasize that in each step from the pattern matched by the object and the pattern approximating transition relation we can decode the pattern matched by the transformed object. In this way, we obtain an input pattern for the next step. In each step, an object is transformed into a new state in which the satisfaction degree of a considered concept is better. However, it can appear that one or more steps of one path leads to a worse state. This can be necessary due to necessity of avoiding locally optimal states. Some heuristic search engines, including genetic algorithms and simulated annealing methods, can be utilized to generate optimal paths. Each path obtained should be additionally verified whether it is feasible. In particular, each step of a path should be verified if there are available actions, making it possible to realize this step. There should be also considered costs of performing actions realizing all steps of a given path. Thus, the cost of a path and the quality of destination (by means of satisfaction degree of considered concept) of state should be evaluated while choosing the optimal path.

21.5.2 Some Detailed Issues

Let us explain some details related to the possible realizations of the presented ideas. We assume that the investigated objects define some information system $\mathbb{A} = (U, A)$. The considered concept and its complement is denoted by C and $\neg C$, respectively.

Training Data and Training Process

Each object from the training information system contains information about inclusion degrees to concepts C and $\neg C$. There are induced several hierarchical AR schemes $\{AR_i\} = \{AR_i^C\} \cup \{AR_i^{\neg C}\}$. The input nodes of the schemes are corresponding to the low-level concepts approximated by means of patterns $\{r_i\}$. The set of patterns used in a given AR scheme depends on the low-level concept used in this scheme; however, any object from U can be tested against any pattern.

Actions and Transition Relation Approximation

One of our assumptions is that we can have an influence on the properties of objects by means of some actions $\{ac_i\}$ we can undertake. Each action can have a cost associated with its execution. In the simplest case, an action can be precisely defined in terms of descriptors over the set of attributes A . Thus, each

action can have a form of implication where the premise describes the properties of objects for which the action can be triggered, while the conclusion defines the changes of object's properties. An example of such action is ' $a_1 = 5$ and $a_5 < 7 \Rightarrow a_1 > 10$ and $\Delta a_8 < 5$ ', where $a_i \in A$.

Actions can be obtained in several ways. In many cases, they are just extracted from domain knowledge. Experts can define such rules of changes of attribute values based on their knowledge, experience, and observations of historical cases. From the other hand, actions can be automatically generated from data. One method is based on extraction of so-called action rules [43, 44], i.e., rules of form ' $\omega \wedge [\alpha \rightarrow \beta] \Rightarrow [\phi \rightarrow \psi]$ ', where $\omega, \alpha, \beta, \phi$, and ψ are descriptors over attributes. In particular, α and β are defined over so-called *flexible* attributes that can be influenced by undertaking some action. Such action rules can be generated by combining selected association rules. Observe that we can easily transform action rules into actions in the form defined above.

In a more complex case, we do not know precise definitions of actions but have some training data describing objects before and after action's execution (e.g., we have characteristics of patients before and after application of some medicine). Thus, we also need to induce an AR scheme AR_0 approximating the concept that a given action ac is triggered. AR_0 is then a kind of approximation of transition relation between states of an object where the transition is forced by action ac . The low-level (input) concepts of obtained AR scheme AR_0 are approximated by patterns R_l^{ac} and R_r^{ac} describing properties of objects before and after execution of ac , respectively. Let us also emphasize that some of the low-level concepts can describe properties of pair of objects (x, x') . Those concepts are approximated by yet another set of patterns R_{lr}^{ac} .

In consequence, for a given object x matching patterns R_l^{ac} , we can use scheme AR_0 to decode patterns matched by x after we apply action ac . In this way, we have some approximation of an action in the language of patterns over the set of attributes A .

Reasoning Process

Let x be an investigated object which has been evaluated by induced AR schemes $\{AR_i\}$ as satisfying C . It means that it could be recognized by some schemes from $\{AR_i^C\}$ (let us denote them by AR^C) but also by schemes $AR^{-C} \subseteq \{AR_i^{-C}\}$. (In such a case conflict-resolving strategies should be involved.) The main problem is to find a sequence of actions that should be undertaken in order to transform object x into x' such that x' satisfies C to a satisfactory degree.

One possible way of reasoning is as follows. By examining schemes AR^C and $\{AR_i^{-C}\} \setminus AR^{-C}$, as well as the conflict-resolving strategy, we can select (1) key schemes that recognized and evaluated x as matching C and (2) schemes that could strongly 'vote' for $\neg C$ but some of the input concepts were not matched by x good enough. Then, we can decide the way we want to change x . In the first case, we may force x not to match some patterns previously matched, so we can eliminate some schemes from AR^C . In the second case, we may force x to match some patterns previously not matched, so we can add some 'strong' schemes to AR^{-C} .

In either cases, we have some patterns matched by x and some target patterns we would like to be matched by transformed x . Thus, we can try to iteratively combine AR schemes approximating available actions (or combine just actions in the simpler case), starting from patterns matched by x and going forward. Alternatively, we can go backward starting from the target patterns.

Let us stress a very important fact that approximation of actions can be performed on different levels of generalization. This is possible because the AR schemes used for approximation are hierarchical structures. Thus, by considering patterns from different levels of AR schemes we can obtain approximation of actions in the language of those patterns, and we can talk about actions as well as meta-actions.

21.6 Summary

We discussed some general problems related to reasoning about complex spatiotemporal concepts. We claim that hierarchical approach to reasoning is necessary due to the big gap between low-level information related to sensory data and abstract concepts, very often expressed in a natural language. Domain knowledge given, e.g., in the form of ontology of concepts, can significantly improve the reasoning process.

Information granulation plays an important role in synthesizing more compact and thus more general constructs applicable to broader range of objects. From other hand, approximate reasoning about concepts makes it possible to deal with cases where exact calculations are not feasible. For this purpose rough sets theory comes in handy.

We also discussed the problem of planning for the case when there is available some set of actions. The actions can be used to change state of an object that can be expressed by change of satisfaction degree of some concept. Finding relevant sequence of actions to be undertaken is an important task of the reasoning process.

Acknowledgments

The research has been supported by the Research Center at the Polish-Japanese Institute of Information Technology, Warsaw, Poland, by the grant from Ministry of Science and Higher Education of the Republic of Poland and by the grant Innovative Economy Operational Programme 2007–2013 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland.

References

- [1] J.F. Roddick, K. Hornsby, and M. Spiliopoulou. An updated bibliography of temporal, spatial and spatio-temporal data mining research. In: J.F. Roddick and K. Hornsby (eds), *Post-Workshop Proceedings of the International Workshop on Temporal, Spatial and Spatio-Temporal Data Mining TSDM, Lecture Notes in Artificial Intelligence*, Vol. 2007. Springer-Verlag, Berlin, Germany, 2001, pp. 147–163.
- [2] J.F. Roddick, K. Hornsby, and M. Spiliopoulou. YABTSSTDMR – yet another bibliography of temporal, spatial and spatio-temporal data mining research. In: K.P. Unnikrishnan and R. Uthurusamy (eds), *Proceedings of the SIGKDD Temporal Data Mining Workshop*. ACM, New York, 2001, pp. 167–175.
- [3] Z. Pawlak. Rough sets. *Int. J. Comput. Inf. Sci.* 11 (1982) 341–356.
- [4] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data. D: System Theory, Knowledge Engineering and Problem Solving*, Vol. 9. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.
- [5] L. Polkowski and A. Skowron. Towards adaptive calculus of granules. In: L.A. Zadeh and J. Kacprzyk (eds), *Proceeding of Computing with Words in Information/Intelligent Systems*. Springer-Verlag, Heidelberg, Germany, 1999, pp. 201–227.
- [6] A. Skowron. Toward intelligent systems: Calculi of information granules. *Bull. Int. Rough Set Soc.* 5(1–2) (2001) 9–30.
- [7] L.A. Zadeh. Toward a theory of fuzzy information granulation and its certainty in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.
- [8] L.A. Zadeh. Toward a generalized theory of uncertainty (GTU) – an outline. *Inf. Sci.* 172(1–2) (2005) 1–40.
- [9] A. Skowron and P. Synak. Hierarchical information maps. In: D. Slezak, G. Wang, M.S. Szczuka, I. Düntsch, and Y. Yao (eds), *Proceedings of the Tenth International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing RSFDGrC, Lecture Notes in Computer Science*, Vol. 3641. Springer-Verlag, Heidelberg, Germany, 2005, pp. 622–631.
- [10] A. Skowron and P. Synak. Complex patterns. *Fundam. Inf.* 60(1–4) (2004) 351–366.
- [11] P. Synak, J.G. Bazan, A. Skowron, and J.F. Peters. Spatio-temporal approximate reasoning over complex objects. *Fundam. Inf.* 67(1–3) (2005) 249–269.
- [12] J. Komorowski, L. Polkowski, and A. Skowron. Rough sets: A tutorial. In: S.K. Pal and A. Skowron (eds) *Rough Fuzzy Hybridization: A New Trend in Decision-Making*. Springer-Verlag, Singapore, 1999, pp. 3–98.
- [13] Z. Pawlak. Information systems – theoretical foundations. *Inf. Syst.* 6 (1981) 205–218.
- [14] A. Skowron and P. Synak. Patterns in information maps. In: J.J. Alpigini, J.F. Peters, A. Skowron, and N. Zhong (eds), *Proceedings of the Third International Conference on Rough Sets and Current Trends in Computing RSCTC, Lecture Notes in Artificial Intelligence*, Vol. 2475. Springer-Verlag, Heidelberg, Germany, 2002, pp. 453–460.
- [15] A. Skowron and P. Synak. Reasoning in information maps. *Fundam. Inf.* 59(2–3) (2004) 241–259.
- [16] R. Cooley. *Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data*. Ph.D. Thesis. University of Minnesota, MN, 2000.
- [17] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations* 1(2) (2000) 12–23.

- [18] A.L. Simon and S.L. Shaffer (eds). *Data Warehousing and Business Intelligence for E-Commerce*. Academic Press, San Diego, 2001.
- [19] L. Polkowski and A. Skowron. Rough mereology: A new paradigm for approximate reasoning. *Int. J. Approx. Reason.* 15(4) (1996) 333–365.
- [20] L. Polkowski and A. Skowron. Rough mereology in information systems. A case study: Qualitative spatial reasoning. In: L. Polkowski, T.Y. Lin, and S. Tsumoto (eds), *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems, Studies in Fuzziness and Soft Computing*, Vol. 56. Springer-Verlag/Physica-Verlag, Heidelberg, Germany, 2000, chapter 3, pp. 89–135.
- [21] J.G. Bazan and A. Skowron. Classifiers based on approximate reasoning schemes. In: B. Dunin-Keplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds), *Proceedings on Monitoring, Security, and Rescue Tasks in Multiagent Systems MSRAS, Advances in Soft Computing*. Combridge University Press, Cambridge, UK, 2005, pp. 191–202.
- [22] L. Polkowski and A. Skowron. Rough mereological approach to knowledge-based distributed AI. In: J.K. Lee, J. Liebowitz, and J.M. Chae (eds), *Proceedings of the Third World Congress on Expert Systems*. Cognizant Communication Corporation, New York, 1996, pp. 774–781.
- [23] A. Skowron and J. Stepaniuk. Information granules and rough-neural computing. In: S.K. Pal, L. Polkowski, and A. Skowron (eds), *Rough-Neural Computing: Techniques for Computing with Words, Cognitive Technologies*. Springer-Verlag, Heidelberg, Germany, 2004, pp. 43–84.
- [24] J. Stepaniuk. Approximation spaces, reducts and representatives. In: L. Polkowski and A. Skowron (eds). *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems, Studies in Fuzziness and Soft Computing*, Vol. 19. Physica-Verlag, Heidelberg, Germany, chapter 6, 1998, pp. 109–126.
- [25] J.H. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, Heidelberg, Germany, 2001.
- [26] W. Kloesgen and J. Źytkow (eds). *Handbook of Knowledge Discovery and Data Mining*. Oxford University Press, Oxford, 2002.
- [27] T.M. Mitchell. *Machine Learning*. Mc-Graw Hill, Columbus, USA, 1997.
- [28] L. Breiman. Statistical modeling: The two cultures. *Stat. Sci.* 16(3) (2001) 199–231.
- [29] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.
- [30] L.W. Barsalou. Perceptual symbol systems. *Behav. Brain Sci.* 22 (1999) 577–660.
- [31] M. Fahle and T. Poggio. *Perceptual Learning*. MIT Press, MA, 2002.
- [32] S. Harnad. *Categorical Perception: The Groundwork of Cognition*. Cambridge University Press, New York, 1987.
- [33] L.A. Zadeh. A new direction in AI: Toward a computational theory of perceptions. *AI Mag.* 22(1) (2001) 73–84.
- [34] E. Sandewall (ed.). *Features and Fluents: The Representation of Knowledge about Dynamical Systems*, Vol. 1. Oxford University Press, Oxford, UK, 1994.
- [35] A. Skowron and P. Synak. Complex patterns in spatio-temporal reasoning. In: L. Czaja (ed.). *Proceedings of Concurrency Specification and Programming CSP*, Vol. 2, Czarna, Poland, 2003, pp. 487–499.
- [36] WITAS, 2008. Project Web site. <http://www.ida.liu.se/ext/witas/>.
- [37] J.F. Peters, A. Skowron, P. Synak, and S. Ramanna. Rough sets and information granulation. In: T. Bilgic, D. Baets, and O. Kaynak (eds), *Proceeding of the Tenth International Fuzzy Systems Association World Congress IFSA, Lecture Notes in Artificial Intelligence*, Vol. 2715. Springer-Verlag, Heidelberg, Germany, 2003, pp. 370–377.
- [38] S.K. Pal and A. Skowron (eds). *Rough Fuzzy Hybridization: A New Trend in Decision-Making*. Springer-Verlag, Singapore, 1999.
- [39] L. Polkowski and A. Skowron (eds). *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems, Studies in Fuzziness and Soft Computing*, Vol. 19. Physica-Verlag, Heidelberg, Germany, 1998.
- [40] R. Słowiński, S. Greco, and B. Matarazzo. Rough set analysis of preference-ordered data. In: J.J. Alpigini, J.F. Peters, A. Skowron, and N. Zhong (eds), *Proceedings of the Third International Conference on Rough Sets and Current Trends in Computing RSCTC, Lecture Notes in Artificial Intelligence*, Vol. 2475. Springer-Verlag, Heidelberg, Germany, 2002, pp. 44–59.
- [41] A. Skowron and J. Stepaniuk. Tolerance approximation spaces. *Fundam. Inf.* 27(2–3) (1996) 245–253.
- [42] S. Leśniewski. Grundzüge eines neuen systems der grundlagen der mathematik. *Fundam. Math.* 14 (1929) 1–81.
- [43] A.A. Tzacheva and Z.W. Raš. Action rules mining. *Int. J. Intell. Syst.* 20(7) (2005) 719–736.
- [44] S. Greco, B. Matarazzo, N. Pappalardo, and R. Słowiński. Measuring expected effects of interventions based on decision rules. *J. Exp. Theor. Artif. Intell.* 17(1–2) (2005) 103–118.

Part Two

Hybrid Methods and Models of Granular Computing

22

A Survey of Interval-Valued Fuzzy Sets

Humberto Bustince, Javier Montero, Miguel Pagola, Edurne Barrenechea,
and Daniel Gomez

22.1 Introduction

Zadeh presented the theory of fuzzy sets in 1965 [1]. From the beginning it was clear that this theory was an extraordinary tool for representing human knowledge. Nevertheless, Zadeh himself established in 1973 (see [2]) that sometimes, in decision-making processes, knowledge is better represented by means of some generalizations of fuzzy sets. The so-called extensions of fuzzy set theory arise in this way.

In the applied field, the success of the use of fuzzy set theory depends on the choice of the membership function that we make. However, there are applications in which experts do not have precise knowledge of the function that should be taken. In these cases, it is appropriate to represent the membership degree of each element to the fuzzy set by means of an interval. From these considerations arises the extension of fuzzy sets called *theory of interval-valued fuzzy sets* (IVFSs), that is, fuzzy sets such that the membership degree of each element of the fuzzy set is given by a closed subinterval of the interval $[0, 1]$. Hence, not only vagueness (lack of sharp class boundaries) but also a feature of uncertainty (lack of information) can be addressed intuitively.

Therefore (see [3]), membership functions of IVFSs are not as specific as their counterparts of fuzzy sets, but this lack of specificity makes them more realistic in some applications. Their advantage is that they allow us to express our uncertainty in identifying a particular membership function. This uncertainty is involved when IVFSs are processed, making results of the processing less specific but more credible.

These sets were born in the 1970s. In May 1975 Sambuc (see [4]) presented in his doctoral thesis the concept of an IVFS named a Φ -fuzzy set. That same year, Jahn [5] wrote about these sets and Zadeh [6] discussed the representation of type 2 fuzzy sets and its potential in approximate reasoning. One year later, Grattan-Guinness [7] established a definition of an interval-valued membership function. In that decade, IVFSs appeared in the literature in various guises and it was not until the 1980s, with the work of Gorzalczany and Turksen [8–16], that the importance of these sets, as well as their name, was definitely established.

In this chapter, we present a survey of IVFSs. We describe the most important concepts of this theory and provide a set of references that is intended to represent as best as possible the work carried out on this extension.

In this introduction, we must indicate the three major problems, in our opinion, of the theory that is the object of our study:

- A large number of contributions are generalized adaptations of the theoretical developments of fuzzy set theory. This prevents us from focusing on the nature of IVFSs themselves and studying the properties possessed exclusively by these sets.
- In the early work on this theory, Gorzalczany presented the concept of the degree of compatibility between two IVFSs as an interval (see [9]). In some of the latest publications that we have read on interval-valued measures of information, these measures are defined as a point on $[0, 1]$. With this modelization, what is achieved is that the relation between different interval-valued measures of information is a copy of the relation that exists between these concepts in fuzzy set theory. However, we consider that in this case we lose the information that would be provided by a modelization of these measures using intervals.
- Currently, there are two names for these sets: some authors call them *interval-valued fuzzy sets* and others *interval type 2 fuzzy sets*. In [17] Mendel writes, ‘it turns out that an interval type 2 fuzzy set is the same as an interval-valued fuzzy set for which there is a very extensive literature. These two seemingly different kinds of fuzzy sets were historically approached from very different starting points, which as we shall explain next has turned out to be a very good thing’. Nonetheless, we consider that this duplicity in the name can cause confusion. We have observed that in some papers results that have already been known for many years for IVFSs are presented for interval type 2 fuzzy sets. For this reason, we believe the name should be set; otherwise, the complete bibliography that exists on IVFSs should be taken into account.

Other objections to these sets can be found in [18].

This chapter is organized in the following way. Starting from the definition of an IVFSs and from the study of two construction methods, we present in Section 22.2 the relation between these sets and other representations of fuzzy set theory. Next, in Sections 22.3 and 22.4, we study the connectives and possible combinations between them. In Section 22.5, we analyze the laws for conjunctions and disjunctions. Then, in Section 22.6, we give an interpretation of interval-valued fuzzy operators together with a construction method. In Section 22.7, we recall information measures, and in Section 22.8, we mention the main fields of application of IVFSs. In the final section (Section 22.9), we analyze the use of IVFSs in granular computing.

22.2 Preliminary Definitions

In fuzzy set theory, a function $n : [0, 1] \rightarrow [0, 1]$ such that $n(0) = 1, n(1) = 0$ that is decreasing is called *negation*. If a negation is strictly decreasing and continuous, it is called a strict negation. If $n(n(x)) = x$ for all $x \in [0, 1]$, then n is involutive. A *strong negation* is a negation that is strict and involutive. In 1979, Trillas [19] characterized strong negations using automorphisms (see also [20]).

In this chapter we shall denote by $\mathcal{F}Ss(U)$ the set of all fuzzy sets defined on a finite referential U , where U is a non-empty set. Given a fuzzy set $A = \{(u, \mu_A(u))|u \in U\} \in \mathcal{F}Ss(U)$, the expression $A_n = \{(u, n(\mu_A(u)))|u \in U\}$ will be used here as the complement of the fuzzy set A , where n is a negation. We must point out that two distinct notations are most commonly employed in the literature to denote membership functions. In one of them, the membership function of a fuzzy set A is denoted in the way indicated above by the symbol μ_A ; that is, $\mu_A : U \rightarrow [0, 1]$. In the other one, the function is denoted by A and has, of course, the same form: $A : U \rightarrow [0, 1]$.

We denote by $L([0, 1])$ the set of all closed subintervals of the closed interval $[0, 1]$; that is,

$$L([0, 1]) = \{\mathbf{x} = [\underline{x}, \bar{x}] | (\underline{x}, \bar{x}) \in [0, 1]^2 \text{ and } \underline{x} \leq \bar{x}\}.$$

$L([0, 1])$ is a partially ordered set with respect to the relation \leq_L defined in the following way: given $\mathbf{x}, \mathbf{y} \in L([0, 1])$,

$$\mathbf{x} \leq_L \mathbf{y} \text{ if and only if } \underline{x} \leq \underline{y} \text{ and } \bar{x} \leq \bar{y}.$$

The relation above is transitive and antisymmetric and it expresses the fact that \mathbf{x} links strongly to \mathbf{y} , so that $(L([0, 1]), \leq_L)$ is a complete lattice (see [21–23]), where the smallest element is $0_L = [0, 0]$ and the largest is $1_L = [1, 1]$. Evidently, it is not a linear lattice, for there exist elements that are not comparable. A very interesting study on the arithmetic operations in $L([0, 1])$ can be found in [24–28].

Definition 1. An interval-valued fuzzy set A on the universe $U \neq \emptyset$ is a mapping $A : U \rightarrow L([0, 1])$.

Obviously, $A(u) = [\underline{A}(u), \bar{A}(u)] \in L([0, 1])$ is the membership degree of $u \in U$. We denote by $\text{IVFSs}(U)$ the set of IVFSs on U . We denote by W the length of the interval considered.

In [29], two representation theorems and an equivalent classification theorem for IVFSs are presented. Besides the relation \leq_L above, other relations (see [30]) on IVFSs have been studied, including among others the following: $\mathbf{x} \leq \mathbf{y}$ if and only if $\underline{x} \leq \underline{y}$ and $\bar{y} \leq \bar{x}$.

22.2.1 Relation to Other Extensions

The concept of a *type 2 fuzzy set* was introduced in 1975 by Zadeh [6] as a generalization of an ordinary fuzzy set. Type 2 fuzzy sets are characterized by a fuzzy membership function, that is, the membership value for each element of the set is itself a fuzzy set in $[0, 1]$.

Formally, given the referential set U , a type 2 fuzzy set is defined as an object \bar{A} which has the following form:

$$\bar{A} = \{(u, x, \mu_u(x)) | u \in U, x \in [0, 1]\},$$

where $x \in [0, 1]$ is the primary membership degree of u and $\mu_u(x)$ is the secondary membership level, specific to a given pair (u, x) .

IVFSs can be generalized by assigning to each interval a fuzzy set defined on the referential set $[0, 1]$; that is, IVFSs can be generalized by means of type 2 fuzzy sets. The following equation shows a way of constructing a type 2 fuzzy set from an $A \in \text{IVFSs}(U)$:

$$\mu_u(x) = \begin{cases} 0 & \text{if } 0 \leq x \leq \underline{A}(u) \\ \frac{2}{\bar{A}(u)-\underline{A}(u)}(x - \underline{A}(u)) & \text{if } \underline{A}(u) \leq x \leq \frac{\underline{A}(u)+\bar{A}(u)}{2} \\ \frac{2}{\bar{A}(u)-\underline{A}(u)}(x - \bar{A}(u)) & \text{if } \frac{\underline{A}(u)+\bar{A}(u)}{2} \leq x \leq \bar{A}(u) \\ 0 & \text{if } \bar{A}(u) \leq x \leq 1. \end{cases}$$

Evidently, the choice of a triangular shape for the membership function is totally arbitrary; we can associate another type of membership function, for example, a trapezoidal shape, with the set.

A particular case of a type 2 fuzzy set is an *interval type 2 fuzzy set* (see [17, 31–33]). An interval type 2 fuzzy set \bar{A} in U is defined by

$$\bar{A} = \{(u, A(u), \mu_u(x)) | u \in U, A(u) \in L([0, 1])\},$$

where $A(u)$ is the membership function presented in Definition 1; that is, it is a closed subinterval of $[0, 1]$, and the function $\mu_u(x)$ represents the fuzzy set associated with the element $u \in U$ obtained when x covers the interval $[0, 1]$; $\mu_u(x)$ is given in the following way:

$$\mu_u(x) = \begin{cases} a & \text{if } \underline{A}(u) \leq x \leq \bar{A}(u) \\ 0 & \text{otherwise,} \end{cases}$$

where $0 \leq a \leq 1$. In [17, 31–33], it turns out that an interval type 2 fuzzy set is the same as an IVFS if we take $a = 1$.

Another important extension of fuzzy set theory is the theory of Atanassov's intuitionistic fuzzy sets (A-IFSs) [34, 35]. A-IFSs assign to each element of the universe not only a membership degree, but also a non-membership degree, which is less than or equal to 1 minus the membership degree.

An A-IFS on U is a set

$$A = \{(u, \mu_A(u), v_A(u)) | u \in U\},$$

where $\mu_A(u) \in [0, 1]$ denotes the membership degree and $v_A(u) \in [0, 1]$ the non-membership degree of u in A and where, for all $u \in U$, $\mu_A(u) + v_A(u) \leq 1$.

In [34], Atanassov established that every Atanassov's intuitionistic fuzzy set A on U can be represented by an interval-valued fuzzy set A given by

$$\begin{aligned} A : U &\rightarrow L([0, 1]) \\ u &\mapsto [\mu_A(u), 1 - v_A(u)], \quad \text{for all } u \in U. \end{aligned}$$

Using this representation, Atanassov proposed in 1983 that A-IFS theory was equivalent to the theory of IVFSs. This equivalence was proved in 2003 by Deschrijver and Kerre [23]. Therefore, from a mathematical point of view, the results that we obtain for IVFSs are easily adaptable to A-IFSs and vice versa. Nevertheless, we need to point out that, conceptually, the two types of sets are totally different. This is made clear when applications of these sets are constructed (see [36–39]).

In 1993, Gau and Buehrer introduced the concept of *vague sets* [40]. Later, in 1996, it was proved that vague sets are in fact A-IFSs [41].

A compilation of the sets that are equivalent (from a mathematical point of view) to IVFSs can be found in [42]. Two conclusions are drawn from this study:

1. IVFSs are equivalent to A-IFSs (and therefore vague sets), to grey sets (see [43]), and to L -fuzzy set in Goguen's sense (see [44, 45]) with respect to a special lattice $L([0, 1])$.
2. IVFSs are a particular case of probabilistic sets (see [46]), of soft sets (see [47]), of Atanassov's interval-valued intuitionistic fuzzy sets (see [35]) and evidently of type 2 fuzzy sets.

22.2.2 Some Methods of Construction of Interval-Valued Fuzzy Sets

In 1975 Sambuc used IVFSs for the construction of a computer system that would help make a diagnosis of certain thyroidal pathologies. In this paper, the problem of constructing an interval-valued membership function that is more appropriate for the computer system arises for the first time.

Later, Turksen and Yao (see [16]) proposed a method of constructing IVFSs using the fact that in fuzzy logic it generally occurs that the conjunctive normal form (CNF) gives membership grades greater than the disjunctive normal form (DNF). Ten years later, Turksen (see [13, 48]) proved that when performing a fuzzy set-theoretic aggregation of regular membership functions, the simultaneous use of CNF and DNF forms of the fuzzy connectives yields an IVFSs.

The constructions of IVFSs presented in 1996 (see [49]) are divided into two groups: constructions from a fuzzy set and constructions from two or more fuzzy sets. Next we present an example of each of them.

Construction of an IVFSs from a Fuzzy Set

In 2005 Tizhoosh (see [50]) used IVFSs for determining the threshold of an image Q . The threshold enables to separate the object contained in the image from its background.

In his work, Tizhoosh associates each image with an IVFSs in the following way:

1. Have an expert assign the image a fuzzy set characterized by the membership function μ_Q .
2. For each pixel, construct the interval $[(\mu_Q)^\alpha, (\mu_Q)^{\frac{1}{\alpha}}]$ with $\alpha \in (1, \infty)$, which represents its membership to the IVFS that is going to represent the image (see Figure 22.1).

The relevance of this construction has been proved in an experimental way in [50–52]. In these works one concludes that in images with many pixels for which experts do not agree on whether they belong

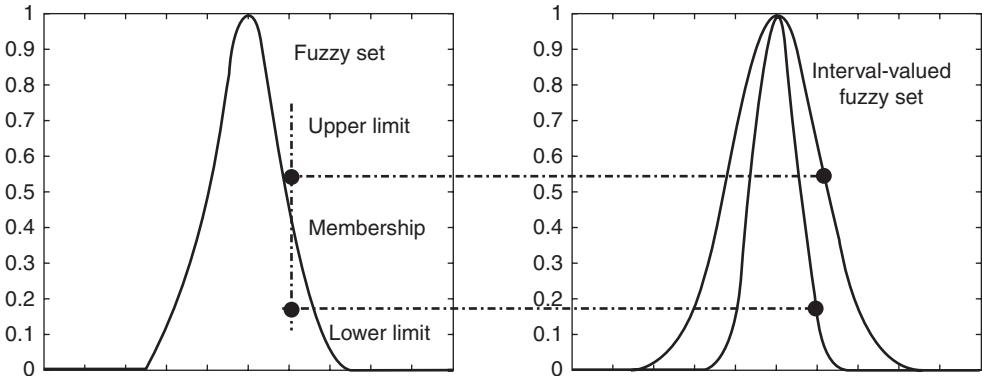


Figure 22.1 Method of construction of an IVFS from a fuzzy set

to the background or to the object, the thresholds obtained with these constructions (together with the algorithm that we will describe in Section 22.8) are much better than those obtained with classical methods or with methods that use fuzzy techniques.

A generalization of this method can be found developed in [53].

Construction of an IVFSs from Two or More Fuzzy Sets

If we ask several experts to construct the membership function that represents the fuzzy set that modelizes a certain action, we discover the fact that in most cases the experts choose different membership functions. Because of this we have uncertainty in choosing the best function. In these conditions it is recommended to work with IVFSs constructed in the following way: each element is assigned an interval whose lower extreme is the lowest value given by the experts for that element and as upper extreme the highest (see [54–57]).

Other construction methods can be found in [38, 58, 59].

22.2.3 A Method of Construction of Fuzzy Sets from Interval-Valued Fuzzy Sets

The operator that we present next will enable us to construct a family of fuzzy sets from an IVFS (see [60]). This concept will be frequently used throughout the whole chapter.

Definition 2. Let $\alpha \in [0, 1]$; we define K_α as a function $K_\alpha : L([0, 1]) \rightarrow [0, 1]$ such that it satisfies the following conditions:

1. If $\underline{x} = \bar{x}$, then $K_\alpha(\underline{x}) = \underline{x}$.
2. $K_0(\underline{x}) = \underline{x}$, $K_1(\bar{x}) = \bar{x}$ for all $\underline{x}, \bar{x} \in L([0, 1])$.
3. If $\underline{x} \leq_L \bar{x}$, with $\underline{x}, \bar{x} \in L([0, 1])$, then $K_\alpha(\underline{x}) \leq K_\alpha(\bar{x})$ for all $\alpha \in [0, 1]$.
4. $K_\alpha(\underline{x}) \leq K_\beta(\bar{x})$ if and only if $\alpha \leq \beta$, for all $\underline{x}, \bar{x} \in L([0, 1])$, with $\beta \in [0, 1]$.

A study of these operators can be found in [35] (and also in [49, 61]). In these papers, the following expressions are proposed for these operators:

$$K_\alpha(\underline{x}) = K_\alpha([\underline{x}, \bar{x}]) = K_\alpha([K_0(\underline{x}), K_1(\bar{x})]) = \underline{x} + \alpha(\bar{x} - \underline{x}) = K_0(\underline{x}) + \alpha W_{\underline{x}},$$

where $W_{\underline{x}}$ represents the length of the interval \underline{x} .

The operator K_α enables every IVFS to be associated with a fuzzy set in the following way (see [21, 35, 41, 49, 53, 54, 61–66]):

$$K_\alpha : \mathcal{IVFSs}(U) \rightarrow \mathcal{FSs}(U)$$

given by

$$K_\alpha(A) = \{(u, \mu_{K_\alpha(A)}(u) = K_\alpha(A(u)) = K_\alpha([\underline{A}(u), \bar{A}(u)])|u \in U\}.$$

Unless otherwise indicated, the operator K_α that we shall use is the general operator presented in Definition 2. That is, we shall not use any particular expression for it.

22.3 Connectives

In various papers (e.g., [4, 9, 21, 35, 41, 47, 49, 51–54, 60–66, 68–82]), the union, intersection, and complementation of $\mathcal{IVFSs}(U)$ are defined in the following way: if $A, B \in \mathcal{IVFSs}(U)$, then

$$\begin{aligned} A \cap B(u) &= [\min(K_0(A(u)), K_0(B(u))), \min(K_1(A(u)), K_1(B(u)))], \\ A \cup B(u) &= [\max(K_0(A(u)), K_0(B(u))), \max(K_1(A(u)), K_1(B(u)))], \\ A_n(u) &= [1 - K_1(A(u)), 1 - K_0(A(u))] \text{ for all } u \in U. \end{aligned}$$

In [21, 64, 65], it is proved that $\{\mathcal{IVFSs}, \cap, \cup\}$ is a distributive, bounded, non-complemented lattice that satisfies De Morgan's laws (with respect to the definition of A_n above).

Next, in the first subsection we thoroughly study the concept of interval-valued negation and in subsequent subsections we describe the manner of modelizing the operations of union and intersection by means t -norms and t -conorms.

22.3.1 IV Negations

Interval-valued negations, hereinafter referred to as IV negations, are an extension of negations and are defined as follows:

Definition 3. An IV negation is a function $N : L([0, 1]) \rightarrow L([0, 1])$ that is decreasing (with respect to \leq_L) such that $N(1_L) = 0_L$ and $N(0_L) = 1_L$. If for all $\mathbf{x} \in L([0, 1])$, $N(N(\mathbf{x})) = \mathbf{x}$, it is said that N is involutive.

Next, we present a theorem related to the construction of IV negations. Historically, this theorem provided the first construction method for such negations (see [21, 53]).

Theorem 1. Let the function $N : L([0, 1]) \rightarrow L([0, 1])$ be given by

$$N(\mathbf{x}) = [n(K_1(\mathbf{x})), n(K_0(\mathbf{x}))],$$

where $n : [0, 1] \rightarrow [0, 1]$ is a negation. Under these conditions N is an IV negation.

An in-depth study of strict IV negations can be found in [54, 77].

Representation Theorem for IV Negations

A representation theorem for IV negations was obtained in [54]. Previously Deschrijver *et al.* (see [77]) proved this theorem for A-IFSs. The only small difference between these two studies lies essentially in the fact that for the negations of the A-IFSs, K_α operators are not used, whereas these operators are used

in the characterization of the IV negations developed in [54]. This fact enables us to prove the following lemma (which is a consequence of Lemma 3.5 proved in [77]):

Lemma 1. *Let there be a K_α with $\alpha \in [0, 1]$. If N is an involutive IV negation, then for every $\mathbf{x} = [K_0(\mathbf{x}), K_1(\mathbf{x})] \in L([0, 1])$,*

$$K_0(N(\mathbf{x})) = K_\alpha(N(\mathbf{x})) = K_1(N(\mathbf{x}))$$

holds for all $\alpha \in [0, 1]$.

Lemma 1, together with other properties (see [21, 49, 54, 61]) of the operators K_α , has made it possible to prove the following characterization theorem (which is an adaptation for IVFSs of Theorem 3.6 proved in [77]).

Theorem 2. *A function $N : L([0, 1]) \rightarrow L([0, 1])$ is an involutive IV negation if and only if there exists an involutive negation n such that*

$$N(\mathbf{x}) = [n(K_1(\mathbf{x})), n(K_0(\mathbf{x}))].$$

Complement of an Interval-Valued Fuzzy Set

Given an interval-valued fuzzy set $A \in \mathcal{IVFS}(U)$, from the functions N we can define the concept of the *complement* of A in the following way:

$$A_N(u) = N(A(u)) \text{ for all } u \in U.$$

22.3.2 IV t-Norms and IV t-Conorms

In fuzzy set theory, *t*-norms are used for modeling the intersection (or conjunction) of two fuzzy sets and *t*-conorms for modeling the union (or disjunction). Similarly, in the theory of IVFSs, we can model the intersection and the union using interval-valued *t*-norms and interval-valued *t*-conorms in the following way: for all $u \in U$ and $A, B \in \mathcal{IVFS}(U)$, we have

$$\begin{aligned} A \cap_T B(u) &= \mathbf{T}(A(u), B(u)), \\ A \cup_S B(u) &= \mathbf{S}(A(u), B(u)), \end{aligned}$$

where **T** and **S** are, according to Definition 4, an IV *t*-norm and an IV *t*-conorm, respectively (see [60, 65, 74]).

Definition 4. *A function $\mathbf{T} : (L([0, 1]))^2 \rightarrow L([0, 1])$ is said to be an interval-valued *t*-norm (IV *t*-norm) if it is commutative, associative, and increasing (in both arguments with respect to the order \leq_L) and has a neutral element $1_L = [1, 1]$. In the same way, a function $\mathbf{S} : (L([0, 1]))^2 \rightarrow L([0, 1])$ is said to be an interval-valued *t*-conorm (IV *t*-conorm) if it is commutative, associative, and increasing and has a neutral element $0_L = [0, 0]$.*

Evidently, $\mathbf{T}(\mathbf{x}, 0_L) \leq_L \mathbf{T}(1_L, 0_L) = 0_L$, and therefore $\mathbf{T}(\mathbf{x}, 0_L) = 0_L$. In a similar way, we have $\mathbf{S}(\mathbf{x}, 1_L) = 1_L$. Note that Definition 4 is an extension of the classical definition of the *t*-norm and *t*-conorm in $[0, 1]$. We only need to substitute $L([0, 1])$ for $[0, 1]$.

A lot has been written on the way in which IV *t*-norms and IV *t*-conorms can be generated from *t*-norms and *t*-conorms in $[0, 1]$. The first idea that appeared was the generation of IV *t*-norms (and IV *t*-conorms) from two *t*-norms T_a , and T_b (and S_a , and S_b) in $[0, 1]$, from the operator K_α and from expressions in the following way (see [60, 65]):

$$\begin{cases} \mathbf{T}(\mathbf{x}, \mathbf{y}) = [T_a(K_\alpha(\mathbf{x}), K_\alpha(\mathbf{y})), T_b(K_\beta(\mathbf{x}), K_\beta(\mathbf{y}))] \\ \mathbf{S}(\mathbf{x}, \mathbf{y}) = [S_a(K_\alpha(\mathbf{x}), K_\alpha(\mathbf{y})), S_b(K_\beta(\mathbf{x}), K_\beta(\mathbf{y}))]. \end{cases} \quad (1)$$

A justification for the choice of these expressions and a demonstration of the following theorem can be found in [54].

Theorem 3. (a) Let $\alpha, \beta \in [0, 1]$ such that $\alpha < \beta$ and let T_a and T_b be two t-norms in $[0, 1]$ such that $T_a \leq T_b$. Let the function $\mathbf{T} : (L([0, 1]))^2 \rightarrow L([0, 1])$ be given by

$$\mathbf{T}(\mathbf{x}, \mathbf{y}) = [T_a(K_\alpha(\mathbf{x}), K_\alpha(\mathbf{y})), T_b(K_\beta(\mathbf{x}), K_\beta(\mathbf{y}))] \text{ for all } \mathbf{x}, \mathbf{y} \in L([0, 1]).$$

Under these conditions,

$$\mathbf{T} \text{ is an IV t-norm if and only if } \begin{cases} K_\alpha(\mathbf{x}) = K_0(\mathbf{x}) \\ K_\beta(\mathbf{x}) = K_1(\mathbf{x}) \end{cases}$$

for all $\mathbf{x} \in L([0, 1])$.

(b) Let $\alpha, \beta \in [0, 1]$ such that $\alpha < \beta$ and let S_a and S_b be two t-conorms in $[0, 1]$ such that $S_a \leq S_b$. Let the function $\mathbf{S} : (L([0, 1]))^2 \rightarrow L([0, 1])$ be given by

$$\mathbf{S}(\mathbf{x}, \mathbf{y}) = [S_a(K_\alpha(\mathbf{x}), K_\alpha(\mathbf{y})), S_b(K_\beta(\mathbf{x}), K_\beta(\mathbf{y}))] \text{ for all } \mathbf{x}, \mathbf{y} \in L([0, 1]).$$

Under these conditions,

$$\mathbf{S} \text{ is an IV t-conorm if and only if } \begin{cases} K_\alpha(\mathbf{x}) = K_0(\mathbf{x}) \\ K_\beta(\mathbf{x}) = K_1(\mathbf{x}) \end{cases}$$

for all $\mathbf{x} \in L([0, 1])$.

A consequence of Theorem 3 is the following corollary (see [22, 23, 43, 51, 52, 60, 64–66, 68–77]). This corollary provides a construction method for IV t-norms and IV t-conorms, which was presented in the first papers on this topic.

Corollary 1. (a) If T_a and T_b are two t-norms in $[0, 1]$ such that $T_a(x, y) \leq T_b(x, y)$ for all $x, y \in [0, 1]$, then the function $\mathbf{T} : (L([0, 1]))^2 \rightarrow L([0, 1])$ defined for each $\mathbf{x}, \mathbf{y} \in L([0, 1])$ by

$$\mathbf{T}(\mathbf{x}, \mathbf{y}) = [T_a(K_0(\mathbf{x}), K_0(\mathbf{y})), T_b(K_1(\mathbf{x}), K_1(\mathbf{y}))]$$

is an IV t-norm.

(b) If S_a and S_b are two t-conorms in $[0, 1]$ such that $S_a(x, y) \leq S_b(x, y)$ for all $x, y \in [0, 1]$, then the function $\mathbf{S} : (L([0, 1]))^2 \rightarrow L([0, 1])$ defined for each $\mathbf{x}, \mathbf{y} \in L([0, 1])$ by

$$\mathbf{S}(\mathbf{x}, \mathbf{y}) = [S_a(K_0(\mathbf{x}), K_0(\mathbf{y})), S_b(K_1(\mathbf{x}), K_1(\mathbf{y}))]$$

is an IV t-conorm.

The following theorem, proved in [54], makes it clear that the converse of Corollary 1 is not true: that is, there are IV t-norms (and IV t-conorms) that are not generated from expressions of the type (1).

Theorem 4. (a) Let there be an operator K_α with $\alpha \in [0, 1]$ and let T be any t-norm in $[0, 1]$. Let there be a function

$$\mathbf{T}_\alpha^T(\mathbf{x}, \mathbf{y}) = \begin{cases} \mathbf{x} \text{ if } \mathbf{y} = 1_L \\ \mathbf{y} \text{ if } \mathbf{x} = 1_L \\ [T(K_\alpha(\mathbf{x}), K_\alpha(\mathbf{y})), T(K_\alpha(\mathbf{x}), K_\alpha(\mathbf{y}))]. \end{cases}$$

Under these conditions,

$$\mathbf{T}_\alpha^T \text{ is an IV t-norm if and only if } K_\alpha(\mathbf{x}) = K_0(\mathbf{x}) \text{ for all } \mathbf{x} \in L([0, 1]).$$

(b) Let there be an operator K_α with $\alpha \in [0, 1]$ and let S be any t-conorm in $[0, 1]$. Let there be a function

$$\mathbf{S}_\alpha^S(\mathbf{x}, \mathbf{y}) = \begin{cases} \mathbf{x} \text{ if } \mathbf{y} = 0_L \\ \mathbf{y} \text{ if } \mathbf{x} = 0_L \\ [S(K_\alpha(\mathbf{x}), K_\alpha(\mathbf{y})), S(K_\alpha(\mathbf{x}), K_\alpha(\mathbf{y}))]. \end{cases}$$

Under these conditions,

$$\mathbf{S}_\alpha^S \text{ is an IV } t\text{-conorm if and only if } K_\alpha(\mathbf{x}) = K_1(\mathbf{x}) \text{ for all } \mathbf{x} \in L([0, 1]).$$

From now on, we shall denote \mathbf{T}_0^T and \mathbf{S}_1^S by \mathbf{T}^T and \mathbf{S}^S , respectively.

t-Representable IV t-norms and s-Representable IV t-Conorms

The objective of constructing characterization theorems for IV t -norms (and IV t -conorms) led Cornelis *et al.* [74] and Deschrijver *et al.* [77] to introduce the concepts of t -representable IV t -norm and s -representable IV t -conorm.

Definition 5. (a) An IV t -norm is said to be t -representable if there are two t -norms T_a and T_b in $[0, 1]$ such that $\mathbf{T}(\mathbf{x}, \mathbf{y}) = [T_a(K_0(\mathbf{x}), K_0(\mathbf{y})), T_b(K_1(\mathbf{x}), K_1(\mathbf{y}))]$ for all $\mathbf{x} \in L([0, 1])$ and for all $\mathbf{y} \in L([0, 1])$, such that $\mathbf{x} = [K_0(\mathbf{x}), K_1(\mathbf{x})]$, $\mathbf{y} = [K_0(\mathbf{y}), K_1(\mathbf{y})]$.

(b) An IV t -conorm is said to be s -representable if there are two t -conorms S_a and S_b in $[0, 1]$ such that $\mathbf{S}(\mathbf{x}, \mathbf{y}) = [S_a(K_0(\mathbf{x}), K_0(\mathbf{y})), S_b(K_1(\mathbf{x}), K_1(\mathbf{y}))]$ for all $\mathbf{x} \in L([0, 1])$ and for all $\mathbf{y} \in L([0, 1])$, such that $\mathbf{x} = [K_0(\mathbf{x}), K_1(\mathbf{x})]$, $\mathbf{y} = [K_0(\mathbf{y}), K_1(\mathbf{y})]$.

We denote by \mathbf{T}_{T_a, T_b} the t -representable IV t -norms obtained by means of the t -norms T_a and T_b in $[0, 1]$. We denote similarly the s -representable IV t -conorms \mathbf{S}_{S_a, S_b} .

Some examples of t -representable IV t -norms and s -representable IV t -conorms are the ones studied in Theorem 3. Some examples of non- t -representable IV t -norms and non- s -representable IV t -conorms are the ones studied in Theorem 4 (see [77]).

In [23, 43, 72–77], it is proved that some important representation theorems can be also shown for IV t -norms, but not for t -representable t -norms. This shows that IVFS theory cannot be reduced to an approach in which all operators are t -representable. In any case, the theoretical developments described in these publications are similar (although in a certain way they generalize them) to the theoretical developments carried out in [3, 20, 82] in order to characterize t -norms (and t -conorms) in $[0, 1]$. In [79], the concept of the interval-valued uninorm was analyzed for the first time.

The Archimedean Property of IV t-Norms

Starting from the work on t -norms in $[0, 1]$ of Klement *et al.* (see [83–86]), Deschrijver adapted the Archimedean property of t -norms in $[0, 1]$ to IV t -norms in [78]. In that publication, new and important concepts of weak Archimedean and strong Archimedean properties were presented, and the definition of the *pseudo-t-representable IV t-norm* was introduced in the following way:

An IV t -norm is called *pseudo-t-representable* if there exists a t -norm T on $([0, 1], \leq)$ such that, for all $\mathbf{x}, \mathbf{y} \in L([0, 1])$,

$$T(\mathbf{x}, \mathbf{y}) = [T(K_0(\mathbf{x}), K_0(\mathbf{y})), \max(T(K_0(\mathbf{x}), K_1(\mathbf{y})), T(K_1(\mathbf{x}), K_0(\mathbf{y})))].$$

In [78], there is a study of the conditions under which t -representable IV t -norms and pseudo- t -representable IV t -norms satisfy the Archimedean property, the weak Archimedean property, or the strong Archimedean property. Particularly it is proved that the pseudo- t -representable IV t -norms satisfy the Archimedean property but the t -representable IV t -norms do not. This also justifies the comment above regarding the fact that IVFS theory cannot be reduced to an approach in which all operators are t -representable.

22.4 Combinations of Operations

Theorem 5. Let N be any involutive IV negation. The following items hold:

(a) If \mathbf{T} is an IV t -norm, then the function defined by

$$\mathbf{S}^*(\mathbf{x}, \mathbf{y}) = N(\mathbf{T}(N(\mathbf{x}), N(\mathbf{y}))) \text{ for all } \mathbf{x}, \mathbf{y} \in L([0, 1]) \text{ is an IV } t\text{-conorm.}$$

- (b) If \mathbf{T} is t -representable, then \mathbf{S}^* is s -representable.
- (c) If \mathbf{S} is an IV t -conorm, then the function defined by

$$\mathbf{T}^*(\mathbf{x}, \mathbf{y}) = N(\mathbf{S}(N(\mathbf{x}), N(\mathbf{y}))) \text{ for all } \mathbf{x}, \mathbf{y} \in L([0, 1])$$
is an IV t -norm.
- (d) If \mathbf{S} is s -representable, then \mathbf{T}^* is t -representable.

Given an IV t -norm \mathbf{T} , we call the expression for item (a) of Theorem 5,

$$\mathbf{S}^*(\mathbf{x}, \mathbf{y}) = N(\mathbf{T}(N(\mathbf{x}), N(\mathbf{y}))) \text{ for all } \mathbf{x}, \mathbf{y} \in L([0, 1]),$$

the *dual* IV t -conorm of \mathbf{T} with respect to the IV negation N . Similarly, we call the expression for \mathbf{T}^* in item (c) the *dual* IV t -norm of the IV t -conorm \mathbf{S} with respect to the IV negation N [77].

Let the triple $(\mathbf{T}, \mathbf{S}, N)$ (where N is an involutive IV negation) denote that \mathbf{T} and \mathbf{S} are dual with respect to N ; any such triple is called a *dual triple* sometimes also called a De Morgan triple (see [54, 77] and Section 4 in [84]).

22.5 Laws for Conjunctions and Disjunctions

In this section, we set out to study properties of set theory such as idempotency, absorption, distributiveness, the law of contradiction, and the law of the excluded middle. These properties must be defined for IVFSs. We start from the work of Jenei [30] (see also [54, 76]) and we analyze the conditions under which IV t -norms and IV t -conorms satisfy or do not satisfy these properties. Some of the theorems that we present in this section are proved in [54] and others in [30].

22.5.1 Law of Contradiction and Law of the Excluded Middle

Here, we study the conditions under which the law of contradiction and the law of the excluded middle are satisfied for IVFSs. For this purpose, we understand the law of contradiction for IVFSs in this manner, analogously to the cases of classical and fuzzy sets: for all $\mathbf{x} \in L([0, 1])$, it should hold that $\mathbf{T}(\mathbf{x}, N(\mathbf{x})) = 0_L$. Similarly, the law of the excluded middle says that for all $\mathbf{x} \in L([0, 1])$, it should hold that $\mathbf{S}(\mathbf{x}, N(\mathbf{x})) = 1_L$.

Theorem 6. *The following items hold:*

- (a) If \mathbf{T}_{T_a, T_b} is a t -representable IV t -norm, then for any involutive IV negation, the law of contradiction is not satisfied.
- (b) If \mathbf{S}_{S_a, S_b} is an s -representable IV t -conorm, then for any involutive IV negation, the law of the excluded middle is not satisfied.

It is necessary to point out that there exist IV t -norms and IV t -conorms that are not t -representable or s -representable and satisfy the law of contradiction and the law of the excluded middle, respectively. This fact is made clear in the following theorem.

Theorem 7. (a) Let there be a t -norm T in $[0, 1]$ that satisfies the law of contradiction with respect to the involutive negation n and let the IV t -norm \mathbf{T}^T be generated by that t -norm. Then \mathbf{T}^T satisfies the law of contradiction with respect to the involutive IV negation N generated by n .

(b) Let there be a t -conorm S in $[0, 1]$ that satisfies the law of the excluded middle with respect to the involutive negation n and let the IV t -conorm \mathbf{S}^S be generated by that t -conorm. Then \mathbf{S}^S satisfies the law of the excluded middle with respect to the involutive IV negation N generated by n .

From Theorem 7 we deduce that there exist non- t -representable IV t -norms that satisfy the law of contradiction; evidently, we can also deduce that there exist non- s -representable IV t -conorms that satisfy the law of the excluded middle. Nevertheless, it is important to indicate that there exist non- t -representable IV t -norms that do not satisfy the law of contradiction, and there also exist non- s -representable IV

t -conorms that do not satisfy the law of the excluded middle; specific examples of these IV t -norms and IV t -conorms can be found in [54]. In that work there are also theorems that establish the manner in which one can construct from automorphisms non- t -representable IV t -norms that satisfy the law of contradiction (and similarly for non- s -representable IV t -conorms and the law of the excluded middle).

22.5.2 Idempotency, Absorption, and Distributivity

Theorem 8. *The following items hold:*

- (a) $\mathbf{T}(\mathbf{x}, \mathbf{x}) = \mathbf{x}$ for all $\mathbf{x} \in L([0, 1])$ (i.e., \mathbf{T} is idempotent) if and only if $\mathbf{T} = \mathbf{T}_{\min, \min}$.
- (b) $\mathbf{S}(\mathbf{x}, \mathbf{x}) = \mathbf{x}$ for all $\mathbf{x} \in L([0, 1])$ (i.e., \mathbf{S} is idempotent) if and only if $\mathbf{S} = \mathbf{S}_{\max, \max}$.
- (c) $\mathbf{T}(\mathbf{x}, \mathbf{S}(\mathbf{x}, \mathbf{y})) = \mathbf{x}$ for all $\mathbf{x}, \mathbf{y} \in L([0, 1])$ (property of absorption) if and only if \mathbf{T} is idempotent.
- (d) $\mathbf{S}(\mathbf{x}, \mathbf{T}(\mathbf{x}, \mathbf{y})) = \mathbf{x}$ for all $\mathbf{x}, \mathbf{y} \in L([0, 1])$ (property of absorption) if and only if \mathbf{S} is idempotent.

A consequence of Theorem 8 is the following: *the algebraic structure $\{\text{IVFSs}, \mathbf{T}, \mathbf{S}\}$ is only a lattice when we take $\mathbf{T} = \mathbf{T}_{\min, \min}$ and $\mathbf{S} = \mathbf{S}_{\max, \max}$.*

Theorem 9. *Let \mathbf{T}_{T_a, T_b} be a t -representable IV t -norm and let \mathbf{S}_{S_a, S_b} be an s -representable IV t -conorm. Under these conditions, the following items hold:*

- (a) $\mathbf{S}_{S_a, S_b}(\mathbf{x}, \mathbf{T}_{T_a, T_b}(\mathbf{y}, \mathbf{z})) = \mathbf{T}_{T_a, T_b}(\mathbf{S}_{S_a, S_b}(\mathbf{x}, \mathbf{y}), \mathbf{S}_{S_a, S_b}(\mathbf{x}, \mathbf{z}))$
for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in L([0, 1])$, if and only if $\mathbf{T}_{T_a, T_b} = \mathbf{T}_{\min, \min}$.
- (b) $\mathbf{T}_{T_a, T_b}(\mathbf{x}, \mathbf{S}_{S_a, S_b}(\mathbf{y}, \mathbf{z})) = \mathbf{S}_{S_a, S_b}(\mathbf{T}_{T_a, T_b}(\mathbf{x}, \mathbf{y}), \mathbf{T}_{T_a, T_b}(\mathbf{x}, \mathbf{z}))$
for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in L([0, 1])$, if and only if $\mathbf{S}_{S_a, S_b} = \mathbf{S}_{\max, \max}$.

By Theorems 8 and 9 we have on the one hand the following result: $\{\text{IVFSs}, \mathbf{T}_{\min, \min}, \mathbf{S}_{\max, \max}\}$ is a distributive lattice, and on the other, the following corollary.

Corollary 2. *Let \mathbf{T}_{T_a, T_b} be a t -representable IV t -norm and let \mathbf{S}_{S_a, S_b} be an s -representable IV t -conorm. Under these conditions, we have distributiveness if and only if we have absorption, if and only if we have idempotency, and*

$$\text{if and only if } \mathbf{T}_{T_a, T_b} = \mathbf{T}_{\min, \min} \text{ and } \mathbf{S}_{S_a, S_b} = \mathbf{S}_{\max, \max}.$$

Regarding the distributive property of non- t -representable IV t -norms and non- s -representable IV t -conorms, there exist situations in which this property does not hold, as we will see in the following theorem which is an immediate consequence of this: if $(\mathbf{T}, \mathbf{S}, N)$ satisfies the distribution laws, then $\mathbf{T} = \mathbf{T}_{\min, \min}$ and $\mathbf{S} = \mathbf{S}_{\max, \max}$, but then \mathbf{T} does not satisfy the law of contradiction, since for any $\mathbf{x} \in L([0, 1])$, $N(\mathbf{x}) \in L([0, 1])$ (since N is involutive), so $\mathbf{T}(\mathbf{x}, N(\mathbf{x})) \geq_L 0_L$.

Theorem 10. *Let $(\mathbf{T}, \mathbf{S}, N)$ be a dual triple that satisfies the law of the excluded middle and the law of contradiction. Then $(\mathbf{T}, \mathbf{S}, N)$ does not satisfy the distributive laws.*

22.6 Interval-Valued Fuzzy Implication Operators

In fuzzy set theory, a fuzzy implication operator I is a function $[0, 1]^2 \rightarrow [0, 1]$ that fulfills a certain set of properties (see [20]), so that $I(\mu_A(x), \mu_B(y))$ represents the degree of truth of the fuzzy conditional *If x is A then y is B*, where A and B are fuzzy sets.

A possible adaptation of the concept of the fuzzy implication operator to the case of IVFS theory has been given in [62] in the following way.

Definition 6. An interval-valued fuzzy implication operator is a function

$$I_{IV} : (L([0, 1]))^2 \rightarrow L([0, 1])$$

that has the following properties:

I_{IV0} . If $\mathbf{x}, \mathbf{y} \in L([0, 1])$ are such that $K_0(\mathbf{x}) = K_1(\mathbf{x})$ and $K_0(\mathbf{y}) = K_1(\mathbf{y})$, then $W_{I_{IV}}(\mathbf{x}, \mathbf{y}) = 0$.

I_{IV1} . If $\mathbf{x} \leq_L \mathbf{x}'$, then $I_{IV}(\mathbf{x}, \mathbf{y}) \geq_L I_{IV}(\mathbf{x}', \mathbf{y})$ for all $\mathbf{y} \in L([0, 1])$.

I_{IV2} . If $\mathbf{y} \leq_L \mathbf{y}'$, then $I_{IV}(\mathbf{x}, \mathbf{y}) \leq_L I_{IV}(\mathbf{x}, \mathbf{y}')$ for all $\mathbf{x} \in L([0, 1])$.

I_{IV3} . $I_{IV}(0_L, \mathbf{x}) = 1_L$ for all $\mathbf{x} \in L([0, 1])$.

I_{IV4} . $I_{IV}(\mathbf{x}, 1_L) = 1_L$ for all $\mathbf{x} \in L([0, 1])$.

I_{IV5} . $I_{IV}(1_L, 0_L) = 0_L$.

If $\mathbf{x}, \mathbf{y} \in L([0, 1])$, I_{IV0} establishes that if the sets are fuzzy, that is, the length of the intervals is zero, then the length of the interval-valued fuzzy implication operator is zero. Moreover, if in Definition 6 we replace $L([0, 1])$ by $[0, 1]$ and we eliminate the condition I_{IV0} , then we have the definition of a fuzzy implication operator in the sense of Fodor (see [20]).

This definition (Definition 6) extends classical two-valued implication; that is

$$I_{IV}(0_L, 0_L) = I_{IV}(0_L, 1_L) = I_{IV}(1_L, 1_L) = 1_L \quad \text{and} \quad I_{IV}(1_L, 0_L) = 0_L.$$

In [73], Cornelis *et al.* presented the following definition: An *implicator on $L([0, 1])$* is any $(L([0, 1]))^2 \rightarrow L([0, 1])$ mapping \mathcal{I} satisfying $\mathcal{I}(0_L, 0_L) = \mathcal{I}(0_L, 1_L) = \mathcal{I}(1_L, 1_L) = 1_L$ and $\mathcal{I}(1_L, 0_L) = 0_L$. Moreover, we require \mathcal{I} to be decreasing in its first component and increasing in its second component.

Beginning from this concept, *S*-implicators on $L([0, 1])$ and *R*-implicators on $L([0, 1])$ were defined in [73]. The axioms of Smets and Magrez (see [87]) were adapted to the case of IVFSs, and various characterization theorems for *S*- and *R*-implicators on $L([0, 1])$ were analyzed. It was also made clear that every interval-valued implication operator I_{IV} (Definition 6) is an implicator on $L([0, 1])$.

We must point out that the characterization and construction theorems for the implicators on $L([0, 1])$ (see [73]) are a generalization of the construction and characterization theorems for fuzzy *S*-implications and fuzzy *R*-implications (see [20]). In this connection, we see the following open problem. In the case of fuzzy sets, various characterizations of fuzzy *S*-implications, fuzzy *R*-implications, fuzzy *QL*-implications, and fuzzy *D*-implications have been studied, using in some cases the property $I(x, n(x)) = n(x)$ for all $x \in [0, 1]$. It is now necessary to carry out a study parallel to that in [88, 89], adapting the definitions and properties described in those publications to the implicators on $L([0, 1])$ defined above.

In [62], the properties usually required of interval-valued fuzzy implication operators are presented. These properties can be divided into two groups: the ones that result from adapting the properties of fuzzy implication operators to the interval-valued case and those that are interval-valued per se. For the latter group, the following three properties have been proposed, among others:

I_{IV6} . $W_{I_{IV}}(\mathbf{x}, \mathbf{y}) \leq \vee(1 - K_0(\mathbf{x}), 1 - K_1(\mathbf{x}), 1 - K_0(\mathbf{y}), 1 - K_1(\mathbf{y}))$.

I_{IV7} . If $\mathbf{x} = \mathbf{y}$, then $W_{I_{IV}}(\mathbf{x}, \mathbf{y}) = W_{\mathbf{x}}$.

I_{IV8} . If $W_{\mathbf{x}} = W_{\mathbf{y}}$, then $W_{I_{IV}}(\mathbf{x}, \mathbf{y}) = W_{\mathbf{x}}$.

Note that if the property I_{IV8} holds, then I_{IV7} holds, but the converse does not hold. Evidently, neither I_{IV7} nor I_{IV8} is in contradiction with I_{IV6} .

22.6.1 A Construction Method

In [73], various methods for the construction of implicators on $L([0, 1])$ are presented. These methods are characterized by the fact that they always use the extremes of the intervals. However, in the construction method for interval-valued fuzzy implication operators that we shall present next (see [62]), other points

not obtained by applying t -norms and t -conorms in $[0, 1]$ to the extremes of the intervals, such as the average point of the interval, can also be used. This method comes out of the following interpretation of the condition rule (with IVFSs).

In the theory of IVFSs, a general rule for an expert system has the form *If u is A then v is B*, where u is a variable taking values in U , v is a variable taking values in V , $A \in \mathcal{IVFSs}(U)$, and $B \in \mathcal{IVFSs}(V)$.

The interval $A(u)$ is the truth degree of the proposition ' u is A '. Let us take two values, the extremes of the intervals, $K_0(A(u))$ and $K_1(A(u))$. Since in reality we are interested in assigning a single value to the degree of truth of the proposition ' u is A ', we shall say that it is given by an aggregation of $K_0(A(u))$ and $K_1(A(u))$, that is, by $M_1(K_0(A(u)), K_1(A(u)))$. The choice of the aggregation will depend on the experimental situation we are dealing with.

With respect to ' v is B ', we can say the same. As before, we shall say that the truth degree of the proposition ' v is B ' is given by an aggregation of $K_0(B(v))$ and $K_1(B(v))$, that is, $M_2(K_0(B(v)), K_1(B(v)))$.

Therefore, bearing in mind these considerations and the interpretation of the fuzzy implication operator I , we shall say that a value that represents the truth degree of the interval-valued fuzzy conditional *If u is A then v is B* (where A, B are interval-valued fuzzy sets) is given by

$$I(M_1(K_0(A(u)), K_1(A(u))), M_2(K_0(B(v)), K_1(B(v)))) ,$$

I being any fuzzy implication operator in Fodor's sense.

We can perform an analogous reasoning for the degree of non-truth of the proposition: if we take as the non-truth degree of the proposition ' u is A ' two values that are the negation of the extremes of the intervals, we obtain the result that the degree of non-truth of the proposition ' u is A ' will be given by any aggregation of $1 - K_1(A(u))$ and $1 - K_0(A(u))$, that is, by $M_3(1 - K_1(A(u)), 1 - K_0(A(u)))$. Likewise, the non-truth degree of the proposition ' v is B ' will be given by $M_4(1 - K_1(B(v)), 1 - K_0(B(v)))$, M_4 being an aggregation operator.

Following further analogous reasoning, we have the result that the non-truth degree of the interval-valued fuzzy conditional *If u is A then v is B* is given by

$$1 - I'(1 - M_3(1 - K_1(A(u)), 1 - K_0(A(u)))), 1 - M_4(1 - K_1(B(v)), 1 - K_0(B(v))) ,$$

I' being any fuzzy implication operator in Fodor's sense.

Evidently, we can say that another value represents the degree of truth of the interval-valued fuzzy conditional *If u is A then v is B* (where A and B are interval-valued fuzzy sets). This is given by

$$I'(1 - M_3(1 - K_1(A(u)), 1 - K_0(A(u)))), 1 - M_4(1 - K_1(B(v)), 1 - K_0(B(v))) .$$

Therefore, we can interpret the interval

$$\begin{aligned} & [I(M_1(K_0(A(u)), K_1(A(u))), M_2(K_0(B(v)), K_1(B(v)))), \\ & I'(1 - M_3(1 - K_1(A(u)), 1 - K_0(A(u)))), 1 - M_4(1 - K_1(B(v)), 1 - K_0(B(v))))] \end{aligned}$$

as the truth degree of the interval-valued fuzzy conditional *If u is A then v is B*. (Obviously, in the expression above, I , I' , M_1 , etc., are taken such that the expression is an interval.)

In the following proposition, we present a construction method for interval-valued fuzzy implication operators in the sense of Definition 6 (see [62]). The aggregations that we use in this construction method are the following: an *aggregation operator* is a $[0, 1]^2 \rightarrow [0, 1]$ mapping M that satisfies the following conditions:

1. $M(0, 0) = 0$; $M(1, 1) = 1$.
2. M is increasing in its first and second argument.
3. $M(x, y) = M(y, x)$ for all $x, y \in [0, 1]$.

Proposition 1. Let I be a fuzzy implication operator in Fodor's sense. Let M_1 , M_2 , M_3 , and M_4 be four idempotent aggregation operators such that

$$\begin{cases} M_1(x, y) + M_3(1 - x, 1 - y) \geq 1 \\ M_2(x, y) + M_4(1 - x, 1 - y) \leq 1, \end{cases}$$

for all $x, y \in [0, 1]$. Then

$$I_{IV} : (L([0, 1]))^2 \rightarrow L([0, 1]), \text{ given by} \\ I_{IV}(\mathbf{x}, \mathbf{y}) = [I(M_1(K_0(\mathbf{x}), K_1(\mathbf{x})), M_2(K_0(\mathbf{y}), K_1(\mathbf{y}))), \\ I(1 - M_3(1 - K_1(\mathbf{x}), 1 - K_0(\mathbf{x})), 1 - M_4(1 - K_1(\mathbf{y}), 1 - K_0(\mathbf{y})))]$$

is an interval-valued fuzzy implication operator in the sense of Definition 6.

In [62], there is a study of the conditions under which the constructions of Proposition 1 fulfill the properties I_{IV6} – I_{IV8} , among others.

Example 1. If $M_1 = M_3 = \max$ and $M_2 = M_4 = \min$, then we obtain the interval-valued fuzzy implication operator introduced by Jenei (see [30]); that is,

$$I_{IV}(\mathbf{x}, \mathbf{y}) = [I(K_1(\mathbf{x}), K_0(\mathbf{y})), I(K_0(\mathbf{x}), K_1(\mathbf{y}))].$$

Example 2. If, under the conditions of Example 1, we take the Kleene–Dienes fuzzy implication operator, that is, $I(x, y) = \max(1 - x, y)$, we have the expression for the first interval-valued fuzzy operator introduced by Atanassov (see [35]),

$$I_{IV}(\mathbf{x}, \mathbf{y}) = [\max(1 - K_1(\mathbf{x}), K_0(\mathbf{y})), \max(1 - K_0(\mathbf{x}), K_1(\mathbf{y}))].$$

In [62], it is proved that the expression in Example 2 satisfies the properties I_{IV6} , I_{IV7} , and I_{IV8} . Cornelis et al. proved in [73] that this expression is an S -implicator on $L([0, 1])$. Moreover, they also proved that if, under the conditions of Proposition 1, we take $M_1(x, y) = M_2(x, y) = M_3(x, y) = M_4(x, y) = (x + y)/2$ and the Kleene–Dienes implication operator, the expression that we obtain is not an S -implicator on $L([0, 1])$ or an R -implicator on $L([0, 1])$.

In [73], the definition of an interval-valued fuzzy S -implication (in Fodor's sense) is presented and in [63] the conditions under which these implications satisfy property I_{IV7} are studied. In this connection, the following open problems are worth mentioning:

1. To study the cases for which interval-valued fuzzy S -implications satisfy the property I_{IV8} .
2. To analyze the conditions that must be met in order for interval-valued fuzzy R -implications to satisfy the property I_{IV8} .
3. To define interval-valued fuzzy D -implications (see [88]) and study the conditions under which these implications satisfy I_{IV8} and the property that results from generalizing the fuzzy property $I(x, n(x)) = n(x)$ for all $x \in [0, 1]$ to the case of IVFSs.

22.7 IV Information Measures

The purpose of this section is to introduce the main developments and results regarding the best-known interval-valued information measures. The study that we present is for finite referentials. The non-finite case has been analyzed in the corresponding references.

Indetermination Index

In 1975, Sambuc [4] presented the following definition: given $A \in IVFSs(U)$, the *indetermination index* of the set A is the following expression:

$$\mathcal{J}(A) = \frac{\sum_{i=1}^{\mathcal{N}} W(A(u_i))}{\mathcal{N}}, \quad \text{where } \mathcal{N} = \text{Card}(U). \quad (2)$$

Sambuc used (2) to determine how far the IVFS considered was from the corresponding fuzzy set that he would have taken if he had used fuzzy set theory in his work. In fact, he reached the expression (2) by applying the concept of the Hamming distance between the extremes of the intervals. He also presented an expression for this index when the Euclidean distance was used for its construction.

IV Distances

A lot has been written on the concept of distance between IVFSs. In the following definition, we present the currently most commonly used expressions (see [90, 91]).

Definition 7. Let U ($\text{Card}(U) = \mathcal{N}$) be the referential set. We define the following distances:

1. The normalized Euclidean distance between A, B belonging to $IVFSs(U)$,

$$D_W^E(A, B) = \sqrt{\frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \frac{(K_0(A(u_i)) - K_0(B(u_i)))^2 + (K_1(A(u_i)) - K_1(B(u_i)))^2 + (W(A(u_i)) - W(B(u_i)))^2}{2}}.$$

2. The normalized Hamming distance between $A, B \in IVFSs(U)$,

$$D_W^H(A, B) = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \frac{|K_0(A(u_i)) - K_0(B(u_i))| + |K_1(A(u_i)) - K_1(B(u_i))| + |W(A(u_i)) - W(B(u_i))|}{2}.$$

3. The normalized Hausdorff distance between $A, B \in IVFSs(U)$,

$$D_W(A, B) = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \max(|K_0(A(u_i)) - K_0(B(u_i))|, |K_1(A(u_i)) - K_1(B(u_i))|, |W(A(u_i)) - W(B(u_i))|).$$

Historically, the expressions developed in Definition 7 were defined without including the term relative to the length of the intervals (see [4, 34, 60, 61]).

Degree of Compatibility

In 1987, Gorzalczany (see [9]) defined the *degree of compatibility* between two interval-valued fuzzy sets A and B on the same referential U in the following way:

The degree of compatibility $\Gamma(A, B)$ of an interval-valued fuzzy set A (such that there is at least one $u \in U$ with $K_0(u) \neq 0$) with an interval-valued fuzzy set B being an element of $L([0, 1])$ is given by

$$\begin{aligned} \Gamma(A, B) = & \left[\min \left(\frac{\max_{u \in U} (\min(K_0(A(u)), K_0(B(u))))}{\max_{u \in U} K_0(A(u))}, \frac{\max_{u \in U} (\min(K_1(A(u)), K_1(B(u))))}{\max_{u \in U} K_1(A(u))} \right), \right. \\ & \left. \max \left(\frac{\max_{u \in U} (\min(K_0(A(u)), K_0(B(u))))}{\max_{u \in U} K_0(A(u))}, \frac{\max_{u \in U} (\min(K_1(A(u)), K_1(B(u))))}{\max_{u \in U} K_1(A(u))} \right) \right]. \end{aligned}$$

A study of the properties of this concept and a very interesting application of it can be found in [60].

IV Entropies

There are two different definitions in the literature of the concept of an interval-valued fuzzy entropy (IV entropy). The first, \mathcal{E}_F , was presented in 1996 in [61], and the second, \mathcal{E}_c , in 2001 in [91]. The difference between the two definitions lies in the fact that \mathcal{E}_F is a measure of how far an IVFS is from a

fuzzy set, whereas \mathcal{E}_c is a measure of how far an IVFS is from a crisp set. Therefore, \mathcal{E}_F is based on the ideas of Sambuc and \mathcal{E}_c is based on the concept of fuzzy entropy.

Definition 8. A real function $\mathcal{E}_F : \text{IVFSs}(U) \rightarrow \mathbb{R}^+$ is called an entropy on $\text{IVFSs}(U)$ if \mathcal{E}_F has the following properties:

- (IF1) $\mathcal{E}_F(A) = 0$ if and only if $A \in \mathcal{FSS}(U)$.
- (IF2) $\mathcal{E}_F(A) = \text{Card}(U) = 1$ if and only if $K_0(A(u)) = 0$ and $K_1(A(u)) = 1$ for all $u \in U$.
- (IF3) $\mathcal{E}_F(A) = \mathcal{E}_F(A_N)$ for all $A \in \text{IVFSs}(U)$. (N is the involutive and strict IV negation generated by $n(x) = 1 - x$ for all $x \in [0, 1]$ according to Theorem 2.)
- (IF4) If $A \preceq B$, then $\mathcal{E}_F(A) \geq \mathcal{E}_F(B)$.

In [61], these entropies \mathcal{E}_F are studied in depth and a theorem for their construction from functions $\varphi : [0, 1] \rightarrow [0, 1]$ is presented. It is also said there that the most commonly used expression is the one obtained when we take $\varphi(x) = x$ so that $\mathcal{E}_F(A) = \mathcal{J}(A)$ for all $A \in \text{IVFSs}(U)$. The edge detector developed in [54] for objects in an image uses the IV entropy \mathcal{E}_F .

Definition 9. A real function $\mathcal{E}_c : \text{IVFSs}(U) \rightarrow \mathbb{R}^+$ is called an entropy on $\text{IVFSs}(U)$ if \mathcal{E}_c has the following properties:

- (Ic1) $\mathcal{E}_c(A) = 0$ if and only if A is a crisp set.
- (Ic2) $\mathcal{E}_c(A) = 1$ if and only if $K_0(A(u)) = 1 - K_1(A(u))$ for all $u \in U$.
- (Ic3) $\mathcal{E}_c(A) = \mathcal{E}_c(A_N)$ for all $A \in \text{IVFSs}(U)$. (N is the involutive and strict IV negation generated by $n(x) = 1 - x$ for all $x \in [0, 1]$ according to Theorem 2.)
- (Ic4) $\mathcal{E}_c(A) \leq \mathcal{E}_c(B)$ if $K_0(A(u)) \leq K_0(B(u))$ and $K_1(A(u)) \leq K_1(B(u))$ for $K_0(B(u)) \leq 1 - K_1(B(u))$ or $K_0(A(u)) \geq K_0(B(u))$ and $K_1(A(u)) \geq K_1(B(u))$ for $K_0(B(u)) \geq 1 - K_1(B(u))$.

A study of the main properties of \mathcal{E}_c can be found in [39, 91]. In this latter paper, a first attempt to relate \mathcal{E}_F to \mathcal{E}_c by means of a novel construction method is presented.

The idea of relating \mathcal{E}_c to IV similarities using item (2) of Definition 7 has led Szmidt and Kacprzyk to modify the definition of \mathcal{E}_c (see [92]).

It is necessary to point out that Tizhoosh (see [50]) experimentally proves that in the determination of the threshold of an image, the entropy \mathcal{E}_F provides very good results. This author at no time considers the IV entropy: \mathcal{E}_c .

In spite of the results obtained by Tizhoosh, we consider that we have the following open problem: defining interval-valued fuzzy entropy so that it gives as a result an element of $L([0, 1])$ and not an element of $[0, 1]$. We should also study the conditions under which we recover Definitions 8 and 9 from the new definition and in this way we can analyze the reasons why the use of \mathcal{E}_F gives good results in threshold computation in image processing.

IV Similarity

In [60], there is also a section on the similarity of IVFSs. First, a normal interval-valued similarity measure $S(A, B)$ between two IVFSs A and B is defined as one that satisfies the following five properties: (i) $S(A, B) = S(B, A)$ for all $A, B \in \text{IVFSs}(U)$; (ii) $S(D, D_C) = 0_L$ for all $D \in \mathcal{P}(U)$, where D_C is the complement of D and $\mathcal{P}(U)$ is the class of all crisp sets of U ; (iii) $S(C, C) = 1_L$ for all $C \in \text{IVFSs}(U)$; (iv) for all $A, B, C \in \text{IVFSs}(U)$, if $A \leq B \leq C$, then $S(A, B) \geq S(A, C)$ and $S(B, C) \geq S(A, C)$; and (v) if $A, B \in \text{IVFSs}(U)$, then $S(A, B) \in L([0, 1])$. In [60], it is proved that the relation

$$S(A, B) = [S_L(A, B), S_U(A, B)], \quad (3)$$

where $S_L(A, B)$ is a fuzzy similarity measure (see [93]) between the lower membership functions of A and B , and $S_U(A, B)$ is a fuzzy similarity measure between the upper membership functions of A

and B , satisfies (i)–(v). There are additional results, but these are beyond the scope of this chapter.

Starting from the work of Mitchell (see [94]), Mendel presented in [17] a new expression for IV similarities. In the same paper, Mendel posed the following open question: finding the possible connection (if there is one) between the similarity measure (3) and the expression that he proposes.

The IV similarities defined in [17, 60] have in common the fact that they give an interval as a result, that is, an element of $L([0, 1])$. However, the IV similarities proposed in [70, 95–98] are an adaptation of the well-known measures of similarity between fuzzy sets, so that they give an element of $[0, 1]$ as a result. Evidently, in this case the expressions that relate IV entropy (\mathcal{E}_c), IV similarity, and IV distances are identical to those obtained by Liu (see [93]) in 1992 for fuzzy sets. At this point, we would like to ask the following question:

Should the information measures between IVFSs be such that they give as a result an element of $L([0, 1])$ or an element of $[0, 1]$?

We think that this is the first problem that we should approach. Our particular opinion is that these measures should give elements of $L([0, 1])$ as a result. Furthermore, we think that if we have an interval-valued measure Π (e.g., the IV similarity), such that for each pair of elements $A, B \in \mathcal{IVFSs}(U)$ it gives as the result the interval $\Pi(A, B) \in L([0, 1])$, then it should turn out that for certain values of $\alpha \in [0, 1]$ when a fuzzy measure π is applied (e.g., fuzzy similarity) in association with Π , the following equality holds:

$$K_\alpha(\Pi(A, B)) = \pi(K_\alpha(A), K_\alpha(B)). \quad (4)$$

We consider that in the future we must analyze the conditions under which information measures satisfy (4). Evidently, these measures should be defined so that they give as a result an element of $L([0, 1])$.

We have said before that in the future we must define the entropy of interval-valued fuzzy sets as an element of $L([0, 1])$. Once this definition has been made we must analyze different methods for the construction of these entropies and study their generation from IV similarities. We must also relate IV distances, IV similarities, and the new definition of IV entropy. To do this we must bear in mind each and every one of the ideas described in the references used up to now on the topic and on the results obtained in the following works: [75, 100–106].

IV Inclusion Measures, IV Correlation, and IV Information Energy

In [60], inclusion measures between IVFSs were studied for the first time. These measures give an element of $L([0, 1])$ as a result. Also, various methods of construction of these measures from fuzzy implication operators were analyzed. In 2003, Kehagias and Konstantinidou proposed a new version of these measures [107].

In [75], Cornelis and Kerre present a definition of interval-valued inclusion measure in the same sense as the one presented in [60]; that is, the measure gives as a result an element of $L([0, 1])$. They propose an axiomatization that enables them to relate their inclusion measure with the expression of IV entropy \mathcal{E}_c . We consider that any study done to relate IV entropy, IV similarity, and IV distance must take into account the results obtained by these authors in order to generate \mathcal{E}_c from IV inclusion measures. We must also highlight the theoretical development made for using IV inclusion measures in approximate reasoning.

Finally, a method for the construction of interval-valued fuzzy entropies from IV inclusion measures is presented in [96]. However, in this case the method proposed is an adaptation to IVFSs of well-known results of fuzzy set theory, for the authors believe that the result of any information measure with IVFSs should always be an element of $[0, 1]$.

Gerstenkorn and Manko [108] introduced the concepts of IV correlation and IV information energy. Later, in [109], a detailed study of these concepts was carried out.

22.8 Some Applications

Our goal in these pages is not to go into all the details of each and every one of the fields where IVFSs are being applied; we wish to present only some representative contributions. Nevertheless, we think that we should make the following remarks:

1. In most of the applications that we are going to present it is proved that when we have great imprecision in the determination of the membership degrees, better results are obtained by modelizing with IVFSs than with FSs.
2. The use of IVFSs does not increase the complexity of the algorithms, it only increases the number of necessary calculations for each algorithm. On the other hand, bearing in mind the latest technological advances, it results that the expense in time for the execution of the algorithms with IVFSs is practically the same as that of the algorithms with FSs. (This fact is made especially clear in algorithms that use IVFSs for image processing.)

Approximate Reasoning

Approximate reasoning is, formally speaking, as Turksen says [110], the process or processes by which a possible imprecise conclusion is deduced from a collection of imprecise premises. In this section, we present a short review of the inference methods most commonly used in the literature when the imprecise premises and the imprecise conclusions are represented using IVFSs.

The generalized modus ponens (GMP) inference rule with IVFSs is represented in the following way (see [9, 65, 73, 111]):

$$\begin{array}{c} \text{If } u \text{ is } A \text{ then } v \text{ is } B \\ u \text{ is } A' \\ \hline v \text{ is } B', \end{array}$$

where u is a variable taking values in U , v is a variable taking values in V , $A, A' \in \mathcal{IVFSs}(U)$, and $B, B' \in \mathcal{IVFSs}(V)$.

The methods given in order to obtain the conclusion B' can be divided into two groups: the ones that use an adaptation of Zadeh's compositional rule (see [1, 112]) to the interval-valued fuzzy case and those that do not.

The idea of applying Zadeh's compositional rule to the GMP with IVFSs led to the study of interval-valued fuzzy relations (IVFRs). In [64, 65, 67, 80, 113], the properties of IVFRs were analyzed and the composition of such relations was studied. Afterward, these relations were applied to the computation of the conclusion of the GMP with IVFSs. In [114], interval-valued fuzzy equations were studied for the first time (see [115]). We must say that the field of IVFRs is the least studied of those presented in this chapter.

Arnould and Tano in 1995 (see [116]) constructed an expert system using rules with IVFSs. In the inference engine of that expert system, Zadeh's compositional rule is first applied to the lower extremes of the intervals and then to the upper extremes.

With respect to the second group, that is, the applications that do not use Zadeh's composition rule, it is worth pointing out that all of them use the algorithm proposed by Gorzalczany (see [12]). This algorithm consists of two steps:

1. Relate A to A' by means of an information measure.
2. Build the consequence B' using the result of the comparison above and B .

In [12], Gorzalczany used for step 1 the concept of the degree of compatibility, whereas in [60] first the degree of inclusion was used and then the IV similarities. Note that the measures used always give an element of $L([0, 1])$ as a result.

We must point out that regardless of the method used for calculating $B' \in \mathcal{IVFSs}(V)$, we must always study properties of the type 'if $A' = A$, then $B' = B$ ', etc. In this connection, in [60] there is an in-depth

analysis of the conditions under which the methods developed in that paper satisfy the axioms of Fukami *et al.* (see [117]) or the axioms of Baldwin and Pilsworth (see [118]). A similar study has been carried out in [65, 73] for the methods that use Zadeh's compositional rule adapted to the interval-valued fuzzy case.

In [12, 64, 65, 68, 70, 116, 119], various methods for obtaining the conclusion from a system of interval-valued rules, in addition to the GMP with IVFSs, can be found.

Image Processing

A very important problem in image processing is the detection of the edges of the objects that make up an image. A pixel is said to belong to an edge if it has associated with it a big enough change in intensity. In [54], each image is associated with an IVFS so that the length of the interval associated with each pixel represents the intensity change between that pixel and its neighbors. Therefore, if the length of the interval is large enough, then the pixel belongs to an edge. The edge detector developed in [54] surpassed the classical detectors in the literature in three of the four types in which images were classified. In [54], an expression for calculating the contrast of an image using interval-valued S -implication operators is given.

In Section 22.2.2 we have said that Tizhoosh developed a method for calculating the threshold of an image using IVFSs. In [50] and later in [51] and [52] it has been experimentally proved that when working with images that have a large number of pixels such that the experts are not able to determine precisely whether they belong to the background or to the object, this method provides better results than the rest of known methods. In any other type of images, it has also been proved that the results obtained with this algorithm are similar to those obtained with algorithms that do not use IVFSs.

The algorithm proposed by Tizhoosh for images with L levels of gray consists of the following steps:

1. Assign L fuzzy sets to each image Q .
2. Associate with each fuzzy set its corresponding IVFS constructed with the method described in Section 22.2.2.
3. For each IVFS, calculate \mathcal{E}_F .
4. Take as threshold the intensity corresponding to IVFS with lowest value of \mathcal{E}_F .

We must point out that Tizhoosh calls \mathcal{E}_F (see again [61]) ultrafuzziness and that the algorithm above has been generalized in [51, 52]. In Figure 22.2 we give an example where it is proved that we obtain the best result with this algorithm. In [54] it has been proved that the time and memory efficiency of the algorithms that use IVFSs (for image processing) is practically the same as the efficiency of those that do not use IVFSs.

In [120], Gaussian noise was eliminated from an image using algorithms with IVFSs. In general, each and every one of the applications constructed with IVFSs for image processing (when it is applied to images with great imprecision in the membership of the pixels that compose it) gives better results than those constructed with fuzzy sets. This is due to the fact that they use a characteristic of IVFSs that fuzzy sets do not have, namely, length (of interval).

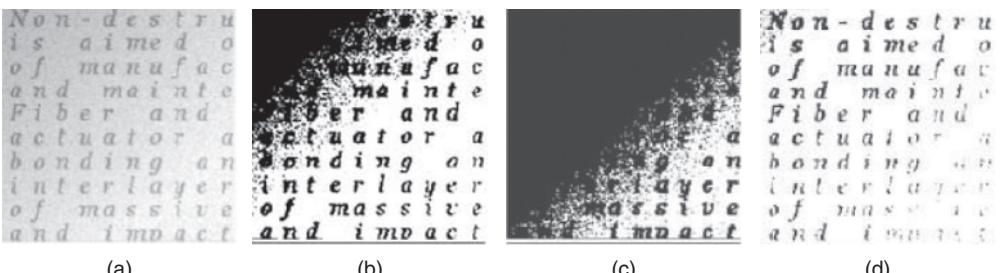


Figure 22.2 (a) Original image; (b) image binarized with the classical Otsu method; (c) image binarized with the Huang fuzzy method; (d) image binarized with IVFS method

Computing with Words

In [121] Zadeh defines computing with words (CWW) in the following way: *a theory in which the objects of computation are words and propositions drawn from a natural language*. We know that words can have different meanings depending on the person and the context in which the person is. Therefore there is uncertainty associated with the words (see [17, 59, 122]). This reasoning has made several authors consider it necessary to use IVFSs in CWW (see [123]).

Decision Making

In decision making, one works with many types of information, among them interval-valued information. In [8], various methods of processing information represented in this way are analyzed. Furthermore, in [124, 125], there is a study of several processes of aggregation of non-homogeneous information with contexts composed of numerical, interval-valued, and linguistic values. It is also worth pointing out the work of Bilgiç (see, e.g., [67]). More applications in decision making can be found in [126–128].

Other Applications

Obviously there exist a lot of applications of IVFSs; here we present the most representative. For applications to fuzzy linear programming, see [69, 129]. For economics, see [130, 131]. For medicine, see [4, 113]. For robotics, see [118, 132–134]. For fuzzy modeling, see [135–137]. For Web intelligence, see [56]. For the theory of possibility, see [73]. For control, see [10, 57].

22.9 Granular Computing and Interval-Valued Fuzzy Sets

The topic of fuzzy information granulation was first proposed and discussed by Zadeh in 1979 (see [138]). In this paper, a granule was defined as a collection of indistinguishable objects and granular information as the grouping of objects into granules. Later, Zadeh established in [139] the following: *the theory of fuzzy information granulation is inspired by the ways in which humans granulate information and reason with it*. On the basis of these considerations Yao (see [140]) says that granular computing may be regarded as a label of theories, methodologies, techniques, and tools that make of granules, that is, groups, classes, or clusters of a universe, in the process of problem solving.

Therefore, granular computing is based on the principle that the description of the world around us by means of techniques that use exclusively numeric precision is often unnecessary and has a very high cost. Furthermore, granular computing also considers the fact that human thinking does not work at a numeric level of precision but at a much more abstract level. In this sense in [81] the following is established: *granular computing is a formalism for expressing that abstraction within computational processes, thus endowing computer systems with a more human-centric view of the world. The central notion in granular computing is that there are many ‘levels’ of precision in which information about the real world can be expressed, with numeric precision being the most refined and a binary value the coarsest* (see [141]).

A granule can be interpreted as one of the numerous particles or elements that compose a unit. The size of the granule is its basic property. Intuitively the size can be interpreted as the degree of abstraction, detail, or precision.

In many applications, when the problem to solve deals with imprecise or vague information, it can be difficult to identify specific data and then we are forced to use granules. Furthermore, we have said in the introduction that the use of IVFSs arises in the applications where it is hard to precisely determine the value of the membership function for the elements. These two arguments allow us to say that IVFSs can be used in the tasks of representing, operating, and reasoning with granules (see [138]).

Depending on the application, a granule can be an element, that is, an interval, or it can be represented by an IVFS. In Section 22.2.2 we have seen that we can construct IVFSs from fuzzy sets; therefore, we can construct hierarchies of granules and represent arbitrary groups (clusters) of granules using IVFSs. This representation has the advantage that it enables us to use all of the results known for IVFSs in granular computing.

22.10 Conclusion

In this chapter, we have reviewed the basic concepts of IVFSs. We have focused on the elementary properties and have given a detailed set of references for more advanced properties. We have also posed several unsolved problems and have demonstrated the importance that the study of properties that differentiate them from fuzzy sets should have in future research.

As with all young theories, there are a great number of open problems in IVFS theory. We have presented some of them at various points in this chapter.

We have detected that currently there is general interest in finding applications of type 2 fuzzy sets and that the best results are almost always obtained when IVFSs are used. For this reason, we believe that these sets are going to be thoroughly studied and used in the next few years.

References

- [1] L.A. Zadeh. Fuzzy sets. *Inf. Control* 8 (1965) 338–353.
- [2] L.A. Zadeh. Outline of a new approach to analysis of complex systems and decision processes. *IEEE Trans. Syst. Man Cybern.* 3 (1973) 28–44.
- [3] G. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, Upper Saddle River, NJ, 1995.
- [4] R. Sambuc. *Function Φ -Flous, Application a l'aide au Diagnostic en Pathologie Thyroïdienne*. These de Doctorat en Medicine. University of Marseille (1975).
- [5] K.U. Jahn. Intervall-wertige Mengen. *Math. Nachr.* 68 (1975) 115–132.
- [6] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning – I. *Inf. Sci.* 8 (1975) 199–249.
- [7] I. Grattan-Guinness. Fuzzy membership mapped onto interval and many-valued quantities. *Z. Math. Log. Grundl. Math.* 22 (1976) 149–160.
- [8] A. Dziech and M.B. Gorzalczany. Decision making in signal transmission problems with interval-valued fuzzy sets. *Fuzzy Sets Syst.* 23(2) (1987) 191–203.
- [9] M.B. Gorzalczany. A method of inference in approximate reasoning based on interval-valued fuzzy sets. *Fuzzy Sets Syst.* 21 (1987) 1–17.
- [10] M.B. Gorzalczany. Interval-valued fuzzy controller based on verbal model of object. *Fuzzy Sets Syst.* 28(1) (1988) 45–53.
- [11] M.B. Gorzalczany. Interval-valued fuzzy inference involving uncertain (inconsistent) conditional propositions. *Fuzzy Sets Syst.* 29(2) (1989) 235–240.
- [12] M.B. Gorzalczany. An interval-valued fuzzy inference method. Some basic properties. *Fuzzy Sets Syst.* 31(2) (1989) 243–251.
- [13] I.B. Turksen. Interval valued fuzzy sets based on normal forms. *Fuzzy Sets Syst.* 20(2) (1986) 191–210.
- [14] I.B. Turksen. Interval-valued fuzzy sets and compensatory AND. *Fuzzy Sets Syst.* 51 (1992) 295–307.
- [15] I.B. Turksen and Z. Zhong. An approximate analogical reasoning schema based on similarity measures and interval-valued fuzzy sets. *Fuzzy Sets Syst.* 34 (1990) 323–346.
- [16] I.B. Turksen and D.D. Yao. Representation of connectives in fuzzy reasoning: The view through normal forms. *IEEE Trans. Syst. Man Cybern.* 14 (1984) 191–210.
- [17] J.M. Mendel. Advances in type-2 fuzzy sets and systems. *Inf. Sci.* 177 (2007) 84–110.
- [18] D. Dubois. Foreword. In: H. Bustince, F. Herrera, and J. Montero (eds), *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*. Springer, New York, 2007.
- [19] E. Trillas. Sobre funciones de negación en la teoría de conjuntos difusos. *Stochastica* III-1 (1979) 47–59 (in Spanish). English version in S. Barro, A. Sobrino, and A. Bugarin (eds), *Advances of Fuzzy Logic*. Universidad de Santiago de Compostela, 1998, pp. 31–43.
- [20] J. Fodor and M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support, Theory and Decision Library*. Kluwer Academic, Dordrecht, 1994.
- [21] P. Burillo and H. Bustince. Orderings in the referential set induced by an intuitionistic fuzzy relation. *Notes IFS* 1 (1995) 93–103.
- [22] C. Cornelis, G. Deschrijver, and E.E. Kerre. Advances and challenges in interval-valued fuzzy logic. *Fuzzy Sets Syst.* 157 (2006) 622–627.
- [23] G. Deschrijver and E.E. Kerre. On the relationship between some extensions of fuzzy set theory. *Fuzzy Sets Syst.* 133(2) (2003) 227–235.

- [24] G. Deschrijver. Arithmetic operators in interval-valued fuzzy set theory. *Inf. Sci.* 177(14) (2007) 2906–2924.
- [25] J. Lazaro, T Calvo, and XAO Operators. The interval universe. In: *Proceedings of 4th EUSFLAT 11th LFA*, Barcelona, Spain, 2005, pp. 189–197.
- [26] R.E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.
- [27] N.S. Nedialkov, V. Kreinovich, and S.A. Starks. Interval arithmetic, affine arithmetic, Taylor series methods: Why, what next? *Numer. Algo.* 37 (1–4 SPEC. ISS) (2004) 325–336.
- [28] Y.Y. Yao and J. Wang. Interval based uncertain reasoning using fuzzy and rough sets. In: P.P. Wang (ed), *Advances in Machine Intelligence & Soft Computing IV*. Department of Electrical Engineering, Duke University, Durham, North Carolina, 1997, pp. 196–215.
- [29] W. Zeng, Y. Shi, and H. Li. Representation theorem of interval-valued fuzzy set. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 14(3) (2006) 259–271.
- [30] S. Jenei. A more efficient method for defining fuzzy connectives. *Fuzzy Sets Syst.* 90 (1997) 25–35.
- [31] J.M. Mendel and R.I.J. Robert. Type-2 fuzzy sets made simple. *IEEE Trans. Fuzzy Syst.* 10(2) (2002) 117–127.
- [32] J.M. Mendel and H. Wu. Type-2 Fuzzistics for symmetric interval type-2 fuzzy sets: Part 1, forward problems. *IEEE Trans. Fuzzy Syst.* 14(6) (2006) 781–792.
- [33] J.M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems*. Prentice Hall, Upper Saddle River, NJ, 2001.
- [34] K. Atanassov. Intuitionistic fuzzy sets. In: *VIIth ITKR Session, Deposited in the Central Science and Technology Library of the Bulgarian Academy of Sciences*, Sofia, Bulgaria, 1983, pp. 1684–1697.
- [35] K. Atanassov. *Intuitionistic Fuzzy Sets. Theory and Applications*. Physica-Verlag, Heidelberg, 1999.
- [36] D. Gómez, J. Montero, and H. Bustince. Sobre los conjuntos intuicionistas fuzzy de Atanassov. In: *XIII Congreso Español sobre Técologías y Lógica Fuzzy*, ESTYLF'06 (in Spanish), Ciudad Real, Spain, 2006, pp. 319–324.
- [37] J. Montero, D. Gómez, and H. Bustince. On the relevance of some families of fuzzy sets. *Fuzzy Sets Syst.* 158(22) (2007) 2429–2442.
- [38] E. Szmidt. *Applications of Intuitionistic Fuzzy Sets in Decision Making*. D.Sc. dissertation. Technical University of Sofia, 2000.
- [39] I.K. Vlachos and G.D. Sergiadis. Inner product based entropy in the intuitionistic fuzzy setting. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 14(3) (2006) 351–367.
- [40] W.L. Gau and D.J. Buehrer. Vague sets. *IEEE Trans. Syst. Man Cybern.* 23(2) (1993) 751–759.
- [41] H. Bustince and P. Burillo. Vague sets are intuitionistic fuzzy sets. *Fuzzy Sets Syst.* 79 (1996) 403–405.
- [42] G. Deschrijver and E.E. Kerre. On the position of intuitionistic fuzzy set theory in the framework of theories modelling imprecision. *Inf. Sci.* 177 (2007) 1860–1866.
- [43] J.L. Deng. Introduction to grey system theory. *J. Grey Syst.* 1 (1989) 1–24.
- [44] J.A. Goguen. L-fuzzy sets. *J. Math. Anal. Appl.* 18(1) (1967) 623–668.
- [45] G.J. Wang and Y.Y. He. Intuitionistic fuzzy sets and L-fuzzy sets. *Fuzzy Sets Syst.* 110 (2000) 271–274.
- [46] K. Hirota. Concepts of probabilistic sets. *Fuzzy Sets Syst.* 5 (1981) 31–46.
- [47] K. Basu, R. Deb, and P.K. Pattanaik. Soft sets: An ordinal formulation of vagueness with some applications to the theory of choice. *Fuzzy Sets Syst.* 45 (1992) 45–58.
- [48] I.B. Turksen. Fuzzy normal forms. *Fuzzy Sets Syst.* 69 (1995) 319–346.
- [49] P. Burillo and H. Bustince. Construction theorems for intuitionistic fuzzy sets. *Fuzzy Sets Syst.* 84 (1996) 271–281.
- [50] H.R. Tizhoosh. Image thresholding using type-2 fuzzy sets. *Pattern Recognit.* 38 (2005) 2363–2372.
- [51] H. Bustince, V. Mohedano, E. Barrenechea, and M. Pagola. An algorithm for calculating the threshold of an image representing uncertainty through A-IFSs. In: *Proceedings of 11th Information Processing and Management of Uncertainty in Knowledge-Based Systems*, IPMU'06, Paris, France, 2006, pp. 2383–2390.
- [52] H. Bustince, E. Barrenechea, M. Pagola, and R. Orduna. Image thresholding computation using Atanassov's intuitionistic fuzzy sets. *J. Adv. Comput. Intell. Intell. Inf.* 11(2) (2007) 187–194.
- [53] H. Bustince, J. Kacprzyk, and V. Mohedano. Intuitionistic fuzzy generators. Application to intuitionistic fuzzy complementation. *Fuzzy Sets Syst.* 114 (2000) 485–504.
- [54] E. Barrenechea. *Image Processing with Interval-Valued Fuzzy Sets. Edge Detection*. Contrast, Ph.D. Thesis. Universidad Publica de Navarra, 2005.
- [55] F. Herrera and L. Martinez. A model based on linguistic 2-tuples for dealing with multigranularity hierarchical linguistic context in multiexpert decision-making. *IEEE Trans. Syst. Man Cybern.* 31(2) (2001) 227–234.
- [56] F. Liu, H. Geng, and Y.-Q. Zhang. Interactive fuzzy interval reasoning for smart web shopping. *Appl. Soft Comput.* 5(4) (2005) 433–439.
- [57] R. Sepulveda, O. Castillo, P. Melin, A. Rodriguez-Diaz, and O. Montiel. Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic. *Inf. Sci.* 177 (2007) 2023–2048.

- [58] A.M. Norwich and I.B. Turksen. The construction of membership functions. In: R.R. Yager (ed), *Fuzzy Sets and Possibility Theory*. Pergamon, New York, 1982, pp. 61–67.
- [59] I.B. Turksen. Type 2 representation and reasoning for CWW. *Fuzzy Sets Syst.* 127 (2002) 17–36.
- [60] H. Bustince. Indicator of inclusion grade for interval-valued fuzzy sets. Application to approximate reasoning based on interval-valued fuzzy sets. *Int. J. Approx. Reason.* 23(3) (2000) 137–209.
- [61] P. Burillo and H. Bustince. Entropy on intuitionistic fuzzy sets and on interval-valued fuzzy sets. *Fuzzy Sets Syst.* 78 (1996) 305–316.
- [62] H. Bustince, E. Barrenechea, and V. Mohedano. Intuitionistic fuzzy implication operators: An expression and main properties. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 12(3) (2004) 387–406.
- [63] H. Bustince, V. Mohedano, E. Barrenechea, and M. Pagola. A study of the intuitionistic fuzzy S-implication operators. In: E. Herrera-Viedma (ed), *Procesos de Toma de Decisiones, Modelado y Agregación de Preferencias (TIC-2002-11492-E)*, Granada, Spain, 2005, pp. 141–151.
- [64] H. Bustince and P. Burillo. Interval-valued fuzzy relations in a set structure. *J. Fuzzy Math.* 4(4) (1996) 765–785.
- [65] H. Bustince and P. Burillo. Mathematical analysis of interval-valued fuzzy relations: Application to approximate reasoning. *Fuzzy Sets Syst.* 113 (2000) 205–219.
- [66] H. Bustince and P. Burillo. Structures on intuitionistic fuzzy relations. *Fuzzy Sets Syst.* 78 (1996) 293–303.
- [67] T. Bilgiç. Interval-valued preference structures. *Eur. J. Oper. Res.* 105 (1998) 162–183.
- [68] O. Castillo and P. Melin. Fuzzy logic for plant monitoring and diagnostics. In: *Proceedings of IEEE International Conference on Fuzzy Systems*, Budapest, Hungary, 2004, pp. 25–29.
- [69] J. Chiang. Fuzzy linear programming based on statistical confidence interval and interval-valued fuzzy set. *Eur. J. Oper. Res.* 129 (2001) 65–86.
- [70] S.M. Chen, W.H. Hsiao, and W.T. Jong. Bidirectional approximate reasoning based on interval-valued fuzzy sets. *Fuzzy Sets Syst.* 91 (1997) 339–353.
- [71] S.M. Chen. Measures of similarity between vague sets. *Fuzzy Sets Syst.* 74 (1995) 217–223.
- [72] C. Cornelis, G. Deschrijver, and E. Kerre. Classification of intuitionistic fuzzy implicators: An algebraic approach. In: *Proceedings of 6th Joint Conference on Information Sciences*, Research Triangle Park, North Carolina, USA, 2002, pp. 105–108.
- [73] C. Cornelis, G. Deschrijver, and E.E. Kerre. Implication in intuitionistic fuzzy and interval-valued fuzzy set theory: Construction, classification, application. *Int. J. Approx. Reason.* 35 (2004) 55–95.
- [74] C. Cornelis, G. Deschrijver, and E. Kerre. Intuitionistic fuzzy connectives revisited. In: *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, IPMU'02, Annecy, France, 2002, pp. 1839–1844.
- [75] C. Cornelis and E.E. Kerre. Inclusion measures in intuitionistic fuzzy set theory. *Lect. Notes in Computer Science (Subseries LNAI)*, Vol. 2711. Springer Verlag, Berlin, Germany, 2003, pp. 345–356.
- [76] S. Cubillo and E. Castineira. Contradiction in intuitionistic fuzzy sets. In: *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, IPMU'04, Perugia, Italy, 2004, pp. 2180–2186.
- [77] G. Deschrijver, C. Cornelis, and E.E. Kerre. On the representation of intuitionistic fuzzy T-norms and T-conorms. *IEEE Trans. Fuzzy Syst.* 12(1) (2004) 45–61.
- [78] G. Deschrijver. The Archimedean property for t-norms in interval-valued fuzzy set theory. *Fuzzy Sets Syst.* 157 (2006) 2311–2327.
- [79] G. Deschrijver and E.E. Kerre. Uninorms in L*-fuzzy set theory. *Fuzzy Sets Syst.* 148(2) (2004) 243–262.
- [80] G. Deschrijver and E.E. Kerre. On the composition of intuitionistic fuzzy relations. *Fuzzy Sets Syst.* 136(3) (2003) 333–361.
- [81] S. Dick, A. Schenker, W. Pedrycz, and A. Kandel. Regranulation: A granular algorithm enabling communication between granular worlds. *Inf. Sci.* 177 (2007) 408–435.
- [82] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, 1980.
- [83] E.P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*. Kluwer, Dordrecht, 2002.
- [84] E.P. Klement, R. Mesiar, and E. Pap. Triangular norms. Position paper I: Basic analytical and algebraic properties. *Fuzzy Sets Syst.* 143(1) (2004) 5–26.
- [85] E.P. Klement, R. Mesiar, and E. Pap. Triangular norms. Position paper II: General constructions and parametrized families. *Fuzzy Sets Syst.* 145(3) (2004) 411–438.
- [86] E.P. Klement, R. Mesiar, and E. Pap. Triangular norms. Position paper III: Continuous t-norms. *Fuzzy Sets Syst.* 145(3) (2004) 439–454.
- [87] P. Smets and P. Magrez. Implication in fuzzy logic. *Int. J. Approx. Reason.* 1 (1987) 327–347.
- [88] M. Mas, M. Monserrat, J. Torrens, and E. Trillas. A survey on fuzzy implication functions. *IEEE Trans. Fuzzy Syst.* 15(6) (2007) 1107–1121.

- [89] M. Mas, M. Monserrat, and J. Torrens. On two types of discrete implications. *Int. J. Approx. Reason.* 40(3) (2005) 262–279.
- [90] P. Grzegorzewski. Distances between intuitionistic fuzzy sets and/or interval-valued fuzzy sets based on the Hausdorff metric. *Fuzzy Sets Syst.* 95(1) (1998) 113–117.
- [91] E. Szmidt and J. Kacprzyk. Entropy for intuitionistic fuzzy sets. *Fuzzy Sets Syst.* 118(3) (2001) 467–477.
- [92] E. Szmidt and J. Kacprzyk. Entropy and similarity of intuitionistic fuzzy sets. In: *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, IPMU'06, Paris, France, 2006, pp. 2375–2382.
- [93] X. Liu. Entropy, distance measure and similarity measure of fuzzy sets and their relations. *Fuzzy Sets Syst.* 52 (1992) 305–318.
- [94] H.B. Mitchell. Pattern recognition using type II fuzzy sets. *Inf. Sci.* 170 (2005) 409–418.
- [95] H. Rezaei and M. Mukaidono. New similarity measures of intuitionistic fuzzy sets. *J. Adv. Comput. Intell. Intell. Inf.* 11(2) (2007) 202–209.
- [96] I.K. Vlachos and G.D. Sergiadis. Subsethood, entropy, and cardinality for interval-valued fuzzy sets an algebraic derivation. *Fuzzy Sets Syst.* 158 (2007) 1384–1396.
- [97] C. Zhang and H. Fu. Similarity measure on three kinds of fuzzy sets. *Pattern Recognit. Lett.* 27(12) (2006) 1307–1317.
- [98] W. Zeng and H. Li. Relationship between similarity measures and entropy of interval valued fuzzy sets. *Fuzzy Sets Syst.* 157(11) (2006) 1477–1484.
- [99] P. Grzegorzewski and E. Mrowka. On the entropy of intuitionistic fuzzy sets and interval-valued fuzzy sets. In: *Proceedings of 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, IPMU'04, Perugia, Italy, 2004, pp. 1419–1426.
- [100] W.-L. Hung and M.-S. Yang. Similarity measures of intuitionistic fuzzy sets based on Hausdorff metric. *Pattern Recognit. Lett.* 25 (2004) 1603–1611.
- [101] D. Li and C. Cheng. New similarity measures of intuitionistic fuzzy sets and application to pattern recognition. *Pattern Recognit. Lett.* 23 (2002) 221–225.
- [102] Z. Liang and P. Shi. Similarity measures on intuitionistic fuzzy sets. *Pattern Recognit. Lett.* 24 (2003) 2687–2693.
- [103] H.B. Mitchell. On the Dengfeng-Chuntian similarity measure and its application to pattern recognition. *Pattern Recognit. Lett.* 24 (2003) 3101–3104.
- [104] E. Szmidt and J. Kacprzyk. Similarity of intuitionistic fuzzy sets and the Jaccard coefficient. In: *Proceedings of 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, IPMU'04, Perugia, Italy, 2004, pp. 1405–1412.
- [105] E. Szmidt, J. Kacprzyk. A measure of similarity for intuitionistic fuzzy sets. In: *Proceedings of 3th International Conference in Fuzzy Logic and Technology*, Zittau, Germany, 2003, pp. 206–209.
- [106] G.-J. Wang and X.-P. Li. On the IV-fuzzy degree and the IV-similar degree of IVFS and their integral representation. *J. Eng. Math.* 21 (2004) 195–201.
- [107] A. Kehagias and M. Konstantinidou. L-fuzzy valued inclusion measure, L-fuzzy similarity and L-fuzzy distance. *Fuzzy Sets Syst.* 136 (2003) 313–332.
- [108] T. Gerstenkorn and J. Manko. Correlation of intuitionistic fuzzy sets. *Fuzzy Sets Syst.* 44 (1991) 39–43.
- [109] G. Wang and X. Li. Correlation and information energy of interval-valued fuzzy numbers. *Fuzzy Sets Syst.* 103 (1999) 169–175.
- [110] I.B. Turksen. Interval-valued strict preference with Zadeh triples. *Fuzzy Sets Syst.* 78 (1996) 183–195.
- [111] L.J. Kohout and W. Bandler. Fuzzy interval inference utilizing the checklist paradigm and BK-relational products. In: R.B. Kearfort and V. Kreinovich (eds), *Application of Interval Computations*. Kluwer, Dordrecht, 1996, pp. 291–335.
- [112] L.A. Zadeh. Theory of approximate reasoning. In: J. Hayes, D. Michie, and L.I. Mikulich (eds), *Machine Intelligence*. Halstead Press, New York, 1979, pp. 149–194.
- [113] M.K. Roy and R. Biswas. I-v fuzzy relations and Sanchez's approach for medical diagnosis. *Fuzzy Sets Syst.* 47 (1992) 35–38.
- [114] M. Wagenknecht. On transitive solutions of fuzzy equations, inequalities and lower approximations of fuzzy relations. *Fuzzy Sets Syst.* 75(2) (1995) 229–240.
- [115] W. Pedrycz. Processing in relational structures: Fuzzy relational equations. *Fuzzy Sets Syst.* 40(1) (1991) 77–106.
- [116] T. Arnould and S. Tano. Interval-valued fuzzy backward reasoning. *IEEE Trans. Fuzzy Syst.* 3(4) (1995) 425–437.

- [117] S. Fukami, M. Mizumoto, and K. Tanaka. Some considerations on fuzzy conditional inference. *Fuzzy Sets Syst.* 4 (1980) 243–273.
- [118] J.F. Baldwin, B.W. Pilsworth. Axiomatic approach to implication for approximate reasoning with fuzzy logic. *Fuzzy Sets Syst.* 3 (1980) 193–219.
- [119] A. Sala, B. Tormos, V. Macià, and E. Royo. Fuzzy diagnosis module based on interval fuzzy logic: Oil analysis application. In: *Proceedings of the International Conference on Informatics in Control, Automation and Robotics*, ICINCO 2005, Barcelona, Spain, 2005, pp. 85–90.
- [120] S. Wang, F.-L. Chung, Y.Y. Li, D. Hu, and X.S. Wu. A new Gaussian noise filter based on interval type-2 fuzzy logic systems. *Soft Comput.* 9(5) (2005) 398–406.
- [121] L.A. Zadeh. From computing with numbers to computing with words – from manipulation of measurements to manipulation of perceptions. *IEEE Trans. Circuits Syst.* 4 (1999) 105–119.
- [122] I.B. Turksen. Meta-linguistic axioms as foundation for computing with words. *Inf. Sci.* 177 (2006) 332–359.
- [123] J.M. Mendel. Computing with words and its relationships with fuzzistics. *Inf. Sci.* 177 (2007) 988–1006.
- [124] F. Herrera, L. Martínez, and P.J. Sanchez. Managing non-homogeneous information in group decision making. *Eur. J. Oper. Res.* 166(1) (2005) 115–132.
- [125] L. Martínez, J. Liu, Da Ruan, and J.-B. Yang. Dealing with heterogeneous information in engineering evaluation processes. *Inf. Sci.* 177 (2007) 1533–1542.
- [126] A. Pankowska and M. Wygralak. On hesitation degrees in IF-set theory. In: L. Rutkowski, J. Siekmann, and R. Tadeusiewicz (eds), *Artificial Intelligence and Soft Computing, Lecture Notes in Artificial Intelligence*, Vol. 3070, Springer-Verlag, Berlin, Germany, 2004, pp. 338–343.
- [127] A. Pankowska and M. Wygralak. General IF-sets with triangular norms and their applications to group decision making. *Inf. Sci.* 176 (2006) 2713–2754.
- [128] E. Szmidt and J. Kacprzyk. Group decision making under intuitionistic fuzzy preference relations. In: *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, IPMU'98, Paris, France, 1998, pp. 172–178.
- [129] H.-F. Wang and M.-L. Wang. A fuzzy multiobjective linear programming. *Fuzzy Sets Syst.* 86(1) (1997) 61–72.
- [130] A. Sergueeva and J. Hunter. Fuzzy interval methods in investment risk appraisal. *Fuzzy Sets Syst.* 142(3) (2004) 443–466.
- [131] J.S. Yao and T.S. Shih. Fuzzy revenue for fuzzy demand quantity based on interval-valued fuzzy sets. *Comput. Oper. Res.* 29 (2002) 1495–1535.
- [132] H. Hagras. A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Trans. Fuzzy Syst.* 12(4) (2004) 524–539.
- [133] H.T. Nguyen, V. Kreinovich, R.N. Lea, and D. Tolbert. How to control if even experts are not sure: Minimizing intervals of uncertainty. In: *Abstracts of Workshop on Interval Methods, International Conference on Interval Computations*, Lafayette, Louisiana, 1993, p. 27.
- [134] K.C. Wu. Fuzzy interval control of mobile robots. *Comput. Electr. Eng.* 22(3) (1996) 211–229.
- [135] W. Pedrycz. Fuzzy modelling: Fundamentals, construction and evaluation. *Fuzzy Sets Syst.* 41(1) (1991) 1–15.
- [136] W. Pedrycz. Relevancy of fuzzy models. *Inf. Sci.* 52(3) (1990) 285–302.
- [137] W. Pedrycz. Direct and inverse problem in comparison of fuzzy data. *Fuzzy Sets Syst.* 34(2) (1990) 223–235.
- [138] L.A. Zadeh. Fuzzy sets and information granularity. In: M. Gupta, R. Ragade, and R. Yager (eds), *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, 1979, pp. 3–18.
- [139] L.A. Zadeh. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 19 (1996) 103–111.
- [140] Y.Y. Yao. Granular Computing: basic issues and possible solutions. In: *Proceedings of 5th Joint Conference on Information Sciences*, Cary, North Carolina, USA, 2000, pp. 186–189.
- [141] A. Bargiela, W. Pedrycz. *Granular Computing: An Introduction*. Kluwer Academic Publishers, Boston, MA, 2003.

23

Measurement Theory and Uncertainty in Measurements: Application of Interval Analysis and Fuzzy Sets Methods

Leon Reznik

23.1 Introduction

Measurement is one of the most fundamental processes in science and technology. It is commonly defined [1] as the process of gathering information from the physical world, which is achieved by means of measuring instruments. This definition considers the measurement as an empirical process [2] of acquiring information about an object. A measure of an object's property gives us an ability to express facts and conventions about it in the formal language of mathematics and in creating various models representing the systems and processes under measurement. Despite the variety of models used, none of them is ideal as no model could be. The reasons of imperfection are various: an approximate definition of the measurand,¹ a limited knowledge of an environmental context, and a variability of influence factors. These effects lead to shifts of the values and/or to fluctuations of the values, i.e. uncertainties. To represent and process these uncertainties, the most popular methods, which are currently in place, are the probability theory [4] and the interval (or uncertainty) calculus [5].

Any measurement procedure results in providing quantitative characteristics of the object or process under measurement. The measurement science has developed a unanimously accepted approach, which recognizes an importance of providing the characteristics of the measurement result's uncertainty along with the value of the result itself. According to the modern measurement science, whose basic concepts are briefly reviewed in Section 23.2, any measurement procedure produces some sort of results' distribution that includes the true result's location. This way, the measurement could be considered as a process of producing some sort of a granule (see Figure 23.1). An expression of the measurement result should give characteristics of this granule. Concurrently, granulation is considered as a substantial feature of the cognition process, which could be attributed to the bounded ability of sensors, both human organs and

¹ Measurand is a physical parameter being quantified by measurement [3].

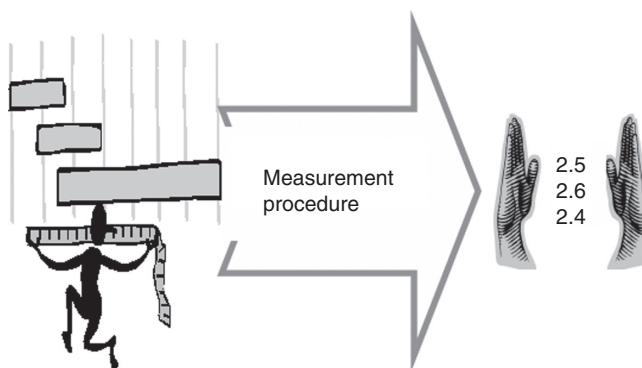


Figure 23.1 Measurement procedure as a process of granulation of information from the object under measurement

technical instruments, to resolve details and store information. L. Zadeh [6] points out that any granulation involves clumping, with a granule being a clump of attribute values drawn together by indistinguishability, similarity, proximity, or functionality. The concepts of information granulation and granular computing have been proposed and researched for some time with a huge contribution into this field made by pioneer work of Witold Pedrycz [7–10]. Also he investigated and designed a few granular computing applications in signal processing and measurement systems [9] and linguistic modeling [10], which is an important part of any measurement procedure.

Other aspects of measurement result modeling are considered in Section 23.3. From this section one may better appreciate the idea of granulation for formalizing and further processing measurement results in modern science and technology. Having made this conclusion, we will concentrate on the description of the measurement procedures, which are currently in place (see Sections 23.4 and 23.5) or have a good prospective to be adopted in a near future (Sections 23.6 and 23.7).

23.2 Measurement Science: Basic Concepts

The modern measurement science, which has been developed over a few centuries of extensive research based on both theoretical investigation and practical work in various fields, commonly includes the following basic ideas and concepts [11]:

1. A measurement produces an estimate of a parameter of an abstract mental model that is supposed to represent reality. It means that before even starting the measurement process some a priori created model of the object or a process under measurement had to exist (see Figure 23.2).
2. The model must be valid for all cases of interest. The quality of a particular model can be judged by observing the fidelity of the model compared to the real physical output for a variety of different inputs.
3. Every model is incomplete. Some model parameters may actually represent several real parameters, either because that is seemed suitable or because the correct model is too difficult to understand. There can be parameters that appear to be distinct and unrelated, but are actually parts of a single parameter, in which case the parts cannot be measured separately unless additional data are collected to resolve the contribution of each part. Sometimes a particular parameter may not even be measurable, because its value has no apparent effect on the output of the model. Even the topology of the model may be wrong in some way. Since the model is only partially known, approximations and uncertainty are a commonplace.
4. Parameter estimates vary in accuracy. For any given set of observation data, some parameter values can generally be estimated with more accuracy than others, because some parameters have more effect on the model output than others. In addition, some parameters may be affected by noise or interference

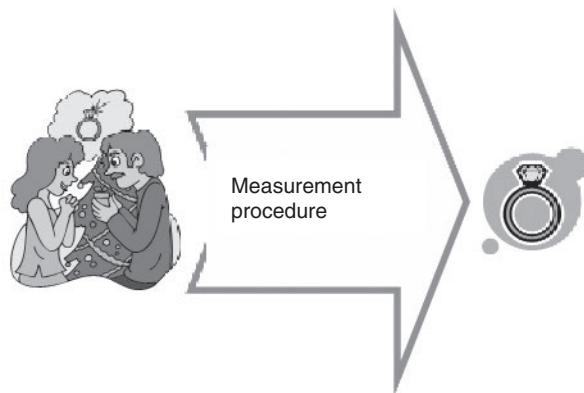


Figure 23.2 Measurement procedure as a process of an initial model parameter's estimation

more than others, depending on the locations of the various sources of contamination in the system and on a particular method or algorithm used in making the measurement.

5. Some model parameters can be measured explicitly, like a car speed, while others can only be implied because they are buried deeply in the model and are visible only via their effects on the output.
6. The values of implicit model parameters can only be estimated by simultaneously adjusting all parameter values until the best fit is obtained between the physical observations and the model predictions. It is generally not feasible to adjust one parameter at a time, because the parameters usually interact with one another.

From the concepts given above the first thing which one may draw up is that there are a lot of factors causing the constant changes in the measurements. The most influential reasons for those changes are fluctuations in the measured attributes of the system or a process under measurement, imperfection of the measuring instruments, and incompleteness of the models applied in measurement. This theoretical consideration is confirmed by a measurement practice: whenever one implements any measurement procedure, one never gets measurement results exactly the same as before. These perturbations may be within the system under measurement or they may be introduced when one tries to make observations of the system under measurement. Each measurement procedure will produce a set of observations or measurement results which are supposed to characterize a measured value. Those observations are tied together by an assumption that they represent a certain attribute of the system under measurement at a particular time moment. Altogether they compose a clump of values, which are related to each other by the fact that they were born by the same measurement procedure at supposedly the same time.

23.3 Measurement Uncertainty Formalization: Generalized Theory and Granulation

A measured parameter value expressed as a single number has little meaning and conveys only a minuscule amount of information when standing alone, unless many similar measurements have already been made or unless a careful error analysis of the measurement model has been completed [11]. All measurement results are associated with an uncertainty. The uncertainty of the measurement result reflects the lack of exact knowledge of the specified measurand. The international [12] and national standards request providing some quantitative indication of the quality of the measurement result along with the result itself, so those who use it can assess its reliability. The international guide [12] defines uncertainty

(Section D.5.2) as an expression of the fact that for a given measurand and a given result of measurement of it, there is not one value but an infinite number of values dispersed about the result that are consistent with all of the observations and data and one's knowledge of the physical world, and with varying degrees of credibility can be attributed to the measurand (Figure 23.1). This approach substantiates the description of the measurement result as a granule, which includes both the measurand value and its uncertainty characteristics.

As a result of any measurement procedure some characteristics of a measurement granule need to be provided. These characteristics describe the uncertainty of the measurement result and at the same time describe the measurement granule. Larger uncertainty corresponds to bigger granules. As each measurement granule represents the value of some attributes of the system or a process under measurement, its characteristics need to be properly dealt with at further steps of information processing.

From the information point of view, any measurement procedure delivers a new one or modifies a previously existed description of some properties of the system under measurement. This description or a model (see Figure 23.2) could be considered as a constraint on possible values, which the property under question may have. The measurement uncertainty could be dealt with under the generalized theory of uncertainty (GTU), which is outlined in L. Zadeh's paper [13]. A fundamental premise of GTU is that information, whatever its form, may be represented as what is called a generalized constraint. A generalized constraint is a constraint of the form $X \text{ is } r R$, where X is the constrained variable, R is a constraining relation, generally non-bivalent, and r is an indexing variable, which identifies the modality of the constraint, i.e., its semantics. A generalized constraint could be described with various formalization models. Under certain circumstances, a combination of models could be employed. The principal constraints are possibilistic (r -blank), probabilistic ($r = p$), veristic ($r = v$), usuality ($r = u$), random set ($r = rs$), fuzzy graph ($r = fg$), bimodal ($r = bm$), and group ($r = g$). Generalized constraints may be qualified, combined, and propagated. Depending on the constraint formalization choice, various approaches could be applied in formalizing and handling measurement uncertainty. In the sections below the most widely used probabilistic (statistical) and fuzzy (interval) approaches are described in greater detail.

23.4 Measurement Uncertainty Description: International and U.S. Perspectives

Over the years, many different approaches to evaluating and expressing the uncertainty of measurement results have been developed and adopted. Because of this lack of international agreement on the expression of uncertainty in measurement, in 1977 the International Committee for Weights and Measures (CIPM), the world's highest authority in the field of measurement science, asked the International Bureau of Weights and Measures (BIPM), to address the problem in collaboration with the various national metrology institutes and to propose a specific recommendation for its solution. This led to the development of Recommendation INC-1 (1980) by the Working Group on the Statement of Uncertainties convened by the BIPM (see Figure 23.3), a recommendation that the CIPM approved in 1981 and reaffirmed in 1986 via its own Recommendations I (CI-1981) and II (CI-1986) which are given in Figure 23.3 [12].

The group work resulted in producing the 100-page *Guide to the Expression of Uncertainty in Measurement* [12] (or GUM as it is now often called). It was published in 1993 (corrected and reprinted in 1995) by ISO in the name of the seven international organizations that supported its development in ISO/TAG 4. Moreover, the GUM has been adopted by the U.S. National Institute of Standards and Technology (NIST) and most of NIST's sister national metrology institutes throughout the world, such as the National Research Council in Canada, the National Physical Laboratory in the United Kingdom, and the Physikalisch-Technische Bundesanstalt in Germany. Most recently, the GUM has been adopted by the American National Standards Institute (ANSI) as an American National Standard. Its official designation is ANSI/NCSL Z540-2-1997 and its full title is *American National Standard for Expressing Uncertainty – U.S. Guide to the Expression of Uncertainty in Measurement*.

1. The uncertainty in the result of a measurement generally consists of several components which may be grouped into two categories according to the way in which their numerical value is estimated.

Type A. Those which are evaluated by statistical methods.

Type B. Those which are evaluated by other means.

There is not always a simple correspondence between the classification into categories A or B and the previously used classification into ‘random’ and ‘systematic’ uncertainties. The term ‘systematic uncertainty’ can be misleading and should be avoided.

Any detailed report of uncertainty should consist of a complete list of the components, specifying for each the method used to obtain its numerical value.

2. The components in category A are characterized by the estimated variances s_i^2 (or the estimated ‘standard deviations’ s_i) and the number of degrees of freedom v_i . Where appropriate, the covariances should be given.

3. The components in category B should be characterized by quantities u_j^2 , which may be considered approximations to the corresponding variances, the existence of which is assumed. The quantities u_j^2 may be treated like variances and the quantities u_j like standard deviations. Where appropriate, the covariances should be treated in a similar way.

4. The combined uncertainty should be characterized by the numerical value obtained by applying the usual method for the combination of variances. The combined uncertainty and its components should be expressed in the form of ‘standard deviations’.

5. If, for particular applications, it is necessary to multiply the combined uncertainty by an overall uncertainty, the multiplying factor must always be stated.

The above recommendation, INC-1 (1980), is a brief outline rather than a detailed prescription. Consequently, the CIPM asked the International Organization for Standardization (ISO) to develop a detailed guide based on the recommendation because ISO could more easily reflect the requirements stemming from the broad interests of industry and commerce.

Figure 23.3 Recommendation INC-1 (1980) expression of experimental uncertainties

23.5 Measurement Procedures and Uncertainty Evaluation: Standard Definitions and Models – Probabilistic Approach

In this section we will be following up the definitions, explanations, and examples given in the GUM. All the measurements could be classified into two big groups: direct and indirect measurements. In direct measurement the quantity under measurement is directly assessable for measurement and the system under measurement is made to interact with a measuring instrument. The value of the quantity Y being measured, called the measurand, is read from or provided directly by the measuring instrument [14]. In many industrial applications the measurand is not measured directly, but is determined from N other quantities X_1, X_2, \dots, X_N through a functional relation f , often called the *measurement equation*:

$$Y = f(X_1, X_2, \dots, X_N). \quad (1)$$

Included among the quantities X_i are corrections (or correction factors), as well as quantities that take into account other sources of variability, such as different observers, instruments, samples, laboratories, and times at which observations are made (e.g., different days). Thus, the function f of equation (1) should express not simply a physical law but a measurement process, and in particular, it should contain all quantities that can contribute a significant uncertainty to the measurement result.

An estimate of the measurand or *output quantity* Y , denoted by y , is obtained from equation (1) using *input estimates* x_1, x_2, \dots, x_N for the values of the N *input quantities* X_1, X_2, \dots, X_N . Thus the *output estimate* y , which is the result of the measurement, is given [12, section 4.1.4] by

$$y = f(x_1, x_2, \dots, x_N). \quad (2)$$

For example, as pointed out in the GUM, Section 4.1.1, if a potential difference V is applied to the terminals of a temperature-dependent resistor that has a resistance R_0 at the defined temperature t_0 and a linear temperature coefficient of resistance α , the power P (the measurand) dissipated by the resistor

at the temperature t depends on V , R_0 , α , and t according to the formula (3):

$$P = f(V, R_0, \alpha, t) = V^2/R_0[1 + \alpha(t - t_0)]. \quad (3)$$

Classification of Uncertainty Components

The uncertainty of the measurement result y arises from the uncertainties $u(x_i)$ (or u_i for brevity) of the input estimates x_i that enter equation (2). In the example of equation (3), the uncertainty of the estimated value of the power P arises from the uncertainties of the estimated values of the potential difference V , resistance R_0 , temperature coefficient of resistance b , and temperature t . In general, components of uncertainty may be categorized according to the method used to evaluate them:

- Type A evaluation method of evaluation of uncertainty by the statistical analysis of series of observations, and
- Type B evaluation method of evaluation of uncertainty by means other than the statistical analysis of series of observations.

Representation and Evaluation of Different Uncertainty Components Types

Standard Uncertainty

Each component of uncertainty, however evaluated, is represented by an estimated standard deviation, termed standard uncertainty with suggested symbol u_i and is equal to the positive square root of the estimated variance.

Standard Uncertainty: Type A

An uncertainty component obtained by a Type A evaluation is represented by a statistically estimated standard deviation s_i , equal to the positive square root of the statistically estimated variance s_i^2 , and the associated number of degrees of freedom v_i . For such a component the standard uncertainty is $u_i = s_i$.

A Type A evaluation of standard uncertainty may be based on any valid statistical method for treating data. Examples are calculating the standard deviation of the mean of a series of independent observations; using the method of least squares to fit a curve to data in order to estimate the parameters of the curve and their standard deviations; and carrying out an analysis of variance in order to identify and quantify random effects in certain kinds of measurements.

Mean and standard deviation: As an example of a Type A evaluation, consider an input quantity X_i whose value is estimated from n independent observations $X_{i,k}$ of X_i obtained under the same conditions of measurement. In this case the input estimate x_i is usually the sample mean

$$x_i = \bar{X}_i = \frac{1}{n} \sum_{k=1}^n X_{i,k} \quad (4)$$

and the standard uncertainty $u(x_i)$ to be associated with x_i is the estimated standard deviation of the mean

$$u(x_i) = s(\bar{X}_i) = \sqrt{\frac{1}{n(n-1)} \sum_{k=1}^n (X_{i,k} - \bar{X}_i)^2}. \quad (5)$$

Standard Uncertainty: Type B

In a similar manner, an uncertainty component obtained by a Type B evaluation is represented by a quantity u_j , which may be considered an approximation to the corresponding standard deviation; it is equal to the square root of u_j^2 , which may be considered an approximation to the corresponding variance and which is obtained from an assumed probability distribution based on all the available information. Since the quantity u_j^2 is treated like a variance and u_j like a standard deviation, for such a component the standard uncertainty is simply u_j .

A Type B evaluation of standard uncertainty is usually based on scientific judgment using all of the relevant information available, which may include the following:

- previous measurement data;
- experience with, or general knowledge of, the behavior and property of relevant materials and instruments;

- manufacturer's specifications;
- data provided in calibration and other reports; and
- uncertainties assigned to reference data taken from handbooks.

Below are some examples from [12] of Type B evaluations in different situations, depending on the available information and the assumptions of the experimenter. Broadly speaking, the uncertainty is either obtained from an outside source or obtained from an assumed distribution:

1. Uncertainty obtained from an outside source

(a) Multiple of a standard deviation

Procedure: Convert an uncertainty quoted in a handbook, manufacturer's specification, and calibration certificate, which is a stated multiple of an estimated standard deviation to a standard uncertainty by dividing the quoted uncertainty by the multiplier.

(b) Confidence interval

Procedure: Convert an uncertainty quoted in a handbook, manufacturer's specification, calibration certificate, etc., which defines a 'confidence interval' having a stated level of confidence, such as 95% or 99%, to a standard uncertainty by treating the quoted uncertainty as if a normal probability distribution had been used to calculate it (unless otherwise indicated) and dividing it by the appropriate factor for such a distribution. These factors are 1.960 and 2.576 for the two levels of confidence given.

2. Uncertainty obtained from an assumed distribution

(a) Normal distribution: '1 out of 2'

Procedure: Model the input quantity in question by a normal probability distribution and estimate lower and upper limits $m - \sigma$ and $m + \sigma$ such that the best estimated value of the input quantity is m (i.e., the center of the limits) and there is one chance out of two (i.e., a 50% probability) that the value of the quantity lies in the interval $m - \sigma$ and $m + \sigma$. Then, u_j is approximately 1.48σ , where σ is the half-width of the interval.

(b) Normal distribution: '2 out of 3'

Procedure: Model the input quantity in question by a normal probability distribution and estimate lower and upper limits $m - \sigma$ and $m + \sigma$ such that the best estimated value of the input quantity is m (i.e., the center of the limits) and there are two chances out of three (i.e., a 67% probability) that the value of the quantity lies in the interval $m - \sigma$ and $m + \sigma$. Then, u_j is approximately σ .

(c) Normal distribution: '99.73%'

Procedure: If the quantity in question is modeled by a normal probability distribution, there are no finite limits that will contain 100% of its possible values. However, plus and minus 3 standard deviations about the mean of a normal distribution corresponds to 99.73% limits. Thus, if the limits $m - \sigma$ and $m + \sigma$ of a normally distributed quantity with mean m are considered to contain 'almost all' of the possible values of the quantity, i.e., approximately 99.73% of them, then u_j is approximately $\sigma/3$.

(d) Uniform (rectangular) distribution

Procedure: Estimate lower and upper limits $m - \sigma$ and $m + \sigma$ for the value of the input quantity in question such that the probability that the value lies in the interval is, for all practical purposes, 100%. Provided that there is no contradictory information, treat the quantity as if it is equally probable for its value to lie anywhere within the interval, i.e., model it by a uniform (i.e., rectangular) probability distribution. The best estimate of the value of the quantity is then m with $u_j = \sigma$ divided by the square root of 3, where σ is the half-width of the interval.

(e) Triangular distribution

The rectangular distribution is a reasonable default model in the absence of any other information. But if it is known that values of the quantity in question near the center of the limits are more likely than values close to the limits, a normal distribution or, for simplicity, a triangular distribution, may be a better model.

Procedure: Estimate lower and upper limits $m - \sigma$ and $m + \sigma$ for the value of the input quantity in question such that the probability that the value lies in the interval $m - \sigma$ and $m + \sigma$ is, for all

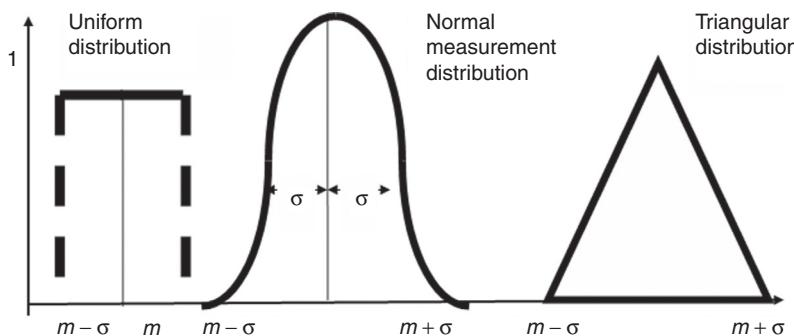


Figure 23.4 Illustration of different probability density distribution types: uniform (rectangular), normal (Gaussian), and triangular (used as normal approximation)

practical purposes, 100%. Provided that there is no contradictory information, model the quantity by a triangular probability distribution. The best estimate of the value of the quantity is then m with $u_j = \sigma$ divided by the square root of 6, where σ is the half-width of the interval.

In Figure 23.4 three examples of various probability distributions referenced above are given, where m is the expectation or mean of the distribution and σ is the standard deviation. For a normal distribution, $\pm\sigma$ area encompasses about 68% of the distribution; for a uniform distribution, $\pm\sigma$ area encompasses about 58% of the distribution; and for a triangular distribution, $\pm\sigma$ area encompasses about 65% of the distribution.

Summary information regarding measurement uncertainty classification in relation to the type of granulation is given in Table 23.1. Table 23.2 compares probabilistic and possibilistic approaches in measurement uncertainty modeling, summarizing their advantages and disadvantages.

23.6 Measurement Procedures and Uncertainty Evaluation: Perspectives and Feasibility of a Fuzzy Approach

The probability theory and mathematical statistics have been considered as a conventional framework for conducting the theoretical analysis and practical measurement uncertainty evaluation despite that even the definition of applied terms can be considered as fuzzy in some degree. Strictly speaking, the definition of the ‘true measurand value’ is very uncertain for some measurands, such as the cloudiness depth, the water temperature in the Atlantic Ocean, the temperature of moving vapor, and the tree branch radius. Some other values can be less fuzzy. Certainly, one can conclude that no measurement procedure can deliver absolutely certain physical values. This uncertainty, even at the definition level, makes it hard to understand a measurement error as a difference between the measurement result and the true measurand value. In a number of practical applications [15], a strictly statistical approach based on probabilistic models only does not satisfy the requirements.

The idea of measurement uncertainty formulation in terms of fuzzy systems theory looks rather reasonable. Some steps in this direction have already been made. In a number of the international and national standards, the term ‘measurement error’ has been replaced with the term ‘measurement uncertainty,’ which is a better fit to the fuzzy systems terminology.

Publications, criticizing the probabilistic models applied in measurement science, are now followed up by research results presenting models describing measurement uncertainty in fuzzy sets theory terms or combining both theories [16–22]. For example, in [21] a priori fuzzy information about the object under measurement is applied to increase the measurement accuracy and/or reliability.

In order to apply the fuzzy sets and systems methodology in theoretical metrology and measurement practice, one has to prove that this methodology is able to perform mathematical and logical operations

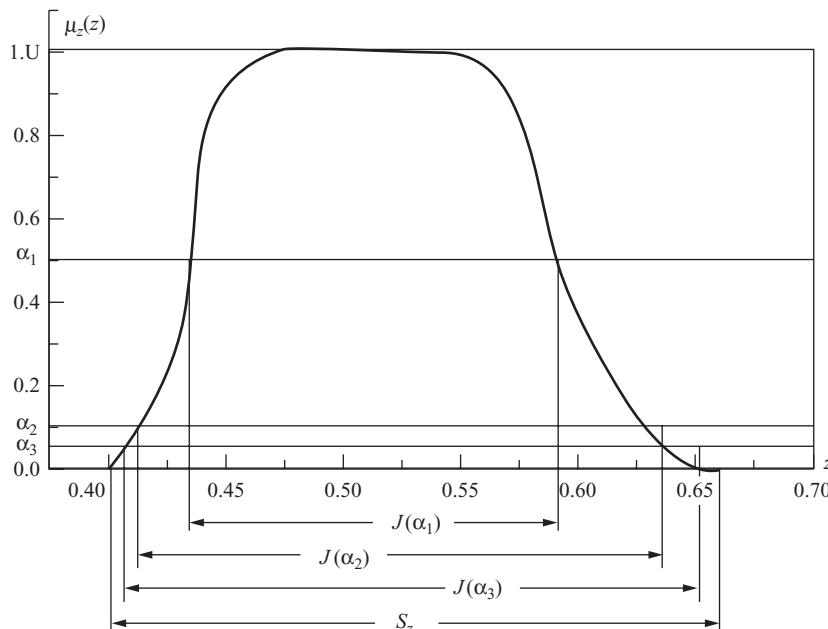
Table 23.1 Classification of uncertainty in measurement

	Type A uncertainty	Type B uncertainty
Uncertainty origins	Results deviation due to repeated measurement	Bias in measurement results due to manufacturing defects, lack of calibration, limitations of the methodology, and/or technology used in sensors
Method of evaluation	Based on statistics	Based on scientific judgments, which may include various methodologies
Main models used	Probability models	Probability and fuzzy (under investigation)
Data used in evaluation	Measurement (observation) results	<ul style="list-style-type: none"> -Previous measurement data, -Experience with, or general knowledge of, the behavior and property of relevant materials and instruments -Manufacturer's specifications -Data provided in calibration and other reports -Uncertainties assigned to reference data taken from handbooks
Uncertainty representation	A statistically estimated standard deviation s_i , equal to the positive square root of the statistically estimated variance s_i^2 , and the associated number of degrees of freedom v_i ; for such a component, the standard uncertainty is $u_i = s_i$	A quantity u_j , which may be considered an approximation to the corresponding standard deviation; it is equal to the positive square root of u_j^2 , which may be considered an approximation to the corresponding variance and which is obtained from an assumed probability distribution based on all the available information; since the quantity u_j^2 is treated like a variance and u_j like a standard deviation, for such a component the standard uncertainty is simply u_j
Examples of calculation	Consider an input quantity X_i whose value is estimated from n independent observations $X_{i,k}$ of X_i obtained under the same conditions of measurement. In this case, the input estimate x_i is usually the sample mean $x_i = \bar{X}_i = \frac{1}{n} \sum_{k=1}^n X_{i,k}$ and the standard uncertainty $u(x_i)$ to be associated with x_i is the estimated standard deviation of the mean $u(x_i) = s(\bar{X}_i) = \sqrt{\frac{1}{n(n-1)} \sum_{k=1}^n (X_{i,k} - \bar{X}_i)^2}$	<p>(a) Multiple of a standard deviation <i>Procedure:</i> Convert an uncertainty quoted in a handbook, manufacturer's specification and calibration certificate, which is a stated multiple of an estimated standard deviation to a standard uncertainty by dividing the quoted uncertainty by the multiplier</p> <p>(b) Confidence interval <i>Procedure:</i> Convert an uncertainty quoted in a handbook, manufacturer's specification, calibration certificate, etc., which defines a 'confidence interval' having a stated level of confidence, such as 95% or 99%, to a standard uncertainty by treating the quoted uncertainty as if a normal probability distribution had been used to calculate it (unless otherwise indicated) and dividing it by the appropriate factor for such a distribution; these factors are 1.960 and 2.576 for the two levels of confidence given</p>
Type of the generalized constraint determining the granulation	Probabilistic	Possibilistic, probabilistic

Table 23.2 Comparison of probabilistic and possibilistic model granules in measurement

	Probabilistic model	Random-fuzzy model	Possibilistic (fuzzy) model
Measuring function	Random function	Random function	Fuzzy function
Measurement results	Real numbers	Fuzzy intervals or membership function	
Advantage	<ul style="list-style-type: none"> -Procedures and calculation are well known; they are automated -Well suited for big samples; has been used in measurement procedures for a long time -Works well for Type A uncertainty evaluation 	<ul style="list-style-type: none"> -Could be better for Type B uncertainty evaluation -Could work better for small samples -Procedures are more universal and could cover a bigger variety of different cases 	
Disadvantage	<ul style="list-style-type: none"> -Not well suited for Type B uncertainty evaluation -May produce low-quality estimates for small samples 	<ul style="list-style-type: none"> -Novel procedures need to be developed and tested -Still a matter of research, not a practice -Professionals need to be trained how to apply 	

with fuzzy values, intervals and functions, typical for measurement science and practice. Any fuzzy set and variable can be determined completely with a membership function $\mu(z)$ whose each value is a numerical expression of a degree of possibility (belief, confidence, preference) of taking this particular value by the variable considered. The membership function is positive and may take values between 0 and 1 only. Some similarity can be found between a membership function and a probability density distribution function. However, the membership function for any particular variable or set may take different shapes and does not have to satisfy to the normality conditions. A fuzzy approach which is based on the application of a possibility theory [23] could be considered as some form of generalization of the interval approach as a fuzzy set can be interpreted as a nested stack of intervals (see Figure 23.5).

**Figure 23.5** Example of a membership function with α -cuts marked

On the other hand, the fuzzy set could be considered as an upper bound of probability distributions [17]. These two considerations place fuzzy interpretation of measurement uncertainty between probabilistic and interval analysis approaches.

As Mauris *et al.* [17] point out, the level of confidence $1 - \alpha$ is a lower bound of the probability that the measured value belongs to the interval J_α . This expression of knowledge is not equivalent to the definition of one single probability distribution and could be particularly suitable for the expression of Type B measurement uncertainty. As illustrated by many examples in the GUM, the Type B uncertainty is often provided as intervals set, each corresponding to particular level of confidence. Mauris *et al.* [17] conclude that building a fuzzy set out of this stack of intervals seems to be a more natural method than deriving the standard deviation, which requires a priori known probability distributions. Table 23.2 discusses advantages and disadvantages of probabilistic and possibilistic (fuzzy and interval) granules in both uncertainty-type modelings.

Fuzzy sets and variables as well as their membership functions can be applied to model a measurement result and its uncertainty. Urbanski and Wasowski [24] classify the existing models of the measurement uncertainty into the following groups:

1. *Statistical model (standard model of uncertainty):* The measuring function f is a random function and the measurement results X_i are real numbers (crisp numbers), but in this model the Type B component of inexactness may not be correctly represented.
2. *Fuzzy set model:* The measuring function is fuzzy, and the space Y of measured results are the fuzzy intervals J_Y characterized by membership function $\mu(x) = \mu_{J(x)}$. In this model,
 - (a) the result of single measurement is a fuzzy interval and the uncertainty of this measurement is described by a membership function;
 - (b) the mathematical operations performed on the measurement results are operations on fuzzy intervals. Arithmetic operations on fuzzy intervals could be defined by a variety of different ways, with the most suitable ones being the classical extension principle and t -norms.
3. *Random-fuzzy model:* The measuring function f is a random function, and the space Y of measurement results is the set of fuzzy intervals J . In this model the results of measurement are random-fuzzy intervals, and it is possible to define the extended uncertainty for a given significance level and degree of possibility as a half of α -cut width.

An interesting example of merging fuzzy sets, rough sets, and statistical expectation maximization (EM) algorithm in measurement and multispectral image segmentation is given in [25]. EM provides a statistical model of the data and handles the associated and representation uncertainties, while rough sets assist in faster convergence and in avoiding the local minima problem. While statistical methods allow handling uncertainties arising from both measurement errors and the presence of mixed pixels, rough and fuzzy sets are used for data analysis by constructing approximation sets of concepts or ‘information granules.’ The information granule formalizes the concept of finite-precision representation of objects in real-life situations. This chapter describes a technique of using rough sets to obtain an initial approximation of Gaussian mixture model parameters.

Over the last few years new applications implementing more complicated schemes of using fuzzy formalized expert information in measurement procedures have been published. Mahajan *et al.* [26] describe an intelligent fusion measurement system in which measurement data from different types of sensors with various resolutions are integrated and fused based on the confidence in them derived from information not usually used in data fusion, such as operating temperature, frequency range, fatigue cycles, etc. These are fed as additional inputs to a fuzzy inference system (FIS) that has predefined membership functions for each of these variables. The outputs of the FIS are weights that are assigned to the different sensor measurement data that reflect the confidence in the sensor’s behavior and performance. In [25] the fuzzy and interval analysis models of a two-dimensional navigation map and rough estimates of a robot position are applied to improve a robot guidance and navigation. Mauris *et al.* [27] aim at reproducing the linguistic evaluation of comfort perception provided by a human through aggregating of the relevant physical measurements such as temperature, humidity, and luminosity.

At the same time new emerging technology of wireless sensor networks (WSN) may provide a technological base for generation and communication of additional models. WSN are composed by many

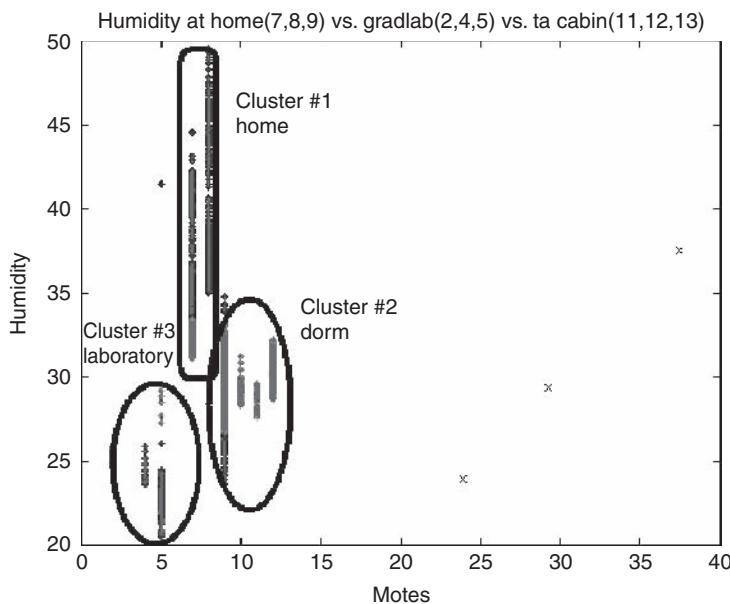


Figure 23.6 Measurements of humidity taken at by sensor nodes at different locations over a period of 2 days

cheap sensor nodes, each of which usually includes a few sensors that may communicate and transmit information between each other and with other processors called base stations. In order to accomplish this task sensor nodes are supposed to be deployed in close proximity to each other. The energy supply of the sensor nodes is one of the main constraints of the technology, which needs to be considered in the design of WSN. One of the techniques widely used to reduce the energy consumption is implemented by grouping a number of nodes into a cluster and restricting outside communications between clusters and base stations. Due to the short distances between sensor nodes within the cluster, their measurement results are expected to be associated with each other as well. However, this association may not be strict and it allows for variations and deviations due to a number of reasons, with the sensors low accuracy being probably the most contributing one. Figure 23.6 demonstrates the measurement results of humidity taken over a period of a few days in three locations under controlled environment: a homeroom, a dorm room, and a college laboratory. The measurements were taken with Telos ver.B sensor motes produced by Crossbow Inc., then they were communicated to, recorded, and processed at the PC base station. One can see a clear association between measurement results produced by sensors positioned in a close proximity to each other. This kind of association could be used for formulating a prediction model. However, Figure 23.6 demonstrates that the results are close but do not exactly repeat themselves. It means that the model incorporating a certain degree of uncertainty needs to be applied. Fuzzy granules seem to be perfect for formalizing this sort of knowledge.

23.7 Measurement Procedures and Uncertainty Evaluation: Fuzzy Approach Implementation

Once a measurement result is formalized as a fuzzy set, which in turn is considered as a stack of fuzzy intervals or α -cut sets, the method of the uncertainty propagation from input measured variables to the model output representing the uncertainty of indirect measurement results needs to be defined. The first method proposed in a few publications [17, 21] was based on a direct application of the Zadeh's

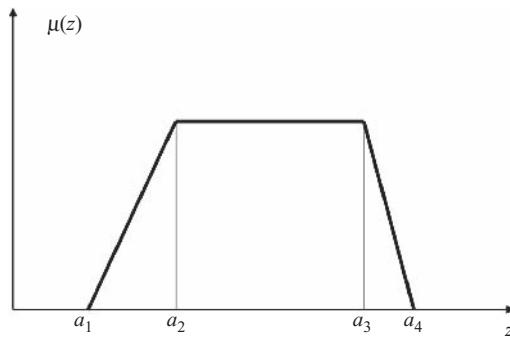


Figure 23.7 Fuzzy set trapezoidal membership function with four parameters

extension principle [28]. If X_1, X_2, \dots, X_n are the measurement results represented as fuzzy sets and $Y = f(X_1, X_2, \dots, X_n)$, then Y could be represented as a fuzzy set, whose membership function is determined according to the Zadeh's extension principle as

$$\begin{aligned}\mu_Y(y) &= \sup_{Y=f(X_1, X_2, \dots, X_n)(X_1, X_2, \dots, X_n) \in D} (\min(\mu_{X_1}(x_1), \dots, \mu_{X_n}(x_n))) \\ \text{or} \\ \mu_Y(y) &= 0 \quad \text{if} \quad f^{-1}(y) = 0.\end{aligned}\tag{6}$$

If the measured variables are not interactive, i.e., no assumptions have been made about the dependence or correlation between them or they can be assigned the values independently on each other, then the domain D becomes a Cartesian product:

$D = X_1 \times X_2 \times \dots \times X_n$ and the formula given above could be rewritten as

$$\begin{aligned}\mu_Y(y) &= \sup_{Y=f(X_1, X_2, \dots, X_n)} (\min(\mu_{X_1}(x_1), \dots, \mu_{X_n}(x_n))) \\ \text{or} \\ \mu_Y(y) &= 0 \quad \text{if} \quad f^{-1}(y) = 0.\end{aligned}\tag{7}$$

Zadeh's extension principle has been successfully applied as the foundation of fuzzy arithmetic as it provides the definition of all basic arithmetic operations, such as summation, subtraction, multiplication, and division.

In many practical cases of membership function particular shapes that could be parameterized, the operations above with the membership functions could be replaced with operations on their parameters. For example, in a case of trapezoidal membership functions (see Figure 23.7) defined with four parameters one can get just operations on those four parameters. This approach is well known [17, 21] and has been applied for processing measurement results and evaluation of their uncertainty.

As one can see from the formulas (6) and (7), the fuzzy arithmetic based on a direct application of the extension principle has the same feature as that of the interval arithmetic: the support of the sum of fuzzy intervals equals to the sum of supports. In a case of averaging the results of a series of measurements, the uncertainty of the average will be evaluated as the average of uncertainty characteristics of each result. This conclusion does not follow up the probabilistic model, where multiple measurements of the same measurand reduce the result's uncertainty. It may give a better expression of the Type B uncertainty but it will not be applicable for expressing the Type A uncertainty, which is supposed to decrease if a number of measurable variables and results received grows up. Type A usually represents a random component of measurement uncertainty, while Type B is better suited for presenting a systematic component. When random effects are actually present, the arithmetic fuzzy variables introduced above cannot take into account the probabilistic compensation that may affect the measurement process. Thus the fuzzy variables and the extension principle are a very effective tool in expressing and processing

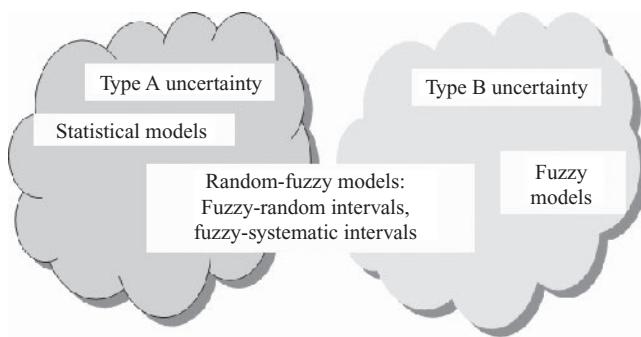


Figure 23.8 Application of different granulation models in measurement

uncertainty in measurement in all cases when the prevailing source of uncertainty could be attributed to systematic reasons. When the random sources are also present, the formulas and procedures given above do not fit the practical situation of the measurement process.

To deal with this situation two main approaches were proposed. One [24] is replacing a simple Zadeh's extension principle with operations on t-norms and another one [16, 29] applies the random-fuzzy variables. Both approaches lie within the framework of the Zadeh's definition of granulation and granular computing.

Urbanski and Wasowski [24] introduce two classes of fuzzy intervals:

1. fuzzy random intervals (FIR) for describing and processing the Type A measurement uncertainty and
2. fuzzy systematic intervals (FIS) for describing and processing the Type B measurement uncertainty.

The elements of FIS are the fuzzy sets S such that $S(x) = 1$ for all elements of its support. The elements of FIR are the fuzzy sets R such that $R(x) = 1$ for only one x . It is possible to prove that the operation of averaging of fuzzy intervals is converging to the crisp set, which ultimately means reducing and eliminating the uncertainty (see [30, 31] for mathematical details). A duality of FIRs and FISs allows describing and proper processing both Type A and Type B measurement uncertainty (see [24] for a numerical example). Figure 23.8 illustrates the classification and the place of different models in measurement procedures and uncertainty types.

We will follow [29] in our description of the approach based on random-fuzzy variables (RFV). The membership function of an RFV, defined on the reference set X , as well as a simple fuzzy set can be

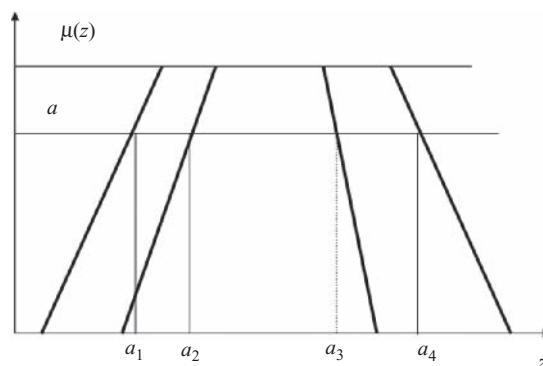


Figure 23.9 Random-fuzzy variable example

defined in terms of α -cuts, but an α -cut will now be represented as a quadruple set of four numbers instead of two only:

$A_\alpha = [a_1^\alpha, a_2^\alpha, a_3^\alpha, a_4^\alpha]$, where $a_1^\alpha \leq a_2^\alpha \leq a_3^\alpha \leq a_4^\alpha$. It can be easily recognized that all these α -cuts still represent nested focal elements of a body of evidence. According to the meaning assigned to an α -cut, interval $[a_1^\alpha, a_4^\alpha]$ represents a confidence interval with the level of $1 - \alpha$ confidence. As shown in Figure 23.9, within this interval three subintervals can be recognized, which differ from each other by the way how the possible values are distributed over the intervals themselves.

As far as intervals $[a_1^\alpha, a_2^\alpha]$ and $[a_3^\alpha, a_4^\alpha]$ are concerned, the possible values a , $a_1^\alpha \leq a \leq a_2^\alpha$ and a , $a_3^\alpha \leq a \leq a_4^\alpha$ are supposed to be randomly distributed according to a normal probability density function. In the case when $a_1^\alpha = a_2^\alpha$ and $a_3^\alpha = a_4^\alpha$, the random effects are removed and the RFV becomes a simple fuzzy variable, while if $a_2^\alpha = a_3^\alpha$, only random effects are represented in the RFV. Ferrero and Salicone [29] provide further and more detail description of the procedures, which generalize an application of a fuzzy approach to cover both Type A and Type B measurement uncertainties. An experimental example of how this method can be applied in a practical case can be found in [32].

23.8 Conclusion

Measurement is the most fundamental scientific process of gathering information from the objects or systems and their environments. The measurement result needs to be quantified. However, from the conceptual point of view the modern measurement science considers measurement results as estimates of the parameters of certain models, which had to be formulated before the measurement process is initiated. Due to imperfection of the models as well as of the measuring instruments, any measurement result is associated with some uncertainty and no measurement procedure is capable of producing the absolutely accurate results.

Being repeated, the measurement procedure delivers some distribution of the results, which is described as a granule incorporating some degree of uncertainty. This uncertainty could be formalized by different models, with probabilistic and interval analysis being the most widely applied in measurement. Procedures based on the probabilistic models have been standardized. They are presented in this chapter and the application examples are provided.

According to the international and national standards, the measurement uncertainty could be classified into two big groups: Type A and Type B. While probability-based models could well fit to the processing of Type A uncertainty, there exist obvious problems with their applications to Type B uncertainty. Over the last years other approaches based on fuzzy and interval analysis models have been proposed. Within a fuzzy approach, a few methods of processing uncertainty are considered, including an application of Zadeh's extension principle and random-fuzzy variables. Designed to become more general than a probabilistic approach, fuzzy-based methods allow avoiding pitfalls of the former. They are capable of the proper treatment of both Type A and Type B measurement uncertainty.

Overall, measurement science and processing of uncertainty in measurement employ the procedures, which could be described as granular computing. Depending on the formalization model chosen, probabilistic, interval, or fuzzy approaches should be applied.

References

- [1] P.H. Sydenham, N.H. Hancock, and R. Thorn. *Introduction to Measurement Science and Engineering*. Wiley, Chichester/New York, 1989.
- [2] L. Finkelstein. Theory and philosophy of measurement. In: P.H. Sydenham (ed.), *Handbook of Measurement Science*, Vol. 1. Wiley, Chichester/New York, 1982, pp. 1–30.
- [3] Wikipedia. Measurement uncertainty. http://en.wikipedia.org/wiki/Measurement_uncertainty, accessed October 1, 2006.
- [4] M. Kendall and A. Stuart. *The Advanced Theory of Statistics*. Griffin, London, 1977.
- [5] R. Moore. *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1966.
- [6] L. Zadeh. Graduation and granulation are keys to computation with information described in natural language. In: *2006 IEEE International Conference on Granular Computing*, Atlanta, USA, May 10–12, 2006, p. 30.

- [7] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer Norwell, MA, 2002.
- [8] A. Bargiela and W. Pedrycz. Granular mapping. *IEEE Trans. Syst. Man Cybern.* 35(2) (2005) 292–297.
- [9] A. Gacek and W. Pedrycz. A granular description of ECG signals. *IEEE Trans. Biomed. Eng.* 53(10) (2006) 1972–1982.
- [10] W. Pedrycz and K.-C. Kwak. Linguistic models as a framework of user-centric system modelling. *IEEE Trans. Syst. Man Cybern.* 36(4) (2006) 727–745.
- [11] R.W. Potter. *The Art of Measurement*. Prentice Hall, Englewood Cliffs, NJ, 2000.
- [12] *Guide to the Expression of Uncertainty in Measurement*. International Organization for Standardization, Geneva, 1995.
- [13] L. Zadeh. Toward a generalized theory of uncertainty (GTU)—an outline. In: *2005 IEEE International Conference on Granular Computing*, Vol. 1, Beijing, China, July 25–27, 2005, p. 16.
- [14] S. Rabinovich. *Measurement Errors and Uncertainty: Theory and Practice*. Springer-Verlag, New York, 2000.
- [15] S.K. Pal and P. Mitra. Multispectral image segmentation using the rough-set-initialized EM algorithm. *IEEE Trans. Geosci. Remote Sens.* 40(11) (2002) 2495–2501.
- [16] A. Ferrero, R. Gamba, and S. Salicone. A method based on random-fuzzy variables for the on-line estimation of measurement uncertainty of DSP-based instruments. *IEEE Trans. Instrum. Meas.* 53(5) (2004) 1362–1369.
- [17] G. Mauris, L. Berrah, L. Foulloy, and A. Haurat. Fuzzy handling of measurement errors in instrumentation. *IEEE Trans. Instrum. Meas.* 49(1) (2000) 89–93.
- [18] G. Mauris and L. Foulloy. A fuzzy symbolic approach to formalize sensory measurements: An application to a comfort sensor. *IEEE Trans. Instrum. Meas.* 51(4) (2002) 712–715.
- [19] L. Reznik and K.P. Dabke. Measurement models: Application of intelligent methods. *Measurement* 35 (2004) 47–58.
- [20] L. Reznik and K.P. Dabke. Evaluation of uncertainty in measurement: A proposal for application of intelligent methods. In: H. Imai (ed.), *Measurement to Improve Quality of Life in the 21st Century*, IMEKO –XV World Congress, June 13–18, 1999, Osaka, Japan, vol. II, pp. 93–100.
- [21] L. Reznik and G.N. Solopchenko. Use of a priori information on functional relations between measured quantities for improving accuracy of measurement. *Measurement* 3(3) (1985) 98–106.
- [22] A.R. Varkony-Koczy, T.P. Dobrowiecki, and G. Peceli. Measurement uncertainty: a soft computing approach. In: *Proceedings of the 1997 IEEE International Conference on Intelligent Engineering Systems, INES '97*, Budapest, Hungary, September 15–17, 1997, pp. 485–490.
- [23] D. Dubois and H. Prade. Fuzzy sets, probability, and measurement. *Eur. J. Oper. Res.* 40 (1989) 135–154.
- [24] M. Urbanski and J. Wasowski. *Application of Fuzzy Arithmetic to Uncertainty Measurement Theory*, Academy of Sciences Series Conference Materials, Gliwice-Uston, May 7–9, 2001, pp. 39–50.
- [25] I. Ashokaraj, A. Tsourdos, P. Silson, B. White, and J. Economou. Feature based robot navigation: Using fuzzy logic and interval analysis. In: *2004 IEEE International Conference on Fuzzy Systems*, Budapest, Hungary, July 25–29, 2004 , Vol. 3, pp. 1461–1466.
- [26] A. Mahajan, K. Wang, and P.K. Ray. Multisensor integration and fusion model that uses a fuzzy inference system. *IEEE/ASME Trans. Mechatronics* 6(2) (2001) 188–196.
- [27] G. Mauris, V. Lasserre, and L. Foulloy. Fuzzy modeling of measurement data acquired from physical sensors. *IEEE Trans. Instrum. Meas.* 49(6) (2000) 1201–1205.
- [28] L. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Syst. Man Cybern.* 3 (1973) 28–44.
- [29] A. Ferrero and S. Salicone. The random-fuzzy variables: A new approach to the expression of uncertainty in measurement. *IEEE Trans. Instrum. Meas.* 53(5) (2004) 1370–1377.
- [30] D.H. Hong and R.I. Ro. The law of large numbers for fuzzy unbounded supports. *Fuzzy Sets Syst.* 116 (2000) 269–274.
- [31] M.K. Urbanski and J. Wasowski. Fuzzy approach to the theory of measurement inexactness. *Measurement* 34 (2003) 67–74.
- [32] A. Ferrero and S. Salicone. An innovative approach to the determination of uncertainty in measurements based on fuzzy variables. *IEEE Trans. Instrum. Meas.* 52(4) (2003) 1174–1181.

24

Fuzzy Rough Sets: From Theory into Practice

Chris Cornelis, Martine De Cock, and Anna Maria Radzikowska

24.1 Introduction

Fuzzy sets [1], as well as the slightly younger rough sets [2], have left an important mark on the way we represent and compute with imperfect information nowadays. Each of them has fostered a broad research community, and their impact has also been clearly felt at the application level. Although it was recognized early on that the associated theories are complementary rather than competitive, perceived similarities between both concepts and efforts to prove that one of them subsumes the other have somewhat stalled progress toward shaping a hybrid theory that combines their mutual strengths.

Still, seminal research on fuzzy rough set theory flourished during the 1990s and early 2000s (e.g., [3–16]), and recently, cross-disciplinary research has also profited from the popularization and widespread adoption of two important computing paradigms: granular computing, with its focus on clustering information entities into granules in terms of similarity, indistinguishability, and so on, has helped the theoretical underpinnings of the hybrid theory to come of age, while soft computing – a collection of techniques that are tolerant of typical characteristics of imperfect data and knowledge and hence adhere closer to the human mind than conventional hard computing techniques – has stressed the role of fuzzy sets and rough sets as partners, rather than adversaries, within a panoply of practical applications.

Within the hybrid theory, Pawlak’s well-known framework for the construction of lower and upper approximations of a concept C given incomplete information (a subset A of a given universe X , containing examples of C), and an equivalence relation R in X that models ‘indiscernibility’ or ‘indistinguishability,’ has been extended in two ways:

1. The set A may be generalized to a fuzzy set in X , allowing that objects can belong to a concept (i.e., meet its characteristics) to varying degrees.
2. Rather than modeling elements’ indistinguishability, we may assess their similarity (objects are similar to a certain degree), represented by a fuzzy relation R . As a result, objects are categorized into classes, or granules, with ‘soft’ boundaries based on their similarity to one another.

In this chapter, we consider the general problem of defining lower and upper approximations of a fuzzy set A by means of a fuzzy relation R . A key ingredient to our exposition will be the fact that elements of X can belong, to varying degrees, to several ‘soft granules’ simultaneously. Not only does

this property lie right at the heart of fuzzy set theory, a similar phenomenon can already be observed in crisp, or traditional, rough set theory as soon as the assumption that R is an equivalence relation (and hence induces a partition of X) is abandoned. Within fuzzy rough set theory, the impact of this property – which plays a crucial role toward defining the approximations – is felt still more strongly, since even fuzzy \mathcal{T} -equivalence relations, the natural candidates for generalizing equivalence relations, are subject to it.

This chapter is structured as follows. In Section 24.2, we first recall the necessary background on rough sets and fuzzy sets. Section 24.3 reviews various proposals for the definition of a fuzzy rough set and examines their respective properties. Furthermore, Section 24.4 reveals that the various alternative definitions are not just of theoretical interest but become useful in a topical application, such as query refinement for searching on the World Wide Web (WWW), especially in the presence of ambiguous query terms.

24.2 Preliminaries

24.2.1 Rough Sets

Rough set analysis makes statements about the membership of some element y of X to the concept of which A is a set of examples, based on the indistinguishability between y and the elements of A . Usually, indistinguishability is described by means of an equivalence relation R on X ; for example, if the elements of X are represented by a set of attributes, two elements of X are indistinguishable if they have the same value for all attributes. In this case, (X, R) is called a standard, or Pawlak, approximation space. More generally, it is possible to replace R by any binary relation in X , not necessarily an equivalence relation; we then call (X, R) a generalized approximation space. In particular, the case of a reflexive R and a tolerance, i.e., reflexive and symmetric, relation R have received ample attention in the literature.

In all cases, A is approximated in two ways, resulting in the lower and upper approximation of the concept. In the next paragraphs, we will review the definitions of these approximations.

For completeness we mention that a second stream concerning rough sets in the literature was initiated by Iwinski [17], who did not use an equivalence relation or tolerance relation as an initial building block to define the rough set concept. Although this formulation provides an elegant mathematical model, the absence of the equivalence relation makes this model hard to interpret. We therefore do not deal with it in this chapter; a more detailed comparison of the different views on rough set theory can be found in, e.g., [18]. For a recent series of survey papers on rough sets, we refer to [19–21].

24.2.1.1 Rough Sets in Pawlak Approximation Spaces

In a Pawlak approximation space (X, R) , an element y of X belongs to the lower approximation $R \downarrow A$ of A if the equivalence class to which y belongs is included in A . On the other hand, y belongs to the upper approximation $R \uparrow A$ of A if its equivalence class has a non-empty intersection with A . Formally, the sets $R \downarrow A$ and $R \uparrow A$ are defined by, for y in X ,

$$y \in R \downarrow A \quad \text{iff} \quad [y]_R \subseteq A, \quad (1)$$

$$y \in R \uparrow A \quad \text{iff} \quad [y]_R \cap A \neq \emptyset. \quad (2)$$

In other words,

$$y \in R \downarrow A \quad \text{iff} \quad \forall x \in X(x, y) \in R \Rightarrow x \in A, \quad (3)$$

$$y \in R \uparrow A \quad \text{iff} \quad \exists x \in X(x, y) \in R \wedge x \in A. \quad (4)$$

The underlying meaning is that $R \downarrow A$ is the set of elements *necessarily* satisfying the concept (strong membership), while $R \uparrow A$ is the set of elements *possibly* belonging to the concept (weak membership).

Table 24.1 Properties of lower and upper approximation in a Pawlak approximation space (X, R) ^a

-
1. $R \uparrow A = co(R \downarrow (coA))$
 $R \downarrow A = co(R \uparrow (coA))$
 2. $R \downarrow A \subseteq A \subseteq R \uparrow A$
 3. $A \subseteq B \Rightarrow (R \downarrow A \subseteq R \downarrow B \text{ and } R \uparrow A \subseteq R \uparrow B)$
 4. $R \downarrow(A \cap B) = R \downarrow A \cap R \downarrow B$
 $R \uparrow(A \cap B) \subseteq R \uparrow A \cap R \uparrow B$
 5. $R \downarrow(A \cup B) \supseteq R \downarrow A \cup R \downarrow B$
 $R \uparrow(A \cup B) = R \uparrow A \cup R \uparrow B$
 6. $R \downarrow(R \downarrow A) = R \downarrow A$
 $R \uparrow(R \uparrow A) = R \uparrow A$
-

^a A and B are subsets of X , and co denotes set-theoretic complement.

Some basic and easily verified properties of lower and upper approximation are summarized in Table 24.1. From (2), it holds that $R \downarrow A \subseteq R \uparrow A$. If y belongs to the boundary region $R \uparrow A \setminus R \downarrow A$, then there is some doubt, because in this case y is at the same time indistinguishable from at least one element of A and at least one element of X that is not in A . Following [13], we call (A_1, A_2) a rough set (in (X, R)) as soon as there is a set A in X , such that $R \downarrow A = A_1$ and $R \uparrow A = A_2$.

24.2.1.2 Rough Sets in Generalized Approximation Spaces

In this section, we assume that R is a tolerance relation¹ in X . In this case, the role of equivalence classes in Pawlak approximation spaces (cf. formulas (1) and (2)) can be subsumed by the more general concept of R -forests; recall that, for y in X , the R -foreset Ry is defined by

$$Ry = \{x \mid x \in X \text{ and } (x, y) \in R\}. \quad (5)$$

It is well known that for an equivalence relation R , R induces a partition of X , so if we consider two equivalence classes then they either coincide or are disjoint. It is therefore not possible for y to belong to two different equivalence classes at the same time. If R is a non-equivalence relation in X , however, then it is quite normal that different foresets may partially overlap.

By the definition used so far, y belongs to the lower approximation of A if Ry is included in A . In view of the discussion above, however, it makes sense to consider also other R -forests that contain y and to assess their inclusion into A , as well for the lower approximation, and their overlap with A for the upper approximation. This idea, explored among others by [25–30], results in the following (inexhaustive) list of candidate definitions for the lower and the upper approximation of A :

1. y belongs to the lower approximation of A iff
 - (a) all R -forests containing y are included in A ,
 - (b) at least one R -foreset containing y is included in A ,
 - (c) Ry is included in A .
2. y belongs to the upper approximation of A iff
 - (a) all R -forests containing y have a non-empty intersection with A ,
 - (b) at least one R -foreset containing y has a non-empty intersection with A ,
 - (c) Ry has a non-empty intersection with A .

¹ For an approach where the indiscernibility relation is replaced by a dominance relation, we refer to [22]. The so-called dominance rough-set-based approximations of fuzzy sets (see also [23, 24]) do not rely on fuzzy connectives and are therefore different from the approach dealt with in the current chapter.

Paraphrasing these expressions, we obtain the following definitions:

1. The tight, loose, and (usual) lower approximation of A are defined as
 - (a) $y \in R_{\downarrow\downarrow} A$ iff $\forall z \in X, y \in Rz \Rightarrow Rz \subseteq A$,
 - (b) $y \in R_{\uparrow\downarrow} A$ iff $\exists z \in X, y \in Rz \wedge Rz \subseteq A$,
 - (c) $y \in R_{\downarrow} A$ iff $Ry \subseteq A$,
for all y in X .
2. The tight, loose, and (usual) upper approximation of A are defined as
 - (a) $y \in R_{\downarrow\uparrow} A$ iff $\forall z \in X, y \in Rz \Rightarrow Rz \cap A \neq \emptyset$,
 - (b) $y \in R_{\uparrow\uparrow} A$ iff $\exists z \in X, y \in Rz \wedge Rz \cap A \neq \emptyset$,
 - (c) $y \in R_{\uparrow} A$ iff $Ry \cap A \neq \emptyset$,

for all y in X .

Note 1. The terminology ‘tight’ refers to the fact that we take all R -foresets classes into account, giving rise to a strict or tight requirement. For the ‘loose’ approximations, we only look at ‘the best one,’ which is clearly a more flexible demand. For an equivalence relation R , all of the above definitions coincide, but in general they can be different as the following example shows.

Example 2. Consider $X = \{x_1, x_2, x_3, x_4\}$, $A = \{x_1, x_3\}$, and the relation R in X defined by

R	x_1	x_2	x_3	x_4
x_1	1	1	1	0
x_2	1	1	0	1
x_3	1	0	1	0
x_4	0	1	0	1

Then

$$\begin{aligned} R_{\downarrow} A &= \{x_3\}, & R_{\uparrow} A &= \{x_1, x_2, x_3\}, \\ R_{\uparrow\downarrow} A &= \{x_1, x_3\}, & R_{\uparrow\uparrow} A &= X, \\ R_{\downarrow\downarrow} A &= \emptyset, & R_{\downarrow\uparrow} A &= \{x_1, x_3\}. \end{aligned}$$

In general, the symmetry of R allows to verify the following relationships between the approximations:

$$R_{\downarrow\downarrow} A = R_{\downarrow}(R_{\downarrow} A), \quad (6)$$

$$R_{\uparrow\downarrow} A = R_{\uparrow}(R_{\downarrow} A), \quad (7)$$

$$R_{\downarrow\uparrow} A = R_{\downarrow}(R_{\uparrow} A), \quad (8)$$

$$R_{\uparrow\uparrow} A = R_{\uparrow}(R_{\uparrow} A). \quad (9)$$

Table 24.2 lists the properties of the different approximations. Interesting observations to make from this table include:

1. By (1) there are three pairs of dual approximation operators w.r.t. complementation.
2. Property 2 shows the relationship between the approximations in terms of inclusion and how A itself fits into this picture. Note how these relationships nicely justify the terminology.
3. Loose lower, resp. tight upper, approximation satisfies only a weak interaction property w.r.t. set intersection, resp. union (Properties 4 and 5).
4. By Property 6 of Table 24.1, when R is an equivalence relation, lower and upper approximation are idempotent. This means that in Pawlak approximation spaces, maximal reduction and expansion are achieved within one approximation step. The same holds true for loose lower and tight upper approximation in a symmetric approximation space, but not for the other operators; for these, a gradual reduction/expansion process is obtained by successively taking approximations.

Table 24.2 Properties of lower and upper approximation in a symmetric approximation space (X, R)

1. $R \uparrow A = co(R \downarrow (coA))$	4. $R \downarrow(A \cap B) = R \downarrow A \cap R \downarrow B$
$R \downarrow A = co(R \uparrow (coA))$	$R \uparrow(A \cap B) \subseteq R \uparrow A \cap R \uparrow B$
$R \downarrow \uparrow A = co(R \downarrow \uparrow (coA))$	$R \downarrow \uparrow(A \cap B) \subseteq R \downarrow \uparrow A \cap R \downarrow \uparrow B$
$R \uparrow \downarrow A = co(R \uparrow \downarrow (coA))$	$R \uparrow \downarrow(A \cap B) \subseteq R \uparrow \downarrow A \cap R \uparrow \downarrow B$
$R \uparrow \uparrow A = co(R \downarrow \downarrow (coA))$	$R \uparrow \uparrow(A \cap B) \subseteq R \uparrow \uparrow A \cap R \uparrow \uparrow B$
$R \downarrow \downarrow A = co(R \uparrow \uparrow (coA))$	$R \downarrow \downarrow(A \cap B) = R \downarrow \downarrow A \cap R \downarrow \downarrow B$
2. $R \downarrow \downarrow A \subseteq R \downarrow A \subseteq R \uparrow \downarrow A \subseteq A$ $A \subseteq R \downarrow \uparrow A \subseteq R \uparrow A \subseteq R \uparrow \uparrow A$	5. $R \downarrow(A \cup B) \supseteq R \downarrow A \cup R \downarrow B$ $R \uparrow(A \cup B) = R \uparrow A \cup R \uparrow B$ $R \downarrow \uparrow(A \cup B) \supseteq R \downarrow \uparrow A \cup R \downarrow \uparrow B$ $R \uparrow \downarrow(A \cup B) \supseteq R \uparrow \downarrow A \cup R \uparrow \downarrow B$ $R \uparrow \uparrow(A \cup B) = R \uparrow \uparrow A \cup R \uparrow \uparrow B$ $R \downarrow \downarrow(A \cup B) \supseteq R \downarrow \downarrow A \cup R \downarrow \downarrow B$
3. $A \subseteq B \Rightarrow \begin{cases} R \downarrow A \subseteq R \downarrow B \\ R \uparrow A \subseteq R \uparrow B \\ R \downarrow \uparrow A \subseteq R \downarrow \uparrow B \\ R \uparrow \downarrow A \subseteq R \uparrow \downarrow B \\ R \uparrow \uparrow A \subseteq R \uparrow \uparrow B \\ R \downarrow \downarrow A \subseteq R \downarrow \downarrow B \end{cases}$	6. $R \downarrow \uparrow(R \downarrow \uparrow A) = R \downarrow \uparrow A$ $R \uparrow \downarrow(R \uparrow \downarrow A) = R \uparrow \downarrow A$

24.2.2 Fuzzy Sets

In the context of fuzzy rough set theory, A is a fuzzy set in X , i.e., an $X \rightarrow [0, 1]$ mapping, while R is a fuzzy relation in X , i.e., a fuzzy set in $X \times X$. Recall that for all y in X , the R -foreset of y is the fuzzy set Ry defined by

$$Ry(x) = R(x, y) \quad (10)$$

for all x in X . The fuzzy logical counterparts of the connectives in (3) and (4) play an important role in the generalization of lower and upper approximations; we therefore recall some important definitions.

First, a negator \mathcal{N} is a decreasing $[0, 1] \rightarrow [0, 1]$ mapping satisfying $\mathcal{N}(0) = 1$ and $\mathcal{N}(1) = 0$. \mathcal{N} is called involutive if $\mathcal{N}(\mathcal{N}(x)) = x$ for all x in $[0, 1]$. The standard negator \mathcal{N}_s is defined by $\mathcal{N}_s(x) = 1 - x$. A negator \mathcal{N} induces a corresponding fuzzy set complement $co_{\mathcal{N}}$: for any fuzzy set A in X and every element x in X ,

$$co_{\mathcal{N}}(A) = \mathcal{N}(A(x)). \quad (11)$$

A triangular norm (t -norm for short) \mathcal{T} is any increasing, commutative, and associative $[0, 1]^2 \rightarrow [0, 1]$ mapping satisfying $\mathcal{T}(1, x) = x$, for all x in $[0, 1]$. Analogously, a triangular conorm (t -conorm for short) \mathcal{S} is any increasing, commutative, and associative $[0, 1]^2 \rightarrow [0, 1]$ mapping satisfying $\mathcal{S}(0, x) = x$, for all x in $[0, 1]$. Table 24.3 mentions some important t -norms and t -conorms. The \mathcal{T} -intersection and \mathcal{S} -union of fuzzy sets A and B in X are defined by

$$(A \cap_{\mathcal{T}} B)(x) = \mathcal{T}(A(x), B(x)), \quad (12)$$

$$(A \cup_{\mathcal{S}} B)(x) = \mathcal{S}(A(x), B(x)), \quad (13)$$

for all x in X . Throughout this chapter, $A \cap_{\mathcal{T}_M} B$ and $A \cup_{\mathcal{S}_M} B$ are abbreviated to $A \cap B$ and $A \cup B$ and called standard complement and union, respectively.

Finally, an impicator is any $[0, 1]^2 \rightarrow [0, 1]$ -mapping \mathcal{I} satisfying $\mathcal{I}(0, 0) = 1$, $\mathcal{I}(1, x) = x$, for all x in $[0, 1]$. Moreover we require \mathcal{I} to be decreasing in its first and increasing in its second component. If

Table 24.3 Well-known t -norms and t -conorms; x and y in $[0, 1]$

t -norm	t -conorm
$T_M(x, y) = \min(x, y)$	$S_M(x, y) = \max(x, y)$
$T_P(x, y) = xy$	$S_P(x, y) = x + y - xy$
$T_W(x, y) = \max(x + y - 1, 0)$	$S_W(x, y) = \min(x + y, 1)$

\mathcal{T} is a t -norm, the mapping $\mathcal{I}_{\mathcal{T}}$ defined by, for all x and y in $[0, 1]$,

$$\mathcal{I}_{\mathcal{T}}(x, y) = \sup\{\lambda | \lambda \in [0, 1] \text{ and } \mathcal{T}(x, \lambda) \leq y\} \quad (14)$$

is an impicator, usually called the residual impicator of \mathcal{T} . If \mathcal{T} is a t -conorm and \mathcal{N} is an involutive negator, then the mapping $\mathcal{I}_{\mathcal{T}, \mathcal{N}}$ defined by, for all x and y in $[0, 1]$,

$$\mathcal{I}_{\mathcal{T}, \mathcal{N}}(x, y) = \mathcal{N}(\mathcal{T}(x, \mathcal{N}(y))) \quad (15)$$

is an impicator, usually called the S -impicator induced by \mathcal{T} and \mathcal{N} . In Table 24.4, we mention some important S - and residual implicators; the S -implicators are induced by means of the standard negator \mathcal{N}_s .

In fuzzy rough set theory, we require a way to express that objects are similar to each other to some extent. In the context of this chapter, similarity is modeled by a fuzzy tolerance relation R ; that is,

$$\begin{aligned} R(x, x) &= 1 && \text{(reflexivity),} \\ R(x, y) &= R(y, x) && \text{(symmetry)} \end{aligned}$$

hold for all x and y in X . Additionally, \mathcal{T} -transitivity (for a particular t -norm \mathcal{T}) is sometimes imposed: for all x , y , and z in X ,

$$\mathcal{T}(R(x, y), R(y, z)) \leq R(x, z) \quad (\mathcal{T}\text{-transitivity}).$$

R is then called a fuzzy \mathcal{T} -equivalence relation; because equivalence relations are used to model equality, fuzzy \mathcal{T} -equivalence relations are commonly considered to represent approximate equality. In general, for a fuzzy tolerance relation R , we will call Ry the ‘fuzzy similarity class’ of y .

24.3 Fuzzy Rough Sets

24.3.1 Definitions

Research on fuzzifying lower and upper approximations in the spirit of Pawlak emerged in the late 1980s. Chronologically, the first proposals are due to Nakamura [11] and to Dubois and Prade [3], who drew inspiration from an earlier publication by Fariñas del Cerro and Prade [31].

Table 24.4 Well-known implicators; x and y in $[0, 1]$

S -implicator	Residual impicator
$\mathcal{I}_{S_M}(x, y) = \max(1 - x, y)$	$\mathcal{I}_{T_M}(x, y) = \begin{cases} 1, & \text{if } x \leq y \\ y, & \text{otherwise} \end{cases}$
$\mathcal{I}_{S_P}(x, y) = 1 - x + xy$	$\mathcal{I}_{T_P}(x, y) = \begin{cases} 1, & \text{if } x \leq y \\ \frac{y}{x}, & \text{otherwise} \end{cases}$
$\mathcal{I}_{S_W}(x, y) = \min(1 - x + y, 1)$	$\mathcal{I}_{S_W}(x, y) = \min(1 - x + y, 1)$

In developing the generalizations, the central focus moved from elements' indistinguishability (for instance, w.r.t. their attribute values in an information system) to their similarity: objects are categorized into classes with 'soft' boundaries based on their similarity to one another. A concrete advantage of such a scheme is that abrupt transitions between classes are replaced by gradual ones, allowing that an element can belong (to varying degrees) to more than one class. An example at hand is an attribute 'age' in an information table: in order to restrict the number of equivalence classes, classical rough set theory advises to discretize age values by a crisp partition of the universe, e.g., using intervals $[0, 10]$, $[10, 20]$, ... This does not always reflect our intuition, however: by imposing such harsh boundaries, a person who has just turned 11 will not be taken into account in the $[0, 10]$ class, even when he/she is only at a minimal remove from full membership in that class.

Guided by that observation, many people have suggested alternatives for defining generalized approximation operators, e.g., using axiomatic approaches [10], based on Iwinski-type rough sets [12], in terms of α -cuts [16], level fuzzy sets [32], or fuzzy inclusion measures [8], etc. Some authors (e.g., [15, 16]) explicitly distinguish between rough fuzzy sets (approximations of a fuzzy set in a crisp approximation space) and fuzzy rough sets (approximations of a crisp set in a fuzzy approximation space, i.e., defined by a fuzzy relation R).

A fairly general definition of a fuzzy rough set, absorbing earlier suggestions in the same direction, was given by Radzikowska and Kerre [13]. They paraphrased formulas (3) and (4), which hold in the crisp case, to define the lower and upper approximation of a fuzzy set A in X as the fuzzy sets $R \downarrow A$ and $R \uparrow A$ in X , constructed by means of an impicator \mathcal{I} , a t -norm \mathcal{T} and a fuzzy \mathcal{T} -equivalence relation R in X ,

$$R \downarrow A(y) = \inf_{x \in X} \mathcal{I}(R(x, y), A(x)), \quad (16)$$

$$R \uparrow A(y) = \sup_{x \in X} \mathcal{T}(R(x, y), A(x)), \quad (17)$$

for all y in X . (A_1, A_2) is called a fuzzy rough set (in (X, R)) as soon as there is a fuzzy set A in X such that $R \downarrow A = A_1$ and $R \uparrow A = A_2$. Formulas (16) and (17) for $R \downarrow A$ and $R \uparrow A$ can also be interpreted as the degree of inclusion of Ry in A and the degree of overlap of Ry and A , respectively, which indicates the semantical link with (1) and (2).

What this definition does not take into account, however, is the fact that if R is a fuzzy \mathcal{T} -equivalence relation then it is quite normal that, because of the intermediate degrees of membership, different foresets are not necessarily disjoint. The following example, taken from [33], illustrates this.

Example 3. In applications, T_W is often used as a t -norm because the notion of fuzzy T_W -equivalence relation is dual to that of a pseudometric [34]. Let the fuzzy T_W -equivalence relation R in \mathbb{R} be defined by

$$R(x, y) = \max(1 - |x - y|, 0)$$

for all x and y in \mathbb{R} . Figure 24.1 depicts the R -foresets of 1.3, 2.2, 3.1, and 4.0.

The R -foresets of 3.1 and 4.0 are clearly different. Still one can easily see that

$$R(3.1, 3.5) = 0.6,$$

$$R(4.0, 3.5) = 0.5.$$

Since $T_W(0.6, 0.5) = 0.1$, 3.5 belongs to degree 0.1 to the T_W -intersection of the R -foresets of 3.1 and 4.0; i.e., these R -foresets are not disjoint.

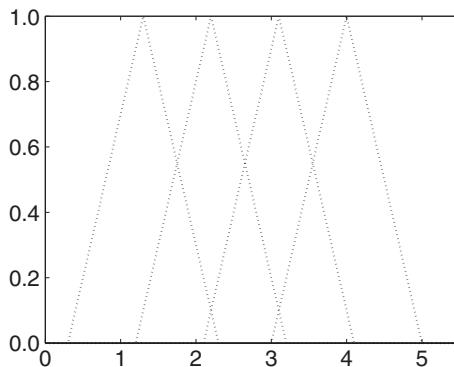


Figure 24.1 Fuzzy similarity classes

In other words, the traditional distinction between equivalence and non-equivalence relations is lost when moving on to a fuzzy \mathcal{T} -equivalence relation, so it makes sense to exploit the fact that an element can belong to some degree to several R -forests of *any* fuzzy relation R at the same time. Natural generalizations to the definitions from Section 24.2.1.2 were therefore proposed in [35, 33].

Definition 4. Let R be a fuzzy relation in X and A a fuzzy set in X .

1. The tight, loose, and (usual) lower approximation of A are defined as

- (a) $R\downarrow\downarrow A(y) = \inf_{z \in X} \mathcal{I}(Rz(y), \inf_{x \in X} \mathcal{I}(Rz(x), A(x)))$,
- (b) $R\uparrow\downarrow A(y) = \sup_{z \in X} \mathcal{T}(Rz(y), \inf_{x \in X} \mathcal{I}(Rz(x), A(x)))$,
- (c) $R\downarrow A(y) = \inf_{x \in X} \mathcal{I}(Ry(x), A(x)),$

for all y in X .

2. The tight, loose, and (usual) upper approximation of A are defined as

- (a) $R\downarrow\uparrow A(y) = \inf_{z \in X} \mathcal{I}(Rz(y), \sup_{x \in X} \mathcal{T}(Rz(x), A(x)))$,
- (b) $R\uparrow\uparrow A(y) = \sup_{z \in X} \mathcal{T}(Rz(y), \sup_{x \in X} \mathcal{T}(Rz(x), A(x)))$,
- (c) $R\uparrow A(y) = \sup_{x \in X} \mathcal{T}(Ry(x), A(x)),$

for all y in X .

In the next section, we investigate the main properties of these alternative approximation operators.

24.3.2 Properties of Fuzzy Rough Sets

In this section, we will assume that R is a fuzzy tolerance relation in X . Some properties require additional \mathcal{T} -transitivity of R ; whenever this is the case we mention it explicitly. An overview of the properties discussed in this section is given in Table 24.5.

24.3.2.1 Links between the Approximations

Just like in the crisp case, tight and loose approximation operators can be expressed in terms of the usual ones, due to the symmetry of R .

Table 24.5 Properties of lower and upper approximation in a fuzzy approximation space $(X, R)^a$

Property	Conditions
1. $R \uparrow A = co_{\mathcal{N}}(R \downarrow (co_{\mathcal{N}}A))$ $R \downarrow A = co_{\mathcal{N}}(R \uparrow (co_{\mathcal{N}}A))$ $R \downarrow \uparrow A = co_{\mathcal{N}}(R \uparrow \downarrow (co_{\mathcal{N}}A))$ $R \uparrow \downarrow A = co_{\mathcal{N}}(R \downarrow \uparrow (co_{\mathcal{N}}A))$ $R \uparrow \uparrow A = co_{\mathcal{N}}(R \downarrow \downarrow (co_{\mathcal{N}}A))$ $R \downarrow \downarrow A = co_{\mathcal{N}}(R \uparrow \uparrow (co_{\mathcal{N}}A))$	\mathcal{N} involutive, $\mathcal{I} = \mathcal{I}_{\mathcal{T}, \mathcal{N}}$; or, \mathcal{T} left continuous, $\mathcal{I} = \mathcal{I}_{\mathcal{T}}$ and $\mathcal{N}(x) = \mathcal{I}(x, 0)$, \mathcal{N} involutive (Proposition 11)
2. $R \downarrow \downarrow A \subseteq R \downarrow A \subseteq R \uparrow \downarrow A \subseteq A$ $A \subseteq R \downarrow \uparrow A \subseteq R \uparrow A \subseteq R \uparrow \uparrow A$	$\mathcal{T}(x, \mathcal{I}(x, y)) \leq y$ and $y \leq \mathcal{I}(x, \mathcal{T}(x, y))$ (Propositions 8 and 9)
3. $A \subseteq B \Rightarrow \begin{cases} R \downarrow A \subseteq R \downarrow B \\ R \uparrow A \subseteq R \uparrow B \\ R \uparrow \downarrow A \subseteq R \downarrow \uparrow B \\ R \uparrow \downarrow A \subseteq R \uparrow \downarrow B \\ R \uparrow \uparrow A \subseteq R \uparrow \uparrow B \\ R \downarrow \downarrow A \subseteq R \downarrow \downarrow B \end{cases}$	Always (Proposition 6)
4. $R \downarrow(A \cap B) = R \downarrow A \cap R \downarrow B$ $R \uparrow(A \cap B) \subseteq R \uparrow A \cap R \uparrow B$ $R \downarrow \uparrow(A \cap B) \subseteq R \downarrow \uparrow A \cap R \downarrow \uparrow B$ $R \uparrow \downarrow(A \cap B) \subseteq R \uparrow \downarrow A \cap R \uparrow \downarrow B$ $R \uparrow \uparrow(A \cap B) \subseteq R \uparrow \uparrow A \cap R \uparrow \uparrow B$ $R \downarrow \downarrow(A \cap B) = R \downarrow \downarrow A \cap R \downarrow \downarrow B$	Always (Proposition 12)
5. $R \downarrow(A \cup B) \supseteq R \downarrow A \cup R \downarrow B$ $R \uparrow(A \cup B) = R \uparrow A \cup R \uparrow B$ $R \downarrow \uparrow(A \cup B) \supseteq R \downarrow \uparrow A \cup R \downarrow \uparrow B$ $R \uparrow \downarrow(A \cup B) \supseteq R \uparrow \downarrow A \cup R \uparrow \downarrow B$ $R \uparrow \uparrow(A \cup B) = R \uparrow \uparrow A \cup R \uparrow \uparrow B$ $R \downarrow \downarrow(A \cup B) \supseteq R \downarrow \downarrow A \cup R \downarrow \downarrow B$	Always (Proposition 12)
6. $R \downarrow \uparrow(R \downarrow \uparrow A) = R \downarrow \uparrow A$ $R \uparrow \downarrow(R \uparrow \downarrow A) = R \uparrow \downarrow A$ $R \uparrow \downarrow A = R \downarrow \downarrow A = R \downarrow A$ $R \downarrow \uparrow A = R \uparrow \uparrow A = R \uparrow A$	\mathcal{T} left continuous, $\mathcal{I} = \mathcal{I}_{\mathcal{T}}$ (Proposition 13) R a fuzzy \mathcal{T} -equivalence relation in X , \mathcal{T} left continuous, $\mathcal{I} = \mathcal{I}_{\mathcal{T}}$ (Propositions 17 and 18)

^a R is a fuzzy tolerance relation.

Proposition 5. For every fuzzy set A in X

$$R \downarrow \downarrow A = R \downarrow(R \downarrow A), \quad (18)$$

$$R \uparrow \downarrow A = R \uparrow(R \downarrow A), \quad (19)$$

$$R \downarrow \uparrow A = R \downarrow(R \uparrow A), \quad (20)$$

$$R \uparrow \uparrow A = R \uparrow(R \uparrow A). \quad (21)$$

The monotonicity of the approximations follows easily due to the monotonicity of the fuzzy logical operators involved. This is reflected in the next proposition.

Proposition 6. For every fuzzy set A and B in X

$$A \subseteq B \Rightarrow \begin{cases} R\downarrow A \subseteq R\downarrow B \\ R\uparrow A \subseteq R\uparrow B \\ R\uparrow\uparrow A \subseteq R\uparrow\uparrow B \\ R\downarrow\uparrow A \subseteq R\downarrow\uparrow B \\ R\uparrow\downarrow A \subseteq R\uparrow\downarrow B \\ R\downarrow\downarrow A \subseteq R\downarrow\downarrow B. \end{cases} \quad (22)$$

The following proposition supports the idea of approximating a concept from the lower and the upper side.

Proposition 7 [13]. For every fuzzy set A in X

$$R\downarrow A \subseteq A \subseteq R\uparrow A. \quad (23)$$

For the tight and loose approximations, due to Propositions 5, 6, and 7, we can make the following general observations.

Proposition 8. For every fuzzy set A in X

$$R\downarrow\downarrow A \subseteq R\downarrow A \subseteq A \subseteq R\uparrow A \subseteq R\uparrow\uparrow A, \quad (24)$$

$$R\downarrow A \subseteq R\uparrow\downarrow A \subseteq R\uparrow A, \quad (25)$$

$$R\downarrow A \subseteq R\downarrow\uparrow A \subseteq R\uparrow A. \quad (26)$$

However, the proposition does not give any immediate information about a direct relationship between the loose lower and the tight upper approximation in terms of inclusion and about how A itself fits in this picture. The following proposition sheds some light on this matter.

Proposition 9. If \mathcal{T} and \mathcal{I} satisfy $\mathcal{T}(x, \mathcal{I}(x, y)) \leq y$ and $y \leq \mathcal{I}(x, \mathcal{T}(x, y))$ for all x and y in $[0, 1]$, then for every fuzzy set A in X ,

$$R\uparrow\downarrow A \subseteq A \subseteq R\downarrow\uparrow A. \quad (27)$$

In particular, if \mathcal{T} is a left-continuous t -norm and \mathcal{I} is its residual impicator, the property holds [36]. Proposition 9 does not hold in general for other choices of t -norms and implicants as the next example illustrates.

Example 10. Consider the fuzzy \mathcal{T} -equivalence relation R on $X = \{a, b\}$ given by

R	a	b
a	1.0	0.2
b	0.2	1.0

and the fuzzy set A in X defined by $A(a) = 1$ and $A(b) = 0.8$. Furthermore, let $\mathcal{T} = \mathcal{T}_M$ and $\mathcal{I} = \mathcal{I}_{S_M, N_S}$. Then, $R\uparrow A(a) = 1$ and $R\uparrow A(b) = 0.8$; hence,

$$(R\downarrow\uparrow A)(a) = \min(\max(0, 1), \max(0.8, 0.8)) = 0.8, \quad (28)$$

which makes it clear that $A \not\subseteq R\downarrow\uparrow A$.

From all of the above we obtain, for any fuzzy relation R in X ,

$$R\downarrow\downarrow A \subseteq R\downarrow A \subseteq R\uparrow\downarrow A \subseteq A \subseteq R\downarrow\uparrow A \subseteq R\uparrow A \subseteq R\uparrow\uparrow A, \quad (29)$$

provided that \mathcal{T} satisfies the conditions of Proposition 9.

24.3.2.2 Interaction with Set-Theoretic Operations

The following proposition shows that, given some elementary conditions on the involved connectives, the usual lower and upper approximation are dual w.r.t. fuzzy set complementation.

Proposition 11 [37]. *If \mathcal{T} is a t-norm, \mathcal{N} an involutive negator, and \mathcal{I} the corresponding S-implicator, or if \mathcal{T} is a left-continuous t-norm, \mathcal{I} its residual implicator, and \mathcal{N} defined by $\mathcal{N}(x) = \mathcal{I}(x, 0)$ for x in $[0, 1]$ is an involutive negator; then*

$$R \uparrow A = co_{\mathcal{N}}(R \downarrow (co_{\mathcal{N}} A)), \quad (30)$$

$$R \downarrow A = co_{\mathcal{N}}(R \uparrow (co_{\mathcal{N}} A)). \quad (31)$$

Combining this result with Proposition 5, it is easy to see that under the same conditions, tight upper and loose lower approximation are dual w.r.t. complementation, as are loose upper and tight lower approximation.

Proposition 12 [13]. *For any fuzzy sets A and B in X ,*

$$R \downarrow(A \cap B) = R \downarrow A \cap R \downarrow B, \quad (32)$$

$$R \uparrow(A \cap B) \subseteq R \uparrow A \cap R \uparrow B, \quad (33)$$

$$R \downarrow(A \cup B) \subseteq R \downarrow A \cup R \downarrow B, \quad (34)$$

$$R \uparrow(A \cup B) = R \uparrow A \cup R \uparrow B. \quad (35)$$

Again by Proposition 5, one can also verify the following equalities:

$$R \downarrow \downarrow(A \cap B) = R \downarrow A \cap R \downarrow B \quad (36)$$

$$R \uparrow \uparrow(A \cup B) = R \uparrow A \cup R \uparrow B, \quad (37)$$

whereas for the remaining interactions, the same inclusions hold as in the crisp case (see Table 24.2).

24.3.2.3 Maximal Expansion and Reduction

Taking an upper approximation of A in practice corresponds to expanding A , while a lower approximation is meant to reduce A . However this refining process does not go on forever. The following property says that with the loose lower and the tight upper approximation maximal reduction and expansion are achieved within one approximation.

Proposition 13 [36]. *If \mathcal{T} is a left-continuous t-norm and \mathcal{I} its residual implicator, then for every fuzzy set A in X ,*

$$R \uparrow \downarrow(R \uparrow \downarrow A) = R \uparrow \downarrow A \quad \text{and} \quad R \downarrow \uparrow(R \downarrow \uparrow A) = R \downarrow \uparrow A. \quad (38)$$

To investigate the behavior of the loose upper and tight lower approximation w.r.t. expansion and reduction, we first establish links with the composition of R with itself. Recall that the composition of fuzzy relations R and S in X is the fuzzy relation $R \circ S$ in X defined by

$$(R \circ S)(x, z) = \sup_{y \in X} \mathcal{T}(R(x, y), S(y, z)) \quad (39)$$

for all x and z in X .

Proposition 14 [33]. *If \mathcal{T} is a left-continuous t-norm, then for every fuzzy set A in X ,*

$$R \uparrow \uparrow A = (R \circ R) \uparrow A. \quad (40)$$

Proposition 15 [33]. If \mathcal{T} is left continuous in its first component and right continuous in its second component, and if \mathcal{T} and \mathcal{I} satisfy the shunting principle

$$\mathcal{I}(\mathcal{T}(x, y), z) = \mathcal{T}(x, \mathcal{I}(y, z)), \quad (41)$$

then for every fuzzy set A in X

$$R \downarrow \downarrow A = (R \circ R) \downarrow A. \quad (42)$$

Note 16. Regarding the restrictions placed on the fuzzy logical operators involved, recall that the shunting principle is satisfied both by a left-continuous t-norm and its residual impicator [38] and by a t-norm and an S-impicator induced by it [13].

Let us use the following notation, for $n > 1$,

$$R^1 = R \quad \text{and} \quad R^n = R \circ R^{n-1}. \quad (43)$$

From Proposition 14 it follows that taking the upper approximation of a fuzzy set under R n times successively corresponds to taking the upper approximation once under the composed fuzzy relation R^n . Proposition 15 states a similar result for the lower approximation. For the particular case of a fuzzy \mathcal{T} -equivalence relation, we have the following important result.

Proposition 17 [13]. If R is a fuzzy \mathcal{T} -equivalence relation in X , then

$$R \circ R = R. \quad (44)$$

In other words, using a \mathcal{T} -transitive fuzzy relation R , options (1a) and (1c) of Definition 4 coincide, as well as options (2b) and (2c). The following proposition states that under these conditions, they also coincide with (1b), respectively (2a).

Proposition 18 [13, 36]. If R is a fuzzy \mathcal{T} -equivalence relation in X , \mathcal{T} is a left-continuous t-norm and \mathcal{I} its residual impicator; then for every fuzzy set A in X

$$R \uparrow \downarrow A = R \downarrow A \quad \text{and} \quad R \downarrow \uparrow A = R \uparrow A. \quad (45)$$

This means that using a fuzzy \mathcal{T} -equivalence relation to model approximate equality, we will obtain maximal reduction or expansion in one phase, regardless of which of the approximations from Definition 4 is used. As Example 2 already illustrated for the crisp case, when we abandon (\mathcal{T})-transitivity, this behavior is not always exhibited. In general, when R is not \mathcal{T} -transitive and the universe X is finite, it is known that the \mathcal{T} -transitive closure of R is given by $R^{|X-1|}$ (assuming $|X| \geq 2$) [39]; hence,

$$R \circ R^{|X-1|} = R^{|X-1|}. \quad (46)$$

In other words with the lower and upper approximation, maximal reduction and expansion will be reached in at most $|X - 1|$ steps, while with the tight lower and the loose upper approximation it can take at most $\lceil |X - 1|/2 \rceil$ steps.

Note 19. The special situation regarding fuzzy \mathcal{T} -equivalence relations deserves some further attention. While they are known as the counterpart of equivalence relations, we illustrated in Section 24.3.1 that their fuzzy similarity classes are not always equal or disjoint; in fact, y can belong at the same time to different fuzzy similarity classes to a certain degree. Hence it is not possible, at first sight, to rule out the usefulness of the tight and loose lower and upper approximations introduced in Definition 4. However, careful investigation of the properties of the approximations shows that interplay between suitably chosen fuzzy logical operators and the \mathcal{T} -transitivity of the fuzzy relation forces the various approximations to coincide. In the next section we will illustrate that this is not always a desirable property in applications, because it does not allow for gradual expansion or reduction of a fuzzy set

by iteratively taking approximations. Omitting the requirement of T -transitivity is precisely the key that allows for a gradual expansion process.

Other undesirable effects of T -transitivity w.r.t. approximate equality were pointed out in [40, 41]. More in particular it is observed there that fuzzy T -equivalence relations can never satisfy the so-called Poincaré paradox. A fuzzy relation R in X is compatible with the Poincaré paradox iff

$$\exists(x, y, z) \in X^3, R(x, y) = 1 \wedge R(y, z) = 1 \wedge R(x, z) < 1. \quad (47)$$

This is inspired by Poincaré's [42] experimental observation that a bag of sugar of 10 g and a bag of 11 g can be perceived as indistinguishable by a human being. The same applies for a bag of 11 g w.r.t. a bag of 12 g, while the subject is perfectly capable of noting a difference between the bags of 10 and 12 g. Now if R is a fuzzy T -equivalence relation, then $R(x, y) = 1$ implies $Rx = Ry$ [3]. Since $Ry(z) = R(y, z) = 1$, also $Rx(z) = R(x, z) = 1$, which is in conflict with $R(x, z) < 1$. The fact that they are not compatible with the Poincaré paradox makes fuzzy T -equivalence relations less suited to model approximate equality. The main underlying cause for this conflict is T -transitivity.

24.4 Application to Query Refinement

One of the most common ways to retrieve information from the WWW is keyword-based search: the user inputs a query consisting of one or more keywords and the search system returns a list of Web documents ranked according to their relevance to the query.² The same procedure is often used in e-commerce applications that attempt to relate the user's query to products from the catalog of some company.

In the basic approach, documents are not returned as search results if they do not contain (one of) the exact keywords of the query. There are various reasons why such an approach might fall short. On one hand there are word mismatch problems: the user knows what he/she is looking for and is able to describe it, but the query terms the user uses do not exactly correspond to those in the document containing the desired information because of differences in terminology. This problem is even more significant in the context of the WWW than in other, more focussed information retrieval applications, because of the very heterogeneous sources of information expressed in different jargon or even in different natural languages.

Besides differences in terminology, it is also not uncommon for a user not to be able to describe accurately what he/she is looking for: the well-known 'I will know it when I see it' phenomenon. Furthermore, many terms in natural language are ambiguous. For example, a user querying for java might be looking for information about either the programming language, the coffee, or the island of Indonesia. To satisfy users who expect search engines to come up with 'what they mean and not what they say,' it is clear that more sophisticated techniques are needed than a straightforward returning of the documents that contain (one of) the query terms given by the user. One option is to adapt the query.

Query refinement has already found its way to popular Web search engines and is even becoming one of those features in which search engines aim to differentiate in their attempts to create their own identity. Simultaneously with search results, Yahoo!³ shows a list of clickable expanded queries in an 'Also Try' option under the search box. These queries are derived from logs containing queries performed earlier by others. Google Suggest⁴ also uses data about the overall popularity of various searches to help rank the refinements it offers, but unlike the other search engines, the suggestions pop up in the search box while you type, i.e., before you search. Ask.com⁵ provides a zoom feature, allowing users to narrow or broaden the field of search results, as well as view results for related concepts.

Since Web queries tend to be short – according to [45] they consist of one or two terms on average – we focus on query expansion, i.e., the process of adding related terms to the query.

² Alternatively, documents can be grouped in clusters. We refer to [43, 44] for a rough set and a fuzzy set approach to clustering of Web search results.

³ <http://search.yahoo.com/>.

⁴ <http://labs.google.com/suggest/>.

⁵ <http://www.ask.com/>.

Query Expansion

Query expansion goes back a long way before the existence of the WWW. Over the last decades several important techniques have been established. The main idea underlying all of them is to extend the query with words related to the query terms. One option is to use an available thesaurus, i.e., a term–term relation, such as WordNet,⁶ expanding the query by adding synonyms [46]. Related terms can also be automatically discovered from the searchable documents though, taking into account statistical information such as cooccurrences of words in documents or in fragments of documents. The more terms cooccur, the more they are assumed to be related. In [45] several of these approaches are discussed and compared. In global document analysis, the whole corpus of searchable documents is preprocessed and transformed into an automatically generated thesaurus. Local document analysis, on the other hand, considers only, the top-ranked documents for the initial query. In its most naive form, terms that appear most frequently in these top-ranked documents are added to the query. Local document analysis is referred to as a pseudorelevance feedback approach, because it tacitly assumes that the highest ranked documents are indeed relevant to the query. A true relevance feedback approach takes into account the documents marked as relevant by the user. Finally, in [47], correlations between terms are computed based on their cooccurrences in query logs instead of in documents.

Once the relationship between terms is known, either through a lexical aid such as WordNet or automatically generated from statistical information, the original query can be expanded in various ways. The straightforward way is to extend the query with all the words that are related to at least one of the query terms. Intuitively, this corresponds to taking the upper approximation of the query. Indeed, a thesaurus characterizes an approximation space in which the query, which is a set of terms, can be approximated from the upper (and the lower) side. By definition, the upper approximation will add a term to the query as soon as it is related to *one* of the words already in the query. This link between query expansion and rough set theory has been established in [48], even involving fuzzy logical representations of the term–term relations and the queries.

In [46], it is pointed out, however, that such an approach requires sense resolution of ambiguous words. Indeed, the precision of retrieved documents is likely to decrease when expanding a query such as *java*, *travel* with the term *applet*. Even though this term is highly related to *java* as a programming language, it has little or nothing to do with the intended meaning of *java* in this particular query, namely, the island. An option to automate sense disambiguation is to add a term only when it is related to at least two words of the original query; experimental results are however unsatisfactory [46].

In [47], the most popular sense gets preference. For example, if the majority of users use windows to search for information about the Microsoft product, the term *windows* has much stronger correlations with terms such as Microsoft, OS, and software, rather than with terms such as decorate, door, and house. The approaches currently taken by Yahoo! and Google Suggest seem to be in line with this principle. Note, however, that these search engines do not apply query expansion automatically but leave the final decision up to the user.

In [49], a virtual term is created to represent the general concept of the query. Terms are selected for expansion based on their similarity to this virtual term. In [45], candidate expansion terms are ranked based on their cooccurrence with all query terms in the top-ranked documents.

Finding the Right Balance

The approach discussed below, first introduced in [50] and taken up also in [51], differs from all techniques mentioned above and takes into account the lower approximation as well. The lower approximation only retain a term in the query will if *all* the words that it is related too are also in the query. It is obvious that the lower approximation will easily result in the empty query; hence, in practice it is often too strict for query refinement. On the other hand, it is not hard to imagine cases where the upper approximation is too flexible as a query expansion technique, resulting not only in an explosion of the query, but possibly even worse, in the addition of non-relevant terms due to the ambiguous nature of one or more of the query

⁶ <http://wordnet.princeton.edu/>.

Table 24.6 Graded thesaurus

words. This is due to the fact that the upper approximation expands each of the query words individually but disregards the query as a whole.

However, it is possible to combine the flexibility of the upper approximation with the strictness of the lower approximation by applying them successively. As such, first the query is expanded by adding all the terms that are known to be related to at least one of the query words. Next, the expanded query is reduced by taking its lower approximation, thereby pruning away all previously added terms that are suspected to be irrelevant for the query. The pruning strategy targets those terms that are strongly related to words that do not belong to the expanded query.

This technique can be used both with a crisp thesaurus in which terms are related or not and with a graded thesaurus in which terms are related to some degree. Furthermore, it can be applied for weighted as well as for non-weighted queries. Whenever the user does not want to go through the effort of assigning individual weights to query terms, he/she is all given the highest weight by default. When a graded thesaurus is used, the query refinement automatically turns the original query into a weighted query. The original user-chosen terms maintain their highest weight, and new terms are added with weights that do not only reflect the strength of the relationship with the original individual query terms as can be read from the thesaurus, but also take into account their relevance to the query as a whole. To be able to deal with graded thesauri and weighted queries and apply the machinery of fuzzy rough sets, we represent the thesaurus as a fuzzy relation and the query as a fuzzy set.

Example 20. Table 24.6 shows a small sample fuzzy thesaurus R based on the cooccurrences of the terms in Web pages found by Google. More details on the construction can be found in [50].

The T_W -transitive closure $R^{|X-1|}$ of R , i.e., the smallest T_W -transitive fuzzy relation in which R is included, is shown in Table 24.7. In our running example, to compute upper and lower approximations, we will keep on using the t -norm T_W as well as its residual impicator I_{T_W} .

Table 24.7 Transitive closure of graded thesaurus

Table 24.8 Crisp thesaurus

$R_{0.5}$	Mac	Computer	Apple	Fruit	Pie	Recipe	Store	Emulator	Hardware
Mac	1	1	1	0	0	0	1	1	1
Computer		1		1	0	0	1	0	1
Apple			1	1	1	1	1	0	1
Fruit				1	0	1	1	0	0
Pie					1	1	1	0	0
Recipe						1	1	0	0
Store							1	0	1
Emulator								1	1
Hardware									1

Finally, a crisp (i.e., non-graded) thesaurus can be construed by taking the 0.5-level of R , defined as

$$(x, y) \in R_{0.5} \text{ iff } R(x, y) \geq 0.5 \quad (48)$$

for all x and y in X . In other words, in the crisp thesaurus, depicted in Table 24.8, two terms are related if and only if the strength of their relationship in the graded thesaurus R of Table 24.6 is at least 0.5. It can be easily verified that $R_{0.5}$ is not transitive. For example, fruit is related to store and store is related to hardware, but fruit is not related to hardware. For comparison purposes, in the remainder, we also include the transitive closure ($R_{0.5}^8$)⁸.

Consider the query

apple, pie, recipe

as shown in the second column in Table 24.9 under the heading A. The intended meaning of the ambiguous word apple, which can refer both to a piece of fruit and to a computer company, is clear in this query.

The disadvantage of using a T -transitive fuzzy thesaurus becomes apparent when we compute the upper approximation $R^8 \uparrow A$, shown in the last column. All the terms are added with high degrees, even though terms like mac and computer have nothing to do with the semantics of the original query. This process can be slowed down a little bit by using the non- T -transitive fuzzy thesaurus and computing $R \uparrow A$, which allows for some gradual refinement. However an irrelevant term such as emulator shows up to a high degree in the second iteration, i.e., when computing $R \uparrow (R \uparrow A)$. The problem is even more prominent when using a crisp thesaurus as shown in Table 24.10.

Table 24.9 Upper-approximation-based query expansion with graded thesaurus

	A	$R \uparrow A$	$R \uparrow (R \uparrow A)$	$R^8 \uparrow A$
Mac	0.00	0.89	0.89	0.89
Computer	0.00	0.94	0.94	0.99
Apple	1.00	1.00	1.00	1.00
Fruit	0.00	0.83	1.00	1.00
Pie	1.00	1.00	1.00	1.00
Recipe	1.00	1.00	1.00	1.00
Store	0.00	1.00	1.00	1.00
Emulator	0.00	0.25	0.99	0.99
Hardware	0.00	0.99	0.99	0.99

Table 24.10 Upper-approximation-based query expansion with crisp thesaurus

	A	$R_{0.5} \uparrow A$	$R_{0.5} \uparrow (R_{0.5} \uparrow A)$	$(R_{0.5})^8 \uparrow A$
Mac	0	1	1	1
Computer	0	1	1	1
Apple	1	1	1	1
Fruit	0	1	1	1
Pie	1	1	1	1
Recipe	1	1	1	1
Store	0	1	1	1
Emulator	0	0	1	1
Hardware	0	1	1	1

It is important to point out that under our assumptions

$$A \subseteq R \downarrow \uparrow A \subseteq R \uparrow A \quad (49)$$

always holds, guaranteeing that the tight upper approximation indeed leads to an expansion of the query – none of the original terms are lost – and at the same time is a pruned version of the upper approximation. When R is a fuzzy T-equivalence relation, the upper approximation and the tight upper approximation coincide (see Table 24.5). However, as we show below, this is not necessarily the case when R is not T-transitive.

The main problem with the query expansion process as used in the previous example, even if it is gradual, is a fast growth of the number of less relevant or irrelevant keywords that are automatically added. This effect is caused by the use of a flexible definition of the upper approximation in which a term is added to a query as soon as it is related to one of its keywords. However, using the tight upper approximation, a term y will only be added to a query A if all the terms that are related to y are also related to at least one keyword of the query. First the usual upper approximation of the query is computed, but then it is stripped down by omitting all terms that are also related to other terms not belonging to this upper approximation. In this way terms that are sufficiently relevant, hence related to most keywords in A, will form a more or less closed context with few or no links outside, while a term related to only one of the keywords in A in general also has many links to other terms outside $R \uparrow A$ and hence is omitted by taking the lower approximation.

Example 21. *The last column of Table 24.11 shows that the tight upper approximation is different from and performs clearly better than the traditional upper approximation for our purpose of Web query*

Table 24.11 Comparison of upper- and tight-upper-approximation-based query expansion with graded thesaurus

	A	$R \uparrow A$	$R^8 \uparrow A$	$R \downarrow \uparrow A$
Mac	0.00	0.89	0.89	0.42
Computer	0.00	0.94	0.99	0.25
Apple	1.00	1.00	1.00	1.00
Fruit	0.00	0.83	1.00	0.83
Pie	1.00	1.00	1.00	1.00
Recipe	1.00	1.00	1.00	1.00
Store	0.00	1.00	1.00	0.83
Emulator	0.00	0.25	0.99	0.25
Hardware	0.00	0.99	0.99	0.25

Table 24.12 Comparison of upper- and tight-upper-approximation-based query expansion with crisp thesaurus

	A	$R_{0.5} \uparrow A$	$(R_{0.5})^8 \uparrow A$	$R_{0.5} \downarrow A$
Mac	0	1	1	0
Computer	0	1	1	0
Apple	1	1	1	1
Fruit	0	1	1	1
Pie	1	1	1	1
Recipe	1	1	1	1
Store	0	1	1	1
Emulator	0	0	1	0
Hardware	0	1	1	0

expansion: irrelevant words such as mac, computer, and hardware are still added to the query, but to a significantly lower degree. The difference becomes even more noticeable when using a crisp thesaurus as illustrated in Table 24.12.

24.5 Summary

Fuzzy sets and rough sets each address an important characteristic of imperfect data and knowledge: while the former allow that objects belong to a set or relation to a given degree, the latter provide approximations of concepts in the presence of incomplete information. Fuzzy rough set theory aims to combine the best of both worlds. At the heart of this synergy lie well-chosen definitions of lower and upper approximations of fuzzy sets under fuzzy relations.

In a traditional Pawlak approximation space, indistinguishability is described by means of an equivalence relation R . Well-known properties of the lower approximation $R \downarrow A$ and the upper approximation $R \uparrow A$ are recalled in Table 24.1. In a generalized approximation space, characterized by a general binary relation R , the tight lower approximation $R \downarrow \downarrow A$, the loose lower approximation $R \uparrow \uparrow A$, the tight upper approximation $R \downarrow \uparrow A$, and the loose upper approximation $R \uparrow \downarrow A$ are useful supplements to the traditional approximations. Their properties are summarized in Table 24.2.

All of these approximations can be generalized to approximate fuzzy sets in an approximation space characterized by a fuzzy tolerance relation R . The preservation of previous properties depends on a careful choice of the fuzzy logical operators involved, as becomes clear in Table 24.5. It is especially interesting to note that when R is \mathcal{T} -transitive, \mathcal{T} is a continuous t -norm and $\mathcal{I}_{\mathcal{T}}$ its residual implicator, then all three lower approximations coincide, as to do all three upper approximations. However, this coincidence, and hence the \mathcal{T} -transitivity of R , is not always desirable in applications, as an example about query refinement illustrates.

In this application, a query is perceived as a fuzzy set of terms, while R is a fuzzy tolerance relation among terms. The upper approximation turns out to be too flexible as a query expansion technique, resulting in the addition of irrelevant terms when one or more of the original query terms are ambiguous. The lower approximation easily results in the empty query; hence, in practice it is too strict for query refinement. In the example given in this chapter, the right balance is found in the tight upper approximation.

Acknowledgment

Chris Cornelis thanks the Research Foundation–Flanders for funding his research.

References

- [1] L.A. Zadeh. Fuzzy sets. *Inf. Control* 8 (1965) 338–353.
- [2] Z. Pawlak. Rough sets. *Int. J. Comput. Inf. Sci.* 11 (5) (1982) 341–356.
- [3] D. Dubois and H. Prade. Rough fuzzy sets and fuzzy rough sets. *Int. J. Gen. Syst.* 17 (1990) 91–209.
- [4] S. Greco, B. Matarazzo, and R. Slowinski. Fuzzy similarity relation as a basis for rough approximations. In: L. Polkowski and A. Skowron (eds), *Rough Sets and Current Trends in Computing, Lecture Notes in Artificial Intelligence*, Vol. 1424. Springer-Verlag, Berlin, Germany, 1998, pp. 283–289.
- [5] S. Greco, B. Matarazzo, and R. Slowinski. The use of rough sets and fuzzy sets in multiple-criteria decision making. In: T. Gal, T. Stewart, and T. Hanne (eds), *Advances in Multiple Criteria Decision Making*. Kluwer Academic Publishers, Boston, 1999, pp. 14.1–14.59.
- [6] S. Greco, B. Matarazzo, and R. Slowinski. Rough set processing of vague information using fuzzy similarity relations. In: C.S. Calude and G. Paun (eds), *Finite versus Infinite – Contributions to an Eternal Dilemma*. Springer-Verlag, London, 2000, pp. 149–173.
- [7] S. Greco, B. Matarazzo, and R. Slowinski. Fuzzy extension of the rough set approach to multicriteria and multiattribute sorting. In: J. Fodor, B. De Baets, and P. Perny (eds), *Preferences and Decisions under Incomplete Knowledge*. Physica-Verlag, Heidelberg, 2000, pp. 131–151.
- [8] L.I. Kuncheva. Fuzzy rough sets: Application to feature selection. *Fuzzy Sets Syst.* 51 (1992) 147–153.
- [9] T.Y. Lin. Topological and fuzzy rough sets. In: R. Slowinski (ed), *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer Academic Publishers, Boston, 1992, pp. 287–304.
- [10] N.N. Morsi and M.M. Yakout. Axiomatics for fuzzy rough sets. *Fuzzy sets Syst.* 100 (1–3) (1998) 327–342.
- [11] A. Nakamura. Fuzzy rough sets. *Note on Mult.-Valued Log. Japan* 9 (1988) 1–8.
- [12] S. Nanda and S. Majumdar. Fuzzy rough sets. *Fuzzy Sets Syst.* 45 (1992) 157–160.
- [13] A.M. Radzikowska and E.E. Kerre. A comparative study of fuzzy rough sets. *Fuzzy Sets Syst.* 126 (2002) 137–156.
- [14] R. Slowinski and J. Stefanowski. Rough set reasoning about uncertain data. *Fundam. Inf.* 27 (1996) 229–243.
- [15] H. Thiele. *Fuzzy Rough Sets versus Rough Fuzzy Sets – An Interpretation and a Comparative Study Using Concepts of Modal Logic*. Technical Report ISSN 1433-3325. University of Dortmund, Dortmund, Germany, 1998.
- [16] Y.Y. Yao. Combination of rough and fuzzy sets based on alpha-level sets. In: T.Y. Lim and N. Cercone (eds), *Rough Sets and Data Mining: Analysis for Imprecise Data*. Kluwer Academic Publishers, Boston, 1997, pp. 301–321.
- [17] T.B. Iwinski. Algebraic approach to rough sets. *Bull. Pol. Acad. Sci. Math.* 35 (1987) 673–683.
- [18] Y.Y. Yao. Two views of the theory of rough sets in finite universes. *Int. J. Approx. Reason.* 15 (4) (1996) 291–317.
- [19] Z. Pawlak and A. Skowron. Rudiments of rough sets. *Inf. Sci.* 177(1) (2007) 3–27.
- [20] Z. Pawlak and A. Skowron. Rough sets: Some extensions. *Inf. Sci.* 177(1) (2007) 28–40.
- [21] Z. Pawlak and A. Skowron. Rough sets and boolean reasoning. *Inf. Sci.* 177(1) (2007) 41–73.
- [22] S. Greco, B. Matarazzo, and R. Slowinski. Fuzzy set extensions of the dominance-based rough set approach. In: H. Bustince, F. Herrera, and J. Montero (eds), *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models, Intelligent Systems from Decision Making to Data Mining, Web Intelligence and Computer Vision, Studies in Fuzziness and Soft Computing*. Springer-Verlag, Berlin, to appear.
- [23] S. Greco, M. Inuiguchi, and R. Slowinski. A new proposal for fuzzy rough approximations and gradual decision rule representation. *Trans. Rough Sets II Lect. Notes Comput. Sci.* 3135 (2004) 319–342.
- [24] S. Greco, M. Inuiguchi, and R. Slowinski. Fuzzy rough sets and multiple-premise gradual decision rules. *Int. J. Approx. Reason.* 41 (2005) 179–211.
- [25] G. Cattaneo. Abstract approximation spaces for rough theories. In: L. Polkowski, and A. Skowron (eds), *Rough Sets in Knowledge Discovery 1: Methodology and Applications*. Physica-Verlag, Heidelberg, 1998, pp. 59–98.
- [26] G. Cattaneo and D. Ciucci. Algebraic structures for rough sets. *Lect. Notes Comput. Sci.* 3135 (2004) 218–264.
- [27] J.A. Pomykala. Approximation operations in approximation space. *Bull. Pol. Acad. Sci. Math.* 35 (1987) 653–662.
- [28] A. Skowron and J. Stepaniuk. Tolerance approximation spaces. *Fundam. Inf.* 27(2/3) (1996) 245–253.
- [29] R. Slowinski and D. Vanderpooten. Similarity relation as a basis for rough approximations. In: P.P. Wang (ed), *Advances in Machine Intelligence and Soft-Computing*, Vol. IV. Duke University Press, Durham, NC, 1997, pp. 17–33.
- [30] R. Slowinski and D. Vanderpooten. A generalized definition of rough approximations based on similarity. *IEEE Trans. Data Knowl. Eng.* 12(2) (2000) 331–336.

- [31] L. Fariñas del Cerro and H. Prade. Rough sets, twofold fuzzy sets and modal logic – fuzziness in indiscernibility and partial information. In: A. Di Nola and A.G.S. Ventre (eds), *The Mathematics of Fuzzy Systems*. Verlag TUV Rheinland, Köln, 1986, pp. 103–120.
- [32] W.N. Liu, J.T. Yao, and Y.Y. Yao. Rough approximations under level fuzzy sets. *Lect. Notes Artif. Intell.* 3066 (2004) 78–83.
- [33] M. De Cock, C. Cornelis, and E.E. Kerre. Fuzzy rough sets: The forgotten step. *IEEE Trans. Fuzzy Syst.* 15(1) (2007) 121–130.
- [34] B. De Baets and R. Mesiar. Pseudo-metrics and \mathcal{T} -equivalences. *J. Fuzzy Math.* 5 (1997) 471–481.
- [35] M. De Cock, C. Cornelis, and E.E. Kerre. Fuzzy rough sets: Beyond the obvious. In: *Proceedings of the 2004 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'04*, Vol. 1, Budapest, Hungary, July 25–29, 2004, pp. 103–108.
- [36] U. Bodenhofer. A unified framework of opening and closure operators with respect to arbitrary fuzzy relations. *Soft Comput.* 7 (2003) 220–227.
- [37] M. De Cock. *A Thorough Study of Linguistic Modifiers in Fuzzy Set Theory*. Ph.D. Thesis (in Dutch). Ghent University, Flanders, Belgium, 2002.
- [38] V. Novák, I. Perfilieva, and J. Močkovič. *Mathematical Principles of Fuzzy Logic*. Kluwer Academic Publishers, Boston, 1999.
- [39] H. Naessens, H. De Meyer, and B. De Baets. Algorithms for the computation of \mathcal{T} -transitive closures. *IEEE Trans. Fuzzy Syst.* 10(4) (2002) 541–551.
- [40] M. De Cock and E.E. Kerre. On (un)suitable fuzzy relations to model approximate equality. *Fuzzy Sets Syst.* 133(2) (2003) 137–153.
- [41] M. De Cock and E.E. Kerre. Why fuzzy \mathcal{T} -equivalence relations do not resolve the Poincaré paradox, and related issues. *Fuzzy Sets Syst.* 133(2) (2003) 181–192.
- [42] H. Poincaré. *La science et l'hypothèse*. Flammarion, Paris, 1902.
- [43] C.L. Ngo and H.S. Nguyen. A tolerance rough set approach to clustering web search results. *Lect. Notes Comput. Sci.* 3202 (2004) 515–517.
- [44] S. Schockaert, M. De Cock, C. Cornelis, and E.E. Kerre. Clustering web search results using fuzzy ants. *Int. J. Intell. Syst.* 22(5) (2007) 455–474.
- [45] J. Xu and W.B. Croft. Query expansion using local and global document analysis. In: *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM SIGIR 1996*, Zürich, Switzerland, August 18–22, 1996, pp. 4–11.
- [46] E.M. Voorhees. Query expansion using lexical-semantic relations. In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM SIGIR 1994*, Dublin, Ireland, July 3–6, 1994, pp. 61–69.
- [47] H. Cui, J.R. Wen, J.Y. Nie, and W.Y. Ma. Probabilistic query expansion using query logs. In: *Proceedings of the 11th International World Wide Web Conference, WWW'02*, ACM Press, New York, 2002, pp. 325–332.
- [48] P. Srinivasan, M.E. Ruiz, D.H. Kraft, and J. Chen. Vocabulary mining for information retrieval: Rough Sets and fuzzy Sets. *Inf. Process. Manage.* 37 (2001) 15–38.
- [49] Y. Qui and H. Frei. Concept based query expansion. In: *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM SIGIR 1993*, Pittsburgh, PA, June 27–July 1, 1993, pp. 160–169.
- [50] M. De Cock and C. Cornelis. Fuzzy rough set based web query expansion. In: *Proceedings of the International Workshop on Rough Sets and Soft Computing in Intelligent Agent and Web Technology*, Compiègne, France, September 19–22, 2005. 2005, pp. 9–16.
- [51] S. Tenreiro de Magalhães, L. Duarte dos Santos, and L. Amaral. Getting the knowledge to the agent the rough sets approach. In: *Proceedings of the 8th International Conference on Current Research Information Systems, CRIS'06*, Bergen, Norway, May 11–13, 2006, pp. 155–166.

25

On Type 2 Fuzzy Sets as Granular Models for Words

Jerry M. Mendel

25.1 Introduction

Words, as modeled by fuzzy sets (FSs), are used in approximate reasoning, rule-based systems, aggregation, summarizations, etc. An FS model for a word is a granulation of the word. Because there are different kinds of FSs, there can be different kinds of granulations of a word, and because words can mean different things to different people, an FS model should be chosen that directly handles (model and minimize the effects of) word uncertainties. Type 2 (T2) FSs can do this. In fact, an interval Type 2 (IT2) FS captures first-order uncertainties about a word [1], whereas a general T2 FS captures both first- and second-order uncertainties. Using such models in any of the above applications, let word uncertainties flow from the application's front end to its output. This is accomplished by using the mathematics of T2 FSs. While this mathematics leads to computations that are today intractable for general T2 FSs, it leads to computations that are today very tractable for IT2 FSs.

T2 FSs are used in rule-based fuzzy logic systems (FLS) to also model the effects of uncertainty about the consequent that is used in a rule, measurements that activate the FLS, and data that are used to tune the parameters of an FLS.

In this chapter, which is not meant to be a tutorial about Type 2 fuzzy sets and systems (for such a tutorial, see [2]), a wealth of material is summarized about T2 FSs, especially IT2 FSs. For a very interesting historical view of T2 fuzzy sets and systems, see [3, 4]. In order to make important results readily accessible to the reader, they are summarized in tables, some of which contrast results for both general and IT2 FSs, whereas others only give results for IT2 FSs. Additionally, Zadeh's computing with words (CWW) paradigm and what the elements are that are needed to implement it are explained, because granulation of words as IT2 FSs is central to CWW. Finally, many existing applications of T2 FSs are tabulated.

25.2 Definitions and Terminology

The concept of a T2 FS was first introduced by Zadeh [5] as an extension of the concept of an ordinary FS, i.e., a T1 FS. T2 FSs have grades of membership that are themselves fuzzy (so, they could be called *fuzzy-fuzzy sets*). At *each value* of the primary variable (e.g., pressure and temperature), the membership is a function (and not just a point value) – the *secondary MF* – whose domain – the *primary membership* – is

in the interval $[0, 1]$ and whose range – *secondary grades* – may also be in $[0, 1]$. Hence, the MF of a T2 FS is three-dimensional, and it is the new third dimension that provides new design degrees of freedom for handling uncertainties [6, 7]. Such sets are useful in circumstances where it is difficult to determine the exact MF for an FS, as in modeling a word by an FS.

As an example [2, 7], suppose the variable of interest is *eye contact*, which is denoted x . Put *eye contact* on a scale of values 0–10. One of the terms that might characterize the amount of perceived *eye contact* (e.g., during flirtation, or an airport security check) is ‘some eye contact.’ Suppose that 100 people are surveyed and are asked to locate the ends of an interval for *some eye contact* on the scale 0–10. Surely, the same results will not be obtained from all of them, because *words mean different things to different people*.

One approach to using the 100 sets of two endpoints is to average the endpoint data and use the average values for the interval associated with *some eye contact*. A triangular (other shapes could be used) MF, $MF(x)$, could then be constructed, one whose base endpoints (on the x -axis) are at the two average values and whose apex is midway between the two endpoints. This type 1 triangular MF can be displayed in two dimensions. Unfortunately, it has completely ignored the uncertainties associated with the two endpoints.

A second approach is to make use of the average values and the standard deviations for the two endpoints. By doing this, the location of the two endpoints is blurred along the x -axis. Now triangles can be located so that their base endpoints are anywhere in the intervals along the x -axis associated with the blurred average endpoints. Doing this leads to a continuum of triangular MFs sitting on the x -axis, e.g., picture a whole bunch of triangles all having the same apex point but different base points, as in Figure 25.1. For purposes of this discussion, suppose there are exactly N such triangles. Then at each value of x , there can be up to N MF values, $MF_1(x), MF_2(x), \dots, MF_N(x)$. Assign a weight to each of the possible MF values, say $w_{x1}, w_{x2}, \dots, w_{xN}$ (see Figure 25.1). These weights can be thought of as the *possibilities* associated with each triangle at this value of x . At each x , the MF is itself a function – the secondary MF – $(MF_i(x), w_{xi})$, where $i = 1, \dots, N$. Consequently, the resulting T2 MF is three-dimensional.

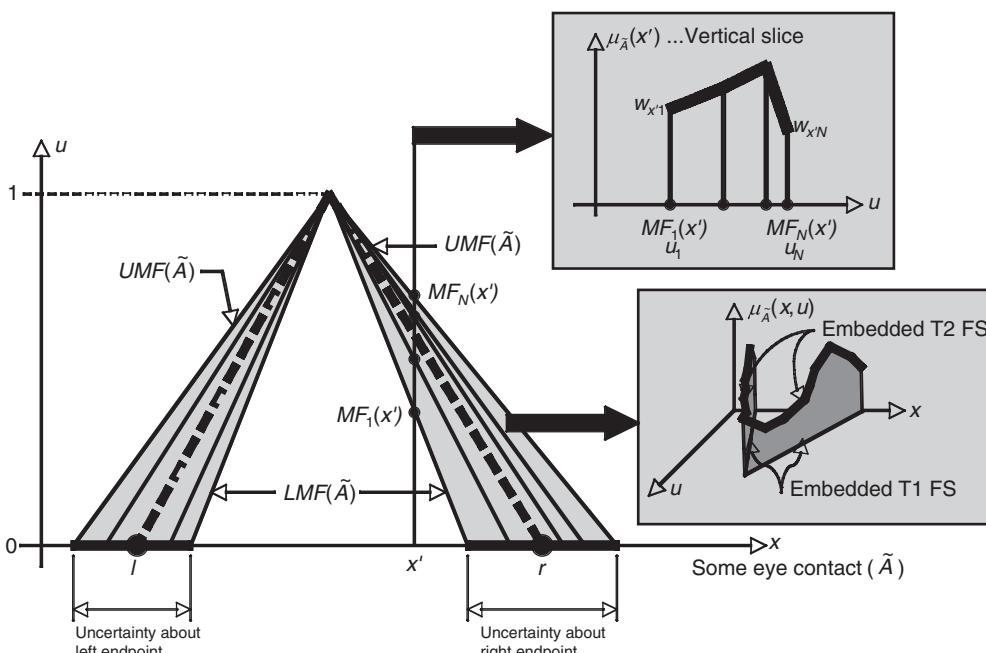


Figure 25.1 Triangular MFs when base endpoints (l and r) have uncertainty intervals associated with them. This is not a unique construction. The resulting footprint of uncertainty (FOU) is shaded. The top insert depicts the secondary MF (vertical slice) at x' and the lower insert depicts the embedded T1 and T2 FSs, the latter called a *wavy slice* (Mendel [2], © 2007, IEEE)

Because it is difficult to draw such three-dimensional MFs, they are rarely shown or used. Instead, it is customary to show the shaded domain for the three-dimensional MF, i.e., the *domain of uncertainty*.

Just as probability has much new terminology and definitions that must be learned in order to use it as a model of unpredictability, T2 FSs have new terminology and definitions that must be learned in order to use it as a model of linguistic uncertainty. The following new terms and definitions are summarized in Table 25.1 for both T2 and IT2 FSs: primary variable, primary membership, secondary variable, secondary

Table 25.1 T2 FS definitions [6, 8]

General T2 FSs	Interval T2 FSs (IT2 FS)
$x \equiv \text{primary variable}, x \in X$ $J_x \equiv \text{primary membership of } x, J_x \subseteq [0, 1]$ $u \equiv \text{secondary variable}, u \in J_x$ $f_x(u) \equiv \text{secondary grade at } x; \text{ an arbitrary function of } u, \forall u \in J_x; 0 \leq f_x(u) \leq 1$	$f_x(u) \equiv \text{secondary grade at } x;$ $f_x(u) = 1, \forall u \in J_x$ $T2 FS^a: \tilde{A} = \{(x, u), \mu_{\tilde{A}}(x, u)\} \forall x \in X, \forall u \in J_x\}$ $\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u)/(x, u)$
<i>Secondary MF</i> (also called a vertical slice):	<i>Secondary MF</i> (also called a vertical slice):
$\mu_{\tilde{A}}(x) = \int_{u \in J_x} f_x(u)/u$	$\mu_{\tilde{A}}(x) = \int_{u \in J_x} 1/u$
<i>Vertical-slice representation of T2 FS:</i>	<i>Vertical-slice representation of IT2 FS:</i>
$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \forall x \in X\} = \int_{x \in X} \mu_{\tilde{A}}(x)/x$ $= \int_{x \in X} \left[\int_{u \in J_x} f_x(u)/u \right] / x$	$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \forall x \in X\} = \int_{x \in X} \mu_{\tilde{A}}(x)/x$ $= \int_{x \in X} \left[\int_{u \in J_x} 1/u \right] / x$
<i>Embedded T2 FS, \tilde{A}_e:</i>	<i>Embedded IT2 FS, \tilde{A}_e:</i>
$\tilde{A}_e = \int_{x \in X} [f_x(\theta)/\theta]/x, \quad \theta \in J_x$	$\tilde{A}_e = \int_{x \in X} [1/\theta]/x, \quad \theta \in J_x$
<i>Embedded TI FS, A_e:</i> $A_e = \int_{x \in X} \theta/x \quad \theta \in J_x$; acts as the domain for \tilde{A}_e	
<i>Primary MF^b:</i> $\mu_A(x p_1 = p_1', p_2 = p_2', \dots, p_v = p_v')$, where $p_i' \in P_i, i = 1, 2, \dots, v$	
<i>Type 2 fuzzy singleton</i>	
$\mu_{\tilde{A}}(x, u) = \begin{cases} 1/1 & \forall x = x' \\ 1/0 & \forall x \neq x' \end{cases}$	
<i>Interval T2 FSs</i>	
<i>Footprint of uncertainty (FOU) of \tilde{A}:</i> $FOU(\tilde{A}) = \bigcup_{x \in X} J_x = \{(x, u) : u \in J_x \subseteq [0, 1]\}$	
<i>Lower MF of \tilde{A}, $\mu_{\tilde{A}}(x)$:</i> $\mu_{\tilde{A}}(x) = \underline{FOU(\tilde{A})}$	
<i>Upper MF of \tilde{A}, $\bar{\mu}_{\tilde{A}}(x)$:</i> $\bar{\mu}_{\tilde{A}}(x) = \overline{FOU(\tilde{A})}$	
$FOU(\tilde{A}) = \bigcup_{x \in X} [\mu_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)]$	

^a $\mu_{\tilde{A}}(x, u) : X \rightarrow [0, 1]^{[0,1]}$.

^b p_i is a parameter whose numerical value p'_i must be chosen from a set of values P_i so that the MF is completely defined.

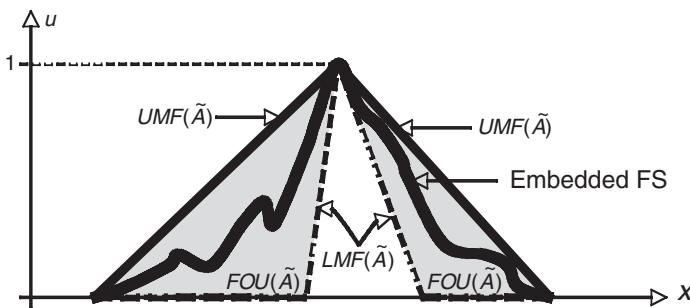


Figure 25.2 T2 FS and associated quantities

grade, T2 FS, IT2 FS, secondary membership function (MF), vertical-slice representation, embedded T2 FS, embedded IT2 FS, primary MF, and T2 fuzzy singleton. Additionally, the following terms and definitions are summarized for IT2 FS \tilde{A} : footprint of uncertainty [$FOU(\tilde{A})$], lower MF [$LMF(\tilde{A})$], and upper MF [$UMF(\tilde{A})$]. Note that in order to distinguish a T2 FS from a T1 FS, a tilde is used over the former, e.g., \tilde{A} . Note also that for a general T2 FS \tilde{A} , $FOU(\tilde{A}) \Leftrightarrow \text{domain of uncertainty } (\tilde{A})$.

A general T2 FS has a three-dimensional (3D) MF. An IT2 FS also has a 3D MF; however, because all of its secondary grades equal 1, there is no information in the third dimension. Hence, an IT2 FS is completely characterized by its 2D FOU, and the FOU is bounded by its LMF and UMF (Figure 25.2). Examples of some representative FOUs are summarized in Table 25.2. Each of these FOUs can be constructed by starting with a primary MF (a T1 FS; see Table 25.1) and assigning some uncertainties to one or more of its parameters; e.g., the FOU for the Gaussian with uncertain mean is obtained by centering a Gaussian at the origin and then sliding it to the left and to the right by $\pm\mu$.

25.3 Important Representations of a T2 FS

There are two very important representations of a T2 FS. The *vertical-slice representation*, stated in Table 25.1, is the basis for most computations, whereas the *wavy-slice representation*, stated in Table 25.3, is the basis for most theoretical derivations. The latter, which states that \tilde{A} is the union of all of its embedded T2 FSs, is also known as the *Mendel–John representation theorem* (RT) [8].

Although the RT is extremely useful for theoretical developments, it is not useful for computation, because the number of embedded sets in the union can be astronomical. Typically, the RT is used to arrive at the structure of a theoretical result (e.g., the union of two IT2 FSs, the centroid of an IT2 FS, etc.), after which practical computational algorithms are found to compute the structure.

For an IT2 FS, the RT states that an IT2 FS is the union of all of the embedded T1 FSs that cover its FOU. The importance of this result is that *it lets us derive everything about IT2 FSs or systems that use them using T1 FS mathematics* [9]. This results in a tremendous savings in learning time for everyone.

25.4 Operations on T2 FSs

Just as the set-theoretic operations of union, intersection, and complement and the arithmetic operations of addition and multiplication are performed on T1 FSs, they are also performed on T2 FSs. Important early works on these operations are [10, 11]. All operations are summarized in Table 25.4. For T2 FSs, the names ‘join (\sqcup)’, ‘meet (\sqcap)’, and ‘negation (\neg)’ are associated with union, intersection, and complement, respectively. Each of the former (e.g., join) is used to compute a vertical slice (at a specific value of primary variable x) of the latter (e.g., union). The RT can be used to derive all of the formulas in this table [8, 9]. Although iterative algorithms are available for computing the join and meet of general T2 FSs when minimum t -norm is used, no such algorithms exist to date when product t -norm is used. Formulas

Table 25.2 Some representative FOUs

General FOU	Special cases	
UMF: Trapezoidal LMF: Triangular	$h=1$ and $a=c$	$c=0$ and $h=1$
UMF: Trapezoidal LMF: Trapezoidal	Triangular FOU segments	
Gaussian with uncertain mean $\in [-\mu, \mu]$ and uncertain standard deviation $\in [\sigma_1, \sigma_2]$	Gaussian with uncertain standard deviation $\in [\sigma_1, \sigma_2]$	Gaussian with uncertain mean $\in [-\mu, \mu]$

are very simple for IT2 FSs and involve only interval arithmetic. Much research is under way to develop practical ways to perform these operations for general T2 FSs (e.g., [14, 15]).

25.5 Fuzzy Logic Systems

A general T2 fuzzy logic system (also called a *fuzzy system* or *fuzzy logic controller*) is depicted in Figure 25.3 [6, 16]. It is very similar to a T1 FLS, the major structural difference being that the defuzzifier of a T1 FLS is replaced by *output processing* in a T2 FLS. The latter consists of *type reduction* (TR) followed by *defuzzification*.

Consider a T2 FLS having p inputs $x_1 \in X_1, \dots, x_p \in X_p$, one output $y \in Y$, and M rules, where the l th rule has the form R^l : IF x_1 is \tilde{F}_1^l and \dots and x_p is \tilde{F}_p^l , THEN y is \tilde{G}^l ($l = 1, \dots, M$). This rule represents a T2 relation between the input space $X_1 \times \dots \times X_p$, and the output space Y of the T2 FLS, and each rule is interpreted as a T2 fuzzy implication.

General formulas for a T2 FLS, which can be used to compute the MF of each fired rule, are summarized in the top portion of Table 25.5. They involve the meet and join operations. Simplifications occur when each input variable is modeled using singleton fuzzification, and are shown in the bottom portion of this

Table 25.3 Wavy-slice representations [8, 9]

$X = \{x_1, x_2, \dots, x_N\}$: Discrete or discretized primary variable	
$u_i^j \in \underbrace{\{\mu_{\tilde{A}}(x_i), \dots, \bar{\mu}_{\tilde{A}}(x_i)\}}_{M_i \text{ elements } (i=1, \dots, N)}$: Sampled secondary variables	
$j = 1, \dots, n_A = \prod_{i=1}^N M_i$: Number of embedded sets	
Kind of T2 fuzzy set	Representation theorem
General T2 FS	$\tilde{A} = \bigcup_{j=1}^{n_A} \tilde{A}_e^j$, where $\tilde{A}_e^j = \sum_{i=1}^N \left[f_{x_i}(u_i^j) / u_i^j \right] / x_i$
Interval T2 FS	$\tilde{A} = 1/\text{FOU}(\tilde{A}) = 1/\bigcup_{j=1}^{n_A} A_e^j$, where $A_e^j = \sum_{i=1}^N u_i^j / x_i$ and this notation means that the secondary grade equals 1 for all elements of $\text{FOU}(\tilde{A})$

table. Even so, it is still very difficult to carry out the meet computations, especially if product t -norm is used. In a T2 FLS when a rule fires, a T2 *firing set* is generated, whereas in a T1 FLS, a *firing level* is generated. The T2 firing set propagates the antecedent MF uncertainties into the meet calculation that involves it and the T2 consequent set. For singleton fuzzification, the T2 firing set reduces to a T1 FS; however, the MF of the l th fired rule is still a T2 FS.

Tremendous simplifications occur when IT2 FSs are used, and those are summarized in Table 25.6 for a Mamdani FLS. This table provides results not only for an IT2 FLS but also, for comparative purposes, for a T1 FLS. Note that the input to a T1 FLS can be modeled in two ways, either as a fuzzy singleton (singleton fuzzification) or as a T1 FS (non-singleton fuzzification); however, the input to an IT2 FLS

Table 25.4 Operations^a

Name of operation	General T2 FS		IT2 FS	
	$\tilde{A} = \int_X [\int_{J_x^u} f_x(u)/u]/x$	$\tilde{B} = \int_X [\int_{J_x^w} g_x(w)/w]/x$	$\tilde{A} = \int_X [\int_{J_x^u} 1/u]/x$	$\tilde{B} = \int_X [\int_{J_x^w} 1/w]/x$
Union/join	$\mu_{\tilde{A} \cup \tilde{B}}(x) = \int_{u \in J_x^u} \int_{w \in J_x^w} f_x(u) \star g_x(w) / u \vee w$ $\equiv \mu_{\tilde{A}}(x) \sqcup \mu_{\tilde{B}}(x) \quad \forall x \in X$		$\mu_{\tilde{A} \cup \tilde{B}}(x) = [\underline{\mu}_{\tilde{A}}(x) \vee \underline{\mu}_{\tilde{B}}(x), \bar{\mu}_{\tilde{A}}(x) \vee \bar{\mu}_{\tilde{B}}(x)]$ $\forall x \in X$	
Intersection/ meet	$\mu_{\tilde{A} \cap \tilde{B}}(x) = \int_{u \in J_x^u} \int_{w \in J_x^w} f_x(u) \star g_x(w) / u \wedge w$ $\equiv \mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x) \quad \forall x \in X$		$\mu_{\tilde{A} \cap \tilde{B}}(x) = [\underline{\mu}_{\tilde{A}}(x) \wedge \underline{\mu}_{\tilde{B}}(x), \bar{\mu}_{\tilde{A}}(x) \wedge \bar{\mu}_{\tilde{B}}(x)]$ $\forall x \in X$	
	Minimum t -norm: see (A-1) in [6] Product t -norm: no simple algorithm exists	Minimum t -norm: see (A-3) in [6] (The second line of (A-3) should be $f(\theta)$). Product t -norm: no simple algorithm exists		
Complement/ negation	$\mu_{\tilde{A}}^-(x) = \int_{u \in J_x^u} f_x(u) / (1 - u) \equiv \neg \mu_{\tilde{A}}(x)$ $\forall x \in X$		$\mu_{\tilde{A}}^-(x) = [1 - \bar{\mu}_{\tilde{A}}(x), 1 - \underline{\mu}_{\tilde{A}}(x)] \quad \forall x \in X$	
Addition	$F + G = \int_{u \in U} \int_{w \in W} [f(u) \star g(w)] / (u + w)$		$F + G = [\underline{\mu}_F + \underline{\mu}_G, \bar{\mu}_F + \bar{\mu}_G]$	
Multiplication	$F \times G = \int_{u \in U} \int_{w \in W} [f(u) \star g(w)] / (u \times w)$		$F \times G = [\underline{\mu}_F \star \underline{\mu}_G, \bar{\mu}_F \star \bar{\mu}_G]$	

^a Note that t -norms (\star) are either minimum or product [6, 12, 13 (see the references therein)].

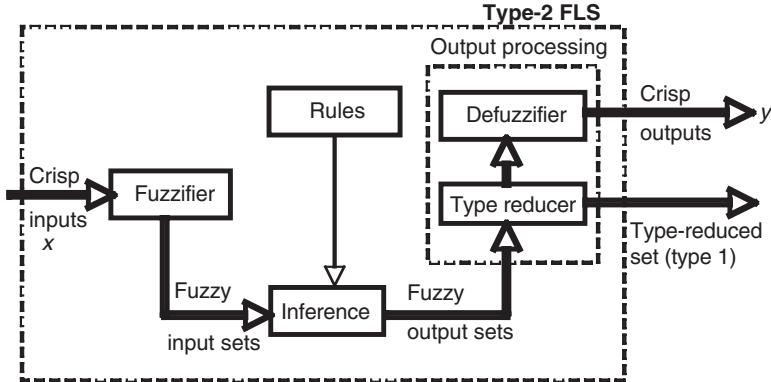


Figure 25.3 Type-2 FLS (Karnik et al. [16], © 1999, IEEE)

can be modeled in three ways, either as a fuzzy singleton (singleton fuzzification) or as a T1 FS (T1 non-singleton fuzzification), or even as an IT2 FS (T2 non-singleton fuzzification). The last one can be very useful if, e.g., the inputs are non-stationary (e.g., they are corrupted by noise whose signal-to-noise ratio is time varying). For an IT2 FLS, the firing set $[F_l(\mathbf{x}') \text{ or } F_l(\mathbf{x})]$ is called a *firing interval*. Observe that the left end (right end) of the firing interval uses only LMF (UMF) values. Blending of LMF and UMF values occurs during type reduction (described below). All of the results in Table 25.6 can be obtained using T1 FS mathematics [9] as a result of the RT.

Because practitioners of T1 FLSs are familiar with a graphical interpretation of the inference engine computation, T1 computations are contrasted with interval T2 computations in Figures 25.4 and 25.5. Figure 25.4, which is for a T1 FLS inference [2], depicts input and antecedent operations for a two-antecedent single-consequent rule, singleton fuzzification, and minimum *t*-norm. When $x_1 = x'_1$, $\mu_{F_1}(x'_1)$ occurs at the intersection of the vertical line at x'_1 with $\mu_{F_1}(x_1)$, and, when $x_2 = x'_2$, $\mu_{F_2}(x'_2)$ occurs at the intersection of the vertical line at x'_2 with $\mu_{F_2}(x_2)$. The *firing level* is a number equal to $\min[\mu_{F_1}(x'_1), \mu_{F_2}(x'_2)]$. The main thing to observe from this figure is that the result of input and antecedent operations is a number – the *firing level* $f(\mathbf{x}')$. This firing level is then *t*-normed with the entire consequent set, G . When $\mu_G(y)$ is a triangle and the *t*-norm is minimum, the resulting *fired-rule FS* is the trapezoid shown on the right side of the figure.

Table 25.5 General formulas for a T2 FLS [6, 12, 16]

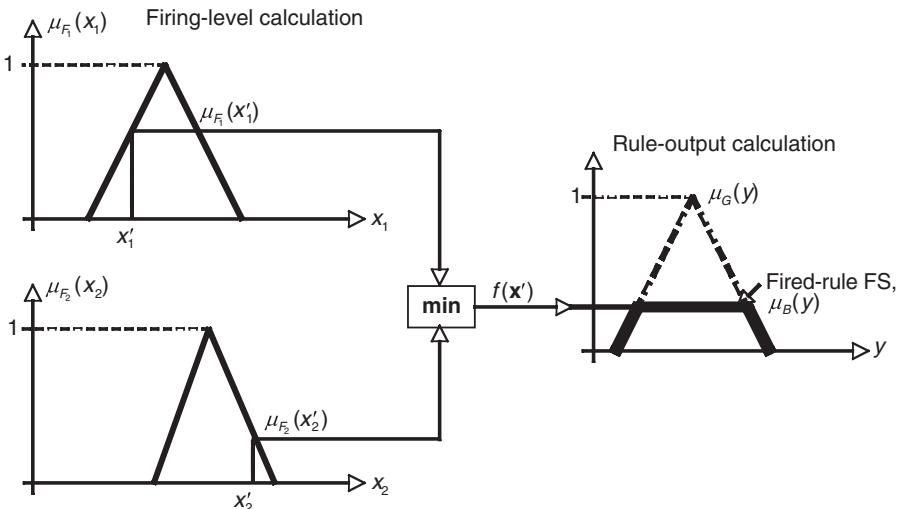
Name of item computed	Formula
MF of <i>l</i> th rule, R^l	$\mu_{R^l}(\mathbf{x}, y) = \left[\prod_{i=1}^p \mu_{\tilde{F}_i^l}(x_i) \right] \sqcap \mu_{\tilde{G}^l}(y)$
MF of inputs	$\mu_{\tilde{A}_{\mathbf{x}}}(\mathbf{x}) = \prod_{i=1}^p \mu_{\tilde{X}_i}(x_i)$
MF of <i>l</i> th fired rule	$\begin{aligned} \mu_{\tilde{B}^l}(y) = \mu_{\tilde{G}^l}(y) \sqcap & \left\{ \left[\sqcup_{x_1 \in X_1} \mu_{\tilde{X}_1}(x_1) \sqcap \mu_{\tilde{F}_1^l}(x_1) \right] \right. \\ & \left. \sqcap \cdots \sqcap \left[\sqcup_{x_p \in X_p} \mu_{\tilde{X}_p}(x_p) \sqcap \mu_{\tilde{F}_p^l}(x_p) \right] \right\}, \forall y \in Y \end{aligned}$
Special case: Singleton fuzzification	
Firing set	$\text{Firing set} = \prod_{i=1}^p \mu_{\tilde{F}_i^l}(x'_i)$
MF of <i>l</i> th fired rule	$\mu_{\tilde{B}^l}(y) = \mu_{\tilde{G}^l}(y) \sqcap [\text{Firing set}], \forall y \in Y$

Table 25.6 FLS formulas for Mamdani T1 FLS and IT2 FLS, $l = 1, \dots, M$ [6, 9, 17]^a

Name of the FLS	Firing level (for T1 FLS) or firing interval (for IT2 FLS) formula
Singleton T1 FLS	$f(\mathbf{x}') = \mu_{F_1^l}(x'_1) \star \dots \star \mu_{F_p^l}(x'_p)$
Non-singleton T1 FLS	$f(\mathbf{x}) = [\sup_{x_1 \in X_1} \mu_{X_1}(x_1)] \star \dots \star [\sup_{x_p \in X_p} \mu_{X_p}(x_p)]$
Singleton IT2 FLS	$F^l(\mathbf{x}') = [\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')] \equiv [\underline{f}^l, \bar{f}^l] = [\underline{\mu}_{\tilde{F}_1^l}(x'_1) \star \dots \star \underline{\mu}_{\tilde{F}_p^l}(x'_p), \bar{\mu}_{\tilde{F}_1^l}(x'_1) \star \dots \star \bar{\mu}_{\tilde{F}_p^l}(x'_p)]$
T1 non-singleton IT2 FLS	$F^l(\mathbf{x}) = [\underline{f}^l(\mathbf{x}), \bar{f}^l(\mathbf{x})] = [T_{i=1}^p (\sup_{x_i} \mu_{x_i}(x_i)) \star \underline{\mu}_{\tilde{F}_i^l}(x_i), T_{i=1}^p (\sup_{x_i} \mu_{x_i}(x_i)) \star \bar{\mu}_{\tilde{F}_i^l}(x_i)]$
T2 non-singleton IT2 FLS	$F^l(\mathbf{x}) = [\underline{f}^l(\mathbf{x}), \bar{f}^l(\mathbf{x})] = [T_{i=1}^p (\sup_{x_i} \mu_{x_i}(x_i)) \star \underline{\mu}_{\tilde{F}_i^l}(x_i), T_{i=1}^p (\sup_{x_i} \bar{\mu}_{x_i}(x_i)) \star \bar{\mu}_{\tilde{F}_i^l}(x_i)]$
Name of the FLS	Fired-rule output fuzzy set formula
Singleton T1 FLS	$\mu_B(y) = \mu_{B^l}(y) \star f(\mathbf{x}') \quad \forall y \in Y$
Non-singleton T1 FLS	$\mu_B(y) = \mu_{B^l}(y) \star f(\mathbf{x}) \quad \forall y \in Y$
Singleton IT2 FLS	$\mu_{\tilde{B}}(y) = [\underline{f}^l(\mathbf{x}') \star \underline{\mu}_{\tilde{G}^l}(y), \bar{f}^l(\mathbf{x}') \star \bar{\mu}_{\tilde{G}^l}(y)] \quad \forall y \in Y$
T1 non-singleton IT2 FLS	$\mu_{\tilde{B}}(y) = [\underline{f}^l(\mathbf{x}) \star \underline{\mu}_{\tilde{G}^l}(y), \bar{f}^l(\mathbf{x}) \star \bar{\mu}_{\tilde{G}^l}(y)] \quad \forall y \in Y$
T2 non-singleton IT2 FLS	$\mu_{\tilde{B}}(y) = [\underline{f}^l(\mathbf{x}) \star \underline{\mu}_{\tilde{G}^l}(y), \bar{f}^l(\mathbf{x}) \star \bar{\mu}_{\tilde{G}^l}(y)] \quad \forall y \in Y$

^a Note that t -norms (\star) are either minimum or product, and l denotes a rule number.

Figure 25.5 [2] shows the comparable calculations for an IT2 FLS. Now when $x_1 = x'_1$, the vertical line at x'_1 intersects $FOU(\tilde{F}_1)$ everywhere in the interval $[\underline{\mu}_{\tilde{F}_1}(x'_1), \bar{\mu}_{\tilde{F}_1}(x'_1)]$, and when $x_2 = x'_2$, the vertical line at x'_2 intersects $FOU(\tilde{F}_2)$ everywhere in the interval $[\underline{\mu}_{\tilde{F}_2}(x'_2), \bar{\mu}_{\tilde{F}_2}(x'_2)]$. Two firing levels are then computed, a lower firing level, $f(\mathbf{x}')$, and an upper firing level, $\bar{f}(\mathbf{x}')$, where $f(\mathbf{x}') = \min[\underline{\mu}_{\tilde{F}_1}(x'_1), \underline{\mu}_{\tilde{F}_2}(x'_2)]$ and $\bar{f}(\mathbf{x}') = \max[\bar{\mu}_{\tilde{F}_1}(x'_1), \bar{\mu}_{\tilde{F}_2}(x'_2)]$. The main thing to observe from this figure is that the result of input and antecedent operations is an interval – the *firing interval* $F(\mathbf{x}')$, where $F(\mathbf{x}') = [f(\mathbf{x}'), \bar{f}(\mathbf{x}')]$. $f(\mathbf{x}')$ is then t -normed with $LMF(\tilde{G})$ and $\bar{f}(\mathbf{x}')$ is t -normed with $UMF(\tilde{G})$. When

**Figure 25.4** T1 FLS inference: from firing level to rule output T1 FS (Mendel [2], © 2007, IEEE)

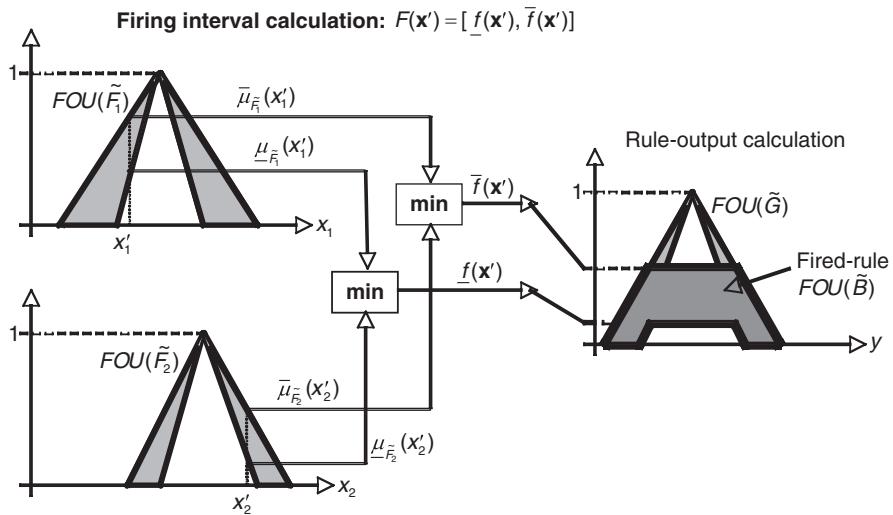


Figure 25.5 IT2 FLS inference: from firing interval to rule output FOU (Mendel [2], © 2007, IEEE)

\$FOU(\tilde{G})\$ is triangular, and the \$t\$-norm is minimum, the resulting *fired-rule FOU* is the trapezoidal FOU shown on the right side of the figure.

Comparing Figures 25.4 and 25.5, it is easy to see how the uncertainties about the antecedents flow through the T2 calculations. The more (less) the uncertainties are, the larger (smaller) the firing interval is and the larger (smaller) the fired-rule FOU is.

Formulas for first-order T1 and IT2 TSK FLSs are given in [6, chapter 13].

TR is a major new concept for a T2 FLS [6, 12, 18]. It projects a T2 FS into a T1 FS, but in such a way that if all sources of uncertainty disappear, then the T2 FLS reduces to a T1 FLS. To date, all TR methods extend T1 defuzzification methods that use some sort of centroid calculation to T2 FSs. Two of the most popular TR methods are summarized in Table 25.7. Each is based on the RT. Observe, from the top lines for both TR methods, that each method computes a collection of 2D points. When all of the points are properly ordered and connected, the result is a T1 FS; hence, TR for general T2 FSs leads to a T1 FS.

Centroid TR is performed after fired-rule T2 consequent sets have been combined by means of the join operation. Center-of-sets TR is performed using the individual firing sets and consequent sets. Unfortunately, because the number of embedded T2 FSs can be astronomical, performing any kind of TR for general T2 FSs is to date problematic. Methods for accelerating these calculations are under study.

TR for IT2 FLSs is easy to compute [6, 12, 18], because for all existing TR methods, the result is an interval set and such a set is completely determined by its two endpoints, \$c_l\$ and \$c_r\$. Although these endpoints cannot be computed in closed form, they can be exactly computed using two iterative algorithms, known as the *Karnik-Mendel* (KM) algorithms. These algorithms, which are summarized in Table 25.8, can be run in parallel and converge monotonically and superexponentially fast [19]. To date, they are the fastest algorithms for computing \$c_l\$ and \$c_r\$.

The entire chain of computations is summarized in Figure 25.6 [2]. Firing intervals are computed for all rules, and they depend explicitly on the input \$\mathbf{x}\$. For center-of-sets TR (see Table 25.7), off-line computations of the centroids are performed for each of the \$M\$ consequent IT2 FSs using KM algorithms and are then stored in memory. Center-of-sets TR combines the firing intervals and precomputed consequent centroids and uses the KM algorithms to perform the actual calculations.

Defuzzification, the last computation in the Figure 25.3 of T2 FLS, is performed by taking the average of \$y_l(\mathbf{x})\$ and \$y_r(\mathbf{x})\$.

Because the KM algorithms are iterative, TR using them can still represent a bottleneck for real-time applications of an IT2 FLS. Although closed-form formulas do not exist for the endpoints of the TR set,

Table 25.7 Two TR methods [6]

Centroid type reduction	
$Y_c(\mathbf{x}') = \left\{ \left(\xi_k, \left(T_{i=1}^N f_{y_i}(\theta_i(\mathbf{x}')) \right)_k \right) \right\}_{k=1}^{\prod_{i=1}^N M_i}$, where $\xi_k = \left(\sum_{i=1}^N y_i \theta_i(\mathbf{x}') / \sum_{i=1}^N \theta_i(\mathbf{x}') \right)_k$	
Step	Operations
1	Compute $\mu_{\tilde{B}}(y)$ as $\mu_{\tilde{B}}(y) = \sqcup_{l=1}^M \mu_{\tilde{B}^l}(y)$ ($\forall y \in Y$). This is possible because $\mu_{\tilde{B}^l}(y)$ ($l = 1, \dots, M$) will already have been computed for all $y \in Y$, as in Table 25.5
2	Discretize the y domain into N points y_1, \dots, y_N
3	Discretize each $J_{y_l}(\mathbf{x}')$ (the primary memberships of $\mu_{\tilde{B}}(y)$ at y_l) into a suitable number of points, say, M_l ($i = 1, \dots, N$). Let $\theta_i(\mathbf{x}') \in J_{y_l}(\mathbf{x}')$
4	Enumerate all the embedded T1 sets of \tilde{B} ; there will be $\prod_{i=1}^N M_i$ of them
5	Compute the centroid of each enumerated embedded T1 set and assign it a membership grade equal to the t -norm of the secondary grades corresponding to that enumerated embedded T1 set
Center of sets (COS) type reduction	
$Y_{\cos}(\mathbf{x}') = \left\{ \left(\xi_k, \left(T_{l=1}^M \mu_{C_{\tilde{G}^l}}(d_l) \star T_{l=1}^M \mu_{E_l(\mathbf{x}')}(e_l(\mathbf{x}')) \right)_k \right) \right\}_{k=1}^{\prod_{l=1}^M M_l N_l}$, where $\xi_k = \left(\sum_{l=1}^M d_l e_l(\mathbf{x}') / \sum_{l=1}^M e_l(\mathbf{x}') \right)_k$	
Step	Operations (M = number of rules)
1	Discretize the output space Y into a suitable number of points and compute the centroid $C_{\tilde{G}^l}$ of each consequent set on the discretized output space using Steps 3–5 of centroid TR. These consequent centroid sets can be computed ahead of time and stored for future use
2	Compute the T1 firing set $E_l(\mathbf{x}') = \sqcap_{i=1}^p \mu_{F_i^l}(x'_i)$ associated with the l th fired-rule consequent set
3	Discretize the domain of each T1 FS $C_{\tilde{G}^l}$ (Step 1) into a suitable number of points, say, N_l ($l = 1, \dots, M$)
4	Discretize the domain of each T1 FS $E_l(\mathbf{x}')$ (Step 2) into a suitable number of points, say, M_l ($l = 1, \dots, M$)
5	Enumerate all the possible combinations $(d_1, \dots, d_M, e_1(\mathbf{x}'), \dots, e_M(\mathbf{x}'))$, such that $d_l \in C_{\tilde{G}^l}$ and $e_l(\mathbf{x}') \in E_l(\mathbf{x}')$. The total number of combinations will be $\prod_{l=1}^M M_l N_l$
6	Compute the centroid $\sum_{l=1}^M d_l e_l(\mathbf{x}') / \sum_{l=1}^M e_l(\mathbf{x}')$ of each of the enumerated combinations and assign it a membership grade equal to the t -norm $T_{l=1}^M \mu_{C_{\tilde{G}^l}}(d_l) \star T_{l=1}^M \mu_{E_l(\mathbf{x}')}(e_l(\mathbf{x}'))$

these endpoints can be replaced by optimal (in a mini–max sense) lower and upper bounds (*uncertainty bounds*) that can be computed without having to perform TR¹ [20]. Because these uncertainty bounds can be used to bypass TR, they are summarized in Table 25.9.

The following helps justify bypassing TR during the operational stage of an IT2 FLS [20]: *For a group of input–output data $\{\mathbf{x}_i, y_i\}_{i=1}^N$ and an IT2 FLS, let the risk function, R_{TR} , associated with the TR set $[y_l(\mathbf{x}), y_r(\mathbf{x})]$, be given as*

$$R_{TR} = \frac{1}{N} \sum_{i=1}^N \left[y_i - \frac{y_l(\mathbf{x}_i) + y_r(\mathbf{x}_i)}{2} \right]^2. \quad (1)$$

¹ In [20] there are detailed derivations of the uncertainty bounds for center-of-sets TR (because it handles non-symmetrical shoulder MFs better than do other kinds of TR); however, these results are also applicable to other kinds of TR, as explained in [20, table V].

Table 25.8 KM algorithms for computing the centroid endpoints of an IT2 FS, \tilde{A} , and their properties [6, 12, 18]^a

Step	KM algorithm for c_l	KM algorithm for c_r
Step	$c_l = \min_{\forall \theta_i \in [\underline{\mu}_{\tilde{A}}(x_i), \bar{\mu}_{\tilde{A}}(x_i)]} \left(\sum_{i=1}^N x_i \theta_i / \sum_{i=1}^N \theta_i \right)$	$c_r = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{A}}(x_i), \bar{\mu}_{\tilde{A}}(x_i)]} \left(\sum_{i=1}^N x_i \theta_i / \sum_{i=1}^N \theta_i \right)$
1	Initialize θ_i by setting $\theta_i = [\underline{\mu}_{\tilde{A}}(x_i) + \bar{\mu}_{\tilde{A}}(x_i)]/2$, $i = 1, \dots, N$ (or $\theta_i = \underline{\mu}_{\tilde{A}}(x_i)$, $i \leq \lfloor (n+1)/2 \rfloor$, and $\theta_i = \bar{\mu}_{\tilde{A}}(x_i)$, $i > \lfloor (n+1)/2 \rfloor$, where $\lfloor \bullet \rfloor$ denotes the first integer equal to or smaller than \bullet), and then compute	$c' = c(\theta_1, \dots, \theta_N) = \sum_{i=1}^N x_i \theta_i / \sum_{i=1}^N \theta_i$
2	Find k ($1 \leq k \leq N-1$), such that $x_k \leq c' \leq x_{k+1}$	
3	Set $\theta_i = \bar{\mu}_{\tilde{A}}(x_i)$, when $i \leq k$, and $\theta_i = \underline{\mu}_{\tilde{A}}(x_i)$, when $i \geq k+1$, and then compute	Set $\theta_i = \underline{\mu}_{\tilde{A}}(x_i)$, when $i \leq k$, and $\theta_i = \bar{\mu}_{\tilde{A}}(x_i)$, when $i \geq k+1$, and then compute
4	Check if $c_l(k) = c'$. If yes, stop and set $c_l(k) = c_l$ and call kk_L . If no, go to Step 5	Check if $c_r(k) = c'$. If yes, stop and set $c_r(k) = c_r$ and call kk_R . If no, go to Step 5
5	Set $c' = c_l(k)$ and go to Step 2	Set $c' = c_r(k)$ and go to Step 2

Properties of the KM algorithms [19]

Convergence is monotonic and superexponentially fast

^a Note that $x_1 \leq x_2 \leq \dots \leq x_N$.

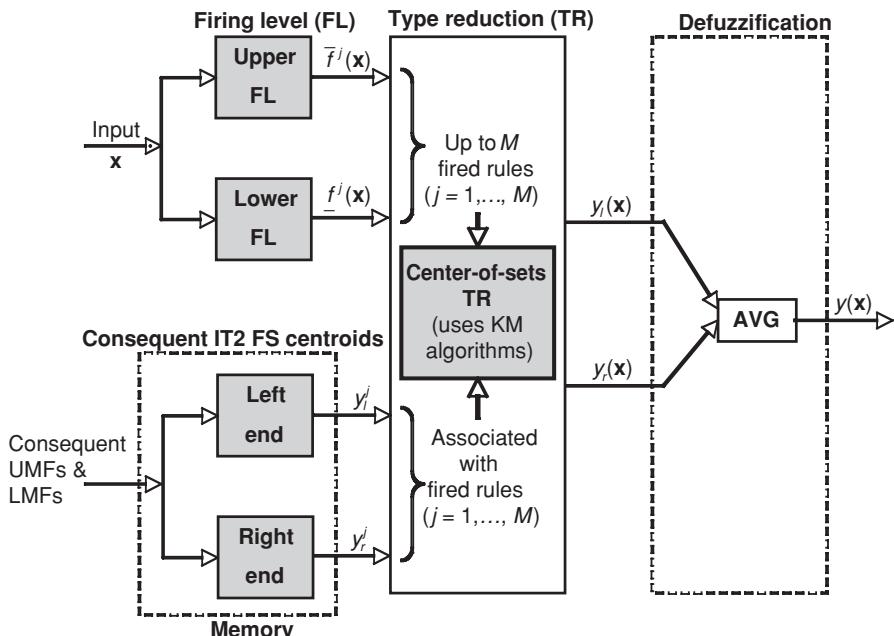


Figure 25.6 Computations in an IT2 FLS that use center-of-sets TR (Mendel [2], © 2007, IEEE)

Table 25.9 Minimax uncertainty bounds and related results [20]

Boundary type 1 FLSs	
$y_l^{(0)}(\mathbf{x}) = \sum_{i=1}^M \underline{f}^i y_l^i / \sum_{i=1}^M f^i$	$y_r^{(0)}(\mathbf{x}) = \sum_{i=1}^M \bar{f}^i y_r^i / \sum_{i=1}^M \bar{f}^i$
$y_l^{(M)}(\mathbf{x}) = \sum_{i=1}^M \bar{f}^i y_l^i / \sum_{i=1}^M \bar{f}^i$	$y_r^{(M)}(\mathbf{x}) = \sum_{i=1}^M \underline{f}^i y_r^i / \sum_{i=1}^M f^i$
y_l^i , and y_r^i ($i = 1, \dots, M$) are the endpoints of the centroids of the consequent IT2 FSs. They are computed using the KM algorithms (Table 25.8)	
Uncertainty bounds	
$\underline{y}_l(\mathbf{x}) \leq y_l(\mathbf{x}) \leq \bar{y}_l(\mathbf{x})$	$\underline{y}_r(\mathbf{x}) \leq y_r(\mathbf{x}) \leq \bar{y}_r(\mathbf{x})$
$\bar{y}_l(\mathbf{x}) = \min \left\{ y_l^{(0)}(\mathbf{x}), y_l^{(M)}(\mathbf{x}) \right\}$	$\underline{y}_r(\mathbf{x}) = \max \left\{ y_r^{(0)}(\mathbf{x}), y_r^{(M)}(\mathbf{x}) \right\}$
$\bar{y}_l(\mathbf{x}) = \bar{y}_l(\mathbf{x}) - \left[\frac{\sum_{i=1}^M (\bar{f}^i - \underline{f}^i)}{\sum_{i=1}^M \bar{f}^i \sum_{i=1}^M \underline{f}^i} \right] \times \frac{\sum_{i=1}^M f^i (y_l^i - y_l^1) \sum_{i=1}^M \bar{f}^i (y_l^M - y_l^i)}{\sum_{i=1}^M \bar{f}^i (y_l^i - y_l^1) + \sum_{i=1}^M \bar{f}^i (y_l^M - y_l^i)}$	$\underline{y}_r(\mathbf{x}) = \underline{y}_r(\mathbf{x}) + \left[\frac{\sum_{i=1}^M (\bar{f}^i - \underline{f}^i)}{\sum_{i=1}^M \bar{f}^i \sum_{i=1}^M \underline{f}^i} \right] \times \frac{\sum_{i=1}^M \bar{f}^i (y_r^i - y_r^1) \sum_{i=1}^M f^i (y_r^M - y_r^i)}{\sum_{i=1}^M \bar{f}^i (y_r^i - y_r^1) + \sum_{i=1}^M \underline{f}^i (y_r^M - y_r^i)}$
Approximate TR set $[\hat{y}_l(\mathbf{x}), \hat{y}_r(\mathbf{x})] \approx [(\underline{y}_l(\mathbf{x}) + \bar{y}_l(\mathbf{x}))/2, (\underline{y}_r(\mathbf{x}) + \bar{y}_r(\mathbf{x}))/2]$	
Approximate defuzzified output $\hat{y}(\mathbf{x}) = \frac{1}{2}[\hat{y}_l(\mathbf{x}) + \hat{y}_r(\mathbf{x})]$	
Error bound between defuzzified and approximate defuzzified outputs	
$\delta(\mathbf{x}) \equiv \left \frac{\underline{y}_l(\mathbf{x}) + y_r(\mathbf{x})}{2} - \frac{1}{2} \left[\frac{\underline{y}_l(\mathbf{x}) + \bar{y}_l(\mathbf{x})}{2} + \frac{\underline{y}_r(\mathbf{x}) + \bar{y}_r(\mathbf{x})}{2} \right] \right $	
$\delta(\mathbf{x}) \leq \frac{1}{4}[(\bar{y}_l(\mathbf{x}) - \underline{y}_l(\mathbf{x})) + (\bar{y}_r(\mathbf{x}) - \underline{y}_r(\mathbf{x}))]$	

and the risk function, R_{APP} , associated with its approximation set $[(\underline{y}_l(\mathbf{x}) + \bar{y}_l(\mathbf{x}))/2, (\underline{y}_r(\mathbf{x}) + \bar{y}_r(\mathbf{x}))/2]$, be given as

$$R_{APP} = \frac{1}{N} \sum_{i=1}^N \left\{ y_i - \frac{1}{2} \left[\frac{\underline{y}_l(\mathbf{x}_i) + \bar{y}_l(\mathbf{x}_i)}{2} + \frac{\underline{y}_r(\mathbf{x}_i) + \bar{y}_r(\mathbf{x}_i)}{2} \right] \right\}^2. \quad (2)$$

Then

$$\left| \sqrt{R_{TR}} - \sqrt{R_{APP}} \right| \leq \frac{1}{4} \sqrt{\frac{1}{N} \sum_{i=1}^N [(\bar{y}_l(\mathbf{x}_i) - \underline{y}_l(\mathbf{x}_i)) + (\bar{y}_r(\mathbf{x}_i) - \underline{y}_r(\mathbf{x}_i))]^2} \quad (3)$$

To date, three approaches for eliminating TR have been proposed:

1. Wu and Mendel [20] use TR during the design of the IT2 FLS but then run the IT2 FLS using the formulas given in Table 25.9. During the design, instead of using only R_{TR} , they use a weighted average between R_{TR} and N values of $\delta^2(\mathbf{x}_i)$, $i = 1, \dots, N$. (δ is defined in Table 25.9.) In this way, they trade-off some RMSE with not having to perform TR. The drawback to this approach is that TR is still performed during the design step.
2. Lynch *et al.* [21, 22] abandon TR completely. They replace all of the IT2 FLS computations with those in Table 25.9; i.e., the Table 25.9 equations are their IT2 FLS. Their design of this IT2 FLS is then based on minimizing R_{APP} . They found that the errors incurred by doing this are very small. This is a very clever approach.

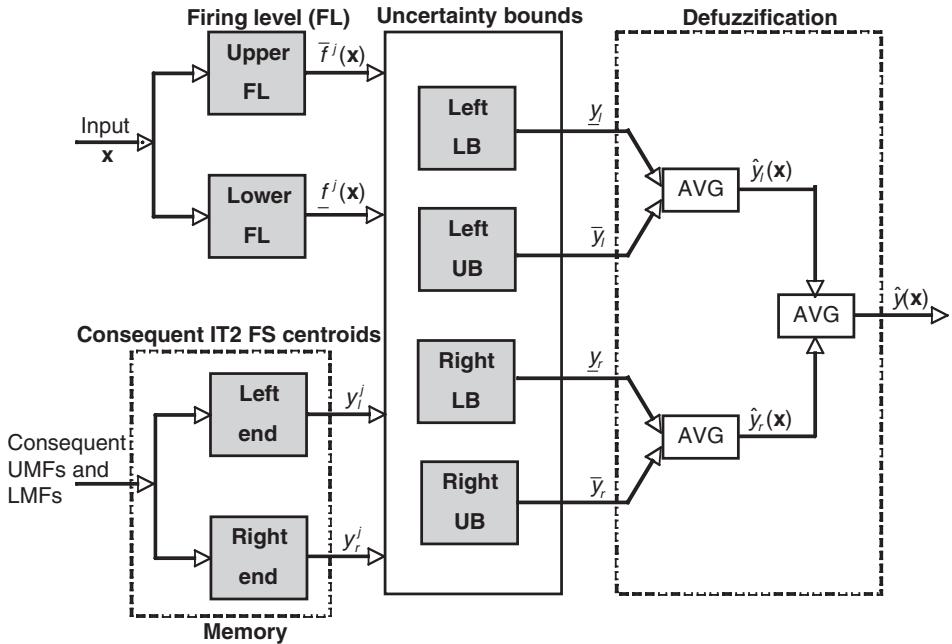


Figure 25.7 Computations in an IT2 FLS that use uncertainty bounds instead of TR. LB, lower bound; UB, upper bound (Mendel, [2], © 2007, IEEE)

3. Melgarejo *et al.* [23, 24] and Melgarejo and Penha-Reyes [25] have designed a VLSI IT2 FLS chip. It is also based on the Table 25.9 equations and the Wu–Mendel design and implementation approach; however, it could also be based on the Lynch *et al.* approach. Because the Table 25.9 equations are hard wired, the design stage can be done very quickly.

Figure 25.7 summarizes the computations in an IT2 FLS that uses uncertainty bounds [2]. The front end of the calculations is identical to the front end using TR (see Figure 25.6). After the uncertainty bounds are computed, the actual values of $y_l(x')$ and $y_r(x')$ (which would have been computed using TR, as in Figure 25.6) are approximated by averaging their respective uncertainty bounds, the results being $\hat{y}_l(x')$ and $\hat{y}_r(x')$. Defuzzification is then achieved by averaging these two approximations, the result being $\hat{y}(x)$, which is an approximation to $y(x)$. Remarkably, very little accuracy is lost when the uncertainty bounds are used. This is proved in [20] and has been demonstrated in [22]. See [26] for additional discussions on IT2 FLSs without TR.

In summary, IT2 FLSs are pretty simple to understand and they can be implemented in two ways, one that uses TR and one that uses uncertainty bounds.

Recent works on general T2 FLSs focus on using triangle secondary MFs [15]. This seems to be the next most logical step to pursue in the hierarchy of T2 FLSs.

25.6 Centroid of an IT2 FS

Since the introduction of the centroid, its properties have been studied by Mendel and Wu [27], Liu and Mendel [28], and Mendel [29]. Three of the most important properties are (1) the centroid is shift invariant; i.e., the centroid of an FOU that is shifted along the axis of the primary variable equals the centroid of the original FOU plus the shift; (2) if the FOU is completely filled in, i.e., $LMF(\tilde{A}) = 0$, then

Table 25.10 Centroid bounds for a symmetric FOU [30]

Symbol	Definition
A_{LMF}	Area under the lower MF
A_{UMF}	Area under the upper MF
A_{FOU}	Area of the FOU, i.e., $A_{FOU} = A_{UMF} - A_{LMF} = 2 \int_m^\infty [\bar{\mu}(x) - \underline{\mu}(x)]dx$
$c_{HFOU}(\tilde{A})$	Centroid of half of $FOU(\tilde{A})$, i.e., $c_{HFOU}(\tilde{A}) = \int_m^\infty x[\bar{\mu}(x) - \underline{\mu}(x)]dx/(A_{FOU}/2)$
Bounds ($x \in [x_1, x_N]$)	
$\underline{c}_r(\tilde{A}) \leq c_r(\tilde{A}) \leq \min(\bar{c}_r(\tilde{A}), x_N)$	
$\underline{c}_r(\tilde{A}) = m + [c_{HFOU}(\tilde{A}) - m] \frac{A_{FOU}}{A_{UMF} + A_{LMF}}$	$\bar{c}_r(\tilde{A}) = m + [c_{HFOU}(\tilde{A}) - m] \frac{A_{FOU}}{2A_{LMF}}$
$\max(x_1, \underline{c}_l(\tilde{A})) \leq c_l(\tilde{A}) \leq \bar{c}_l(\tilde{A})$	
$\underline{c}_l(\tilde{A}) = 2m - \bar{c}_r(\tilde{A})$	$\bar{c}_l(\tilde{A}) = 2m - \underline{c}_r(\tilde{A})$

$UMF(\tilde{A})$ plays no role in determining the centroid, in which case the centroid does not depend on the shape of $FOU(\tilde{A})$; and, (3) if $FOU(\tilde{A})$ is symmetrical about $m \in X$, then the centroid is also symmetrical about $x = m$, and for such a FOU it is only necessary to compute either c_l or c_r , after which, e.g., $c_r = 2m - c_l$, thereby resulting in a 50% savings in computation time.

Wu and Mendel [20] showed that the centroid provides a measure of the uncertainty for an IT2 FS. Intuitively, one anticipates that geometric properties about the FOU, such as its area and the center of gravities (centroids) of its upper and lower MFs, will be associated with the amount of uncertainty in such a T2 FS. Recently, Mendel and Wu [30–32] demonstrated that this intuition is correct. They quantified uncertainty bounds for the centroid of both symmetric and non-symmetric IT2 FSs with respect to such geometric properties. Using these results, it is possible to formulate and solve *forward problems*, i.e., to go from parametric IT2 FS models to data with associated uncertainty bounds. Results for a symmetrical FOU are summarized in Table 25.10. Comparable results for an unsymmetrical FOU are in [32].

These bounds have been used to solve inverse problems [1], i.e., going from interval data about a word to an FOU for that word, something that is needed for computing with words (Section 25.8).

25.7 The Fuzzy Weighted Average

The *fuzzy weighted average* (FWA) is a weighted average involving T1 FSs. It has been studied in multiple-criteria decision making [28, 33–38] and is useful as an aggregation (fusion) method in situations where [see (4)] attributes (x_i) and expert weights (w_i) are modeled as T1 FSs, or where the attributes are modeled as either crisp numbers or interval sets, and the expert weights are still modeled as T1 FSs or interval sets.

Consider the following *weighted average*:

$$y = \sum_{i=1}^n w_i x_i \left/ \sum_{i=1}^n w_i \right. = f(w_1, \dots, w_n, x_1, \dots, x_n). \quad (4)$$

In (4), w_i are *weights* that act on *attributes* (indicators, features, decisions, judgments, etc.), x_i . In the FWA, $\forall x_i$ are T1 fuzzy numbers, i.e., each x_i is described by the MF of a T1 FS, $\mu_{X_i}(x_i)$, which must be prespecified, and $\forall w_i$ are also T1 fuzzy numbers, i.e., each w_i is described by the MF of a T1 FS, $\mu_{W_i}(w_i)$, which must also be prespecified. In (4), y is a T1 FS, with MF $\mu_Y(y)$, but there is no known closed-form formula for computing $\mu_Y(y)$. Instead, α -cuts, an α -cut decomposition theorem [39] of a

Table 25.11 Computing the FWA [28]

Step	Computation ($i = 1, \dots, n$)	Comments ($i = 1, \dots, n$)
1	Discretize the complete range of the membership [0, 1] of the fuzzy numbers into m α -cuts, $\alpha_1, \dots, \alpha_m$	The degree of accuracy depends on m
2	For each α_j , find the corresponding intervals for X_i in x_i and W_i in w_i	Denote the endpoints of the intervals of x_i and w_i by $[a_i(\alpha_j), b_i(\alpha_j)]$ and $[c_i(\alpha_j), d_i(\alpha_j)]$, respectively
	Compute $y(\alpha_j) = [\min_k y_k(\alpha_j), \max_k y_k(\alpha_j)]$, $j = 1, \dots, m$	
3	Compute $\min_k y_k(\alpha_j) = f_L^*(\alpha_j)$	Use the KM algorithm to find k_L in f_L^* , where
	$f_L^* = \min_{\forall w_i \in [c_i, d_i]} \frac{\sum_{i=1}^n a_i w_i}{\sum_{i=1}^n w_i} = \frac{\sum_{i=1}^{k_L} a_i d_i + \sum_{i=k_L+1}^n a_i c_i}{\sum_{i=1}^{k_L} d_i + \sum_{i=k_L+1}^n c_i}$	
4	Compute $\max_k y_k(\alpha_j) = f_U^*(\alpha_j)$	Use the KM algorithm to find k_U in f_U^* , where
	$f_U^* = \max_{\forall w_i \in [c_i, d_i]} \frac{\sum_{i=1}^n b_i w_i}{\sum_{i=1}^n w_i} = \frac{\sum_{i=1}^{k_U} b_i c_i + \sum_{i=k_U+1}^n b_i d_i}{\sum_{i=1}^{k_U} c_i + \sum_{i=k_U+1}^n d_i}$	
5	Compute $\mu_Y(y)$ using $y(\alpha_1), \dots, y(\alpha_m)$ and an α -cut decomposition theorem [39],	Let $I_{\alpha_j}(y) = \begin{cases} 1 & \forall y \in \left[\min_k y_k(\alpha_j), \max_k y_k(\alpha_j) \right] \\ 0 & \forall y \notin \left[\min_k y_k(\alpha_j), \max_k y_k(\alpha_j) \right] \end{cases}$ so that $\mu_Y(y) = \sup_{\forall \alpha_j (j=1, \dots, m)} \alpha_j I_{\alpha_j}(y)$

T1 FS, and a variety of algorithms, including the KM algorithms, can be used to compute $\mu_Y(y)$ (e.g., [28, 36–38]). This procedure is summarized in Table 25.11. To date, using the KM algorithms [28] is the fastest way to compute the FWA. Its use to compute the FWA represents the first ‘spin-off’ of something that was developed within the context of T2 FSs to something outside of T2 FSs.

Recently, the FWA has been extended to the *linguistic weighted average* (LWA) [40], in which all quantities in (4) can be modeled as IT2 FSs. The LWA can be used in decision-making situations where words are used instead of numbers or intervals, and it represents one kind of computing with words engine (Section 25.8).

25.8 Computing with Words

In 1996 Zadeh [41] published his first seminal work on *computing with words* (CWW). Since that time entire books and other articles have been devoted to this important subject. A small sampling of these are [1, 42–49]. Mendel [1, 46] has proposed IT2 FSs as models for words, because words mean different things to different people, and so a fuzzy set model for words is needed that can capture a measure of their uncertainty. Mendel has also argued that an IT2 FS models *first-order uncertainties* about a word, and such uncertainty information can be collected from a group of subjects using simple surveys.

Figure 25.8 diagrams our interpretation of CWW. Words are *encoded*, i.e., mapped into IT2 FS models. There can be a multitude of *CWW engines*, including IF–THEN rules, LWA, fuzzy dynamic programming, fuzzy Choquet integral, linguistic summarizations, etc. The CWW engine maps IT2 FSs into an output IT2 FS. Finally, the output IT2 FS is *decoded*, i.e., mapped into a word. Decoding is accomplished by determining which word, in a preestablished codebook, the output IT2 FS is most similar to.

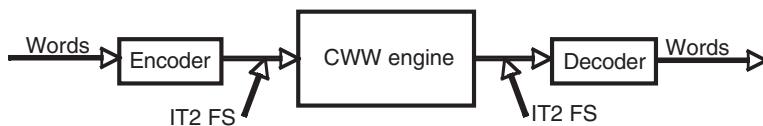


Figure 25.8 CWW paradigm

In order to establish IT2 FS models for words, data should be collected about words from a group of subjects. Doing this captures both the intrauncertainty each individual has about a word and the interuncertainty a group of individuals has about that word. Two very different approaches for doing this have been described. Both approaches map data collected from subjects into a parsimonious parametric model of an FOU and illustrate the combining of fuzzy sets and statistics – *Type 2 fuzzistics*.²

In one approach, called the *person MF approach* [1],

1. Person MF data (a person MF is an FOU that a person provides on a prescribed scale for a primary variable) is collected that reflects both the intra- and interlevels of uncertainties about a word, from a group of people.
2. An IT2 FS model is defined for a word as a specific aggregation of all such person MFs.
3. This aggregation is modeled mathematically.

Person MFs can only be collected from people who are already very knowledgeable about an FS, and therefore other techniques must be established for collecting word data from the vast majority.

In the second approach, called the *interval endpoints approach* [1],

1. Interval endpoint data are collected about a word from a group of subjects.
2. Endpoint statistics are established for the data.
3. Those statistics are mapped into a prespecified parametric FOU.

This method is analogous, in statistical modeling, to first choosing the underlying probability distribution (i.e., data-generating model) and then fitting the parameters of that model using data and a meaningful design method, e.g., the method of maximum likelihood. This method is based on the centroid bounds that were described in Section 25.6. The details of how to perform Step 3 are in [31] for symmetric FOUs and are under study for non-symmetrical FOUs.

It seems that IT2 FSs will play a very important role in implementing Zadeh's paradigm of CWW; however, much more remains to be done to turn CWW into an operational reality.

25.9 Applications and Hardware

One sign of a vibrant field is its applications. Applications that have appeared in the literature for T2 FSs and systems since 2001 are summarized in Table 25.12. For applications prior to that year, see [6, pp. 13–14]. Applications are broadening, and control applications, which were the original bread-and-butter ones for T1 FLSs, are now a major focus of attention for IT2 FLSs, and even general T2 FLSs.

If the reader is bothered by the question ‘Will an IT2 FLS significantly outperform a T1 FLS?’, then see [67] in which three important applications demonstrate that the answer is a definite ‘Yes’.

Recently, IT2 FLSs have also been implemented in hardware [23–25], which should make them more attractive and accessible for industrial applications.

² This term was first coined in [46].

Table 25.12 Applications of type 2 fuzzy set^a

Category	Reference number and subject
<i>Approximation</i>	[12] (function approximation), [50] (TSK/steel strip temperature)
<i>Clustering</i>	[51] (C spherical shells algorithm), [39, 51–54] (Fuzzy c -means)
<i>Communications</i>	[16, 55–57] (equalization of NL fading channels), [58] (Co-channel interference elimination), [59] (wireless sensors/power on–off control), [60] (wireless sensor network lifetime analysis)
<i>Control</i>	[61] (intelligent control of NL dynamic plants), [62] (FLC using fuzzy Lyapunov function), [63–68] (control of robots), [69] (decoupled adaptive control of NL TORA system), [108] (connection admission control using surveys), [70] (control of buck DC–DC converters), [21, 22, 71] (speed control of marine and traction diesel engines), [67, 72] (adaptive model-based control of NL plants), [73] (integrated development platform for intelligent control), [74] (proportional controller), [75] (controller for liquid-level process)
<i>Databases</i>	[76] (extended fuzzy relational databases), [77, 78] (linguistic summaries of databases)
<i>Decision making</i>	[79, 80] (using T2 FSs in decision making), [81] (modeling variation in human decision making)
<i>Embedded agents</i>	[67, 82–85] (ubiquitous computing environments)
<i>Health care</i>	[86] (medical differential diagnosis), [87] (modeling uncertainty in clinical diagnosis), [88] (nursing assessment)
<i>Hidden Markov models</i>	[89] (phoneme recognition)
<i>Knowledge mining</i>	[12, 90] (extracting knowledge from questionnaire surveys)
<i>Neural networks</i>	[91] (fuzzy perceptron), [92] (linguistic perceptron)
<i>Pattern classification</i>	[93] (support vector machine fusion model), [94] (classification of coded video streams), [95] (classification of battlefield vehicles) [96] (fuzzy k -nearest neighbor), [97] (uncertainty in pattern classification)
<i>Quality control</i>	[98] (sound speakers)
<i>Scheduling</i>	[99] (transport scheduling)
<i>Signal processing</i>	[100] (adaptive noise cancelation), [101] (preprocessing radiographic images), [17, 57, 102] (forecasting of time series), [103, 104] (modeling and prediction of exchange rates), [105] (estimating compressor discharge pressure), [106] (geographical information)
<i>Spatial query</i>	[107] (spatial objects)

^aNL, non-linear.

25.10 Miscellaneous Issues

An important question (raised by one of the reviewers of this chapter) is ‘Where does one obtain the T2 MFs?’ In an IT2 FLS, training data can be used to optimize the MF parameters for prespecified FOU shapes, just as they can be used to optimize the MFs in a T1 FLS. See, e.g., [6] for discussions on how to do this using a variety of design techniques, including backpropagation, least squares, singular-value decomposition, and iterative methods. For CWW, data should be collected from a group of subjects and then mapped into the FOU of an IT2 FS, as briefly explained in Section 25.8.

An IT2 FLS is scalable in the sense that the results that are summarized in Tables 25.6–25.9 apply regardless of the number of antecedents in a rule or the number of rules. The complexity of an IT2 FLS is about twice that of a T1 FLS; however, if a T1 FLS cannot provide satisfactory performance due to uncertainties, then an IT2 FLS is the next logical choice. It is usually wise to baseline an IT2 FLS against the best-performing T1 FLS so that one can justify having to use an IT2 FLS. Exactly how much

uncertainty is needed before an IT2 FLS outperforms a T1 FLS is at present an open issue and is worthy of study.

For CWW, since words mean different things to different people, this author believes that one should immediately use an IT2 FS model, for to do otherwise is scientifically incorrect [46]. When unpredictable uncertainties are present, probability is used. Usually, people do not ask ‘How much unpredictability must be present before probability should be used?’ Probability is used regardless of the amount of unpredictable uncertainties. And so, I would argue that interval T2 FSs should be used as word models regardless of how much linguistic uncertainty is present.

25.11 Conclusion

T2 FSs are now very well established and are gaining in more and more popularity. The following questions and answers which were published in [7] are still pertinent:

1. *Why did it take so long for the concept of a T2 FS to emerge?* It seems that science moves in progressive ways where one theory is eventually replaced or supplemented by another and then another. In school we learn about determinism before randomness. Learning about T1 FSs before T2 FSs fits a similar learning model. So, from this point of view it was very natural for fuzzyites to develop T1 FSs as far as possible. Only by doing so was it really possible later to see the shortcomings of such FSs when one tries to use them to model words or to apply them to situations where uncertainties abound.
2. *Why did T2 FSs not immediately become popular?* Although Zadeh introduced T2 FSs in 1975, very little was published about them until the mid-to-late nineties. Until then they were studied by only a relatively small number of people. Recall that in the 1970s people were first learning what to do with T1 FSs, e.g., fuzzy logic control. Bypassing those experiences would have been unnatural. Once it was clear what could be done with T1 FSs, it was only natural for people to then look at more challenging problems. This is where we are today.
3. *Why do we believe that by using T2 FSs we will outperform the use of T1 FSs?* T2 FSs are described by MFs that are characterized by more parameters than are MFs for T1 FSs. Hence, T2 FSs provide us with more design degrees of freedom; so using T2 FSs has the *potential* to outperform a T1 FSs, especially when we are in uncertain environments. At present, *there is no theory that guarantees that a T2 FS will always do this*; but many applications have demonstrated that this is indeed the case.

The T2 field is growing rapidly [4] with applications of T2 methods appearing in diverse fields (e.g., geographical information [106]). IT2 FSs and FLSs are very easy to learn and use. People who already use T1 FSs and FLSs will have no difficulty in learning about and using them, thanks in no small way to the fact that all results for them can be explained using T1 FS mathematics [9]. The case for general T2 FSs and FLSs is still undecided, but it is now under study in many locations and is no doubt where much future research will be directed.

For readers who want to learn more about IT2 FSs and FLSs, the easiest way to do this is to read [9]. For readers who want a very complete treatment about general and interval T2 FSs and FLSs, see [6]. Finally, for readers who may already be familiar with T2 FSs and want to know what has happened since the 2001 publication of [6], see [26]. Also, see <http://www.type2fuzzylogic.org> for a vast amount of happenings about T2 FSs and systems.

References

- [1] J.M. Mendel. Computing with words and its relationships with fuzzistics. *Inf. Sci.* 177 (2007) 988–1006.
- [2] J.M. Mendel. Type-2 fuzzy sets and systems: An overview. *IEEE Comput. Intell. Mag.* 2 (2007) 20–29.
- [3] R.I. John and S. Coupland. Extensions to type-1 fuzzy logic: Type-2 fuzzy logic and uncertainty. In: G.Y. Yen and D.B. Fogel (eds), *Computational Intelligence: Principles and Practice*, IEEE Computational Intelligence Society, Piscataway, NJ, 2006, pp. 89–101.
- [4] R.I. John and S. Coupland. Type-2 fuzzy logic: A historical view. *IEEE Comput. Intell. Mag.* 2 (2007) 57–62.

- [5] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning—I. *Inf. Sci.* 8 (1975) 199–249.
- [6] J.M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall, Upper-Saddle River, NJ, 2001.
- [7] J.M. Mendel. Type-2 fuzzy sets: Some questions and answers. *IEEE Connect. Newslett. IEEE Neural Netw. Soc.* 1 (2003) 10–13.
- [8] J.M. Mendel and R.I. John. Type-2 fuzzy sets made simple. *IEEE Trans. Fuzzy Syst.* 10 (2002) 117–127.
- [9] J.M. Mendel, R.I. John, and F. Liu. Interval type-2 fuzzy logic systems made simple. *IEEE Trans. Fuzzy Syst.* 14 (2006) 808–821.
- [10] M. Mizumoto and K. Tanaka. Some properties of fuzzy sets of type-2. *Inf. Control* 31 (1976) 312–340.
- [11] M. Mizumoto and K. Tanaka. Fuzzy sets of type-2 under algebraic product and algebraic sum. *Fuzzy Sets Syst.* 5 (1981) 277–290.
- [12] N.N. Karnik and J.M. Mendel. *An Introduction to Type-2 Fuzzy Logic Systems*. University of Southern California, Los Angeles, CA, 1998.
- [13] N.N. Karnik and J.M. Mendel. Operations on type-2 fuzzy sets. *Fuzzy Sets Syst.* 122 (2001) 327–348.
- [14] S. Coupland and R.I. John. Geometric type-1 and type-2 fuzzy logic systems. *IEEE Trans. Fuzzy Syst.* 15 (2007) 3–15.
- [15] J.T. Starczewski. Extended triangular norms on Gaussian fuzzy sets. In: *Proceedings of EUSFLAT-LFA*, Barcelona, Spain, 2005, pp. 872–877.
- [16] N.N. Karnik, J.M. Mendel, and Q. Liang. Type-2 fuzzy logic systems. *IEEE Trans. Fuzzy Syst.* 7 (1999) 643–658.
- [17] Q. Liang and J.M. Mendel. Interval type-2 fuzzy logic systems: Theory and design. *IEEE Trans. Fuzzy Syst.* 8 (2000) 535–550.
- [18] N.N. Karnik and J.M. Mendel. Centroid of a type-2 fuzzy set. *Inf. Sci.* 132 (2001) 195–220.
- [19] J.M. Mendel and F. Liu. Super-exponential convergence of the Karnik-Mendel algorithms for computing the centroid of an interval type-2 fuzzy set. *IEEE Trans. Fuzzy Syst.* 15 (2007) 309–320.
- [20] H. Wu and J.M. Mendel. Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. *IEEE Trans. Fuzzy Syst.* 10 (2002) 622–639.
- [21] C. Lynch, H. Hagras, V. Callaghan. Embedded interval type-2 neuro-fuzzy speed controller for marine diesel engines. In: *Proceedings of IPMU*, Paris, France, 2006, pp. 1340–1347.
- [22] C. Lynch, H. Hagras, and V. Callaghan. Using uncertainty bounds in the design of an embedded real-time type-2 neuro-fuzzy speed controller for marine diesel engines. In: *Proceedings of IEEE-FUZZ Conference*, Vancouver, CA, 2006, Paper # FUZZ-4281.
- [23] M.C.A. Melgarejo, A. Garcia, and P.-Reyes. Pro-two: A hardware based platform for real time type-2 fuzzy inference. In: *Proceedings of IEEE FUZZ Conference*, Budapest, Hungary, 2004.
- [24] M.C.A. Melgarejo, P.-Reyes, and A. Garcia. Computational model and architectural proposal for a hardware type-2 fuzzy system. In: *Proceedings of 2nd IASTED Conference on Neural Network and Computational Intelligence*, Grindewald, 2004, pp. 279–284.
- [25] M.C.A. Melgarejo and C.A. Penha-Reyes. Implementing interval type-2 fuzzy processors. *IEEE Comput. Intell. Mag.* 2 (2007) 63–71.
- [26] J.M. Mendel. Advances in type-2 fuzzy sets and systems. *Inf. Sci.* 177 (2007) 84–110.
- [27] J.M. Mendel and H. Wu. New results about the centroid of an interval type-2 fuzzy set, including the centroid of a fuzzy granule. *Inf. Sci.* 177 (2007) 360–377.
- [28] F. Liu and J.M. Mendel. Aggregation using the fuzzy weighted average, as computed by the KM algorithms. *IEEE Trans. Fuzzy Syst.* 16 (2008), 1–12.
- [29] J.M. Mendel. On a 50% savings in the computation of the centroid of a *symmetrical* interval type-2 fuzzy set. *Inf. Sci.* 172 (2005) 417–430.
- [30] J.M. Mendel and H. Wu. Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 1, forward problems. *IEEE Trans. Fuzzy Syst.* 14 (2006) 781–792.
- [31] J.M. Mendel and H. Wu. Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 2, inverse problems. *IEEE Trans. Fuzzy Syst.* 15 (2007) 301–308.
- [32] J.M. Mendel and H. Wu. Type-2 fuzzistics for non-symmetric interval type-2 fuzzy sets: forward problems. *IEEE Trans. Fuzzy Syst.* 15 (2007) 916–930.
- [33] W.M. Dong and F.S. Wong. Fuzzy weighted averages and implementation of the extension principle. *Fuzzy Sets Syst.* 21 (1987) 183–199.
- [34] D. Dubois, H. Fargier, and J. Fortin. A generalized vertex method for computing with fuzzy intervals. In: *Proceedings of FUZZ IEEE Conference*, Budapest, Hungary, 2004, pp. 541–546.

- [35] Y.-Y. Guh, C.-C. Hon, and E.S. Lee. Fuzzy weighted average: The linear programming approach via Charnes and Cooper's rule. *Fuzzy Sets Syst.* 117 (2001) 157–160.
- [36] Y.-Y. Guh, C.-C. Hon, K.-M. Wang, and E.S. Lee. Fuzzy weighted average: A max-min paired elimination method. *Comput. Math. Appl.* 32 (1996) 115–123.
- [37] D.H. Lee and D. Park. An efficient algorithm for fuzzy weighted average. *Fuzzy Sets Syst.* 87 (1997) 39–45.
- [38] T.-S. Liou, M.-J.J. Wang. Fuzzy weighted average: An improved algorithm. *Fuzzy Sets Syst.* 49 (1992) 307–315.
- [39] G.J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, Upper Saddle River, NJ, 1995.
- [40] D. Wu and J.M. Mendel. Aggregation using the linguistic weighted average and interval type-2 fuzzy sets. *IEEE Trans. Fuzzy Syst.* 15 (2007) 1145–1161.
- [41] L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Trans. Fuzzy Syst.* 4 (1996) 103–111.
- [42] J. Lawry, J. Shanahan, and A. Ralescu (eds). *Modeling with Words, Lecture Notes in Artificial Intelligence*, Vol. 2873. Springer, New York, 2003.
- [43] J.M. Mendel. Computing with words, when words can mean different things to different People. In: *Proceedings of Third International ICSC Symposium on Fuzzy Logic and Applications*, Rochester University, Rochester, NY, 1999.
- [44] J.M. Mendel. The perceptual computer: An architecture for computing with words. In: *Proceedings of Modeling with Words Workshop, in the Proceedings of FUZZ-IEEE Conference*, Melbourne, Australia, 2001, pp. 35–38.
- [45] J.M. Mendel. An architecture for making judgments using computing with words. *Int. J. Appl. Math. Comput. Sci.* 12 (2002) 325–335.
- [46] J.M. Mendel. Fuzzy sets for words: A new beginning. In: *Proceedings of FUZZ-IEEE Conference*, St. Louis, MO, 2003, pp. 37–42.
- [47] P.P. Wang (ed.). *Computing with Words*, John Wiley & Sons, New York, 2001.
- [48] L.A. Zadeh. From computing with numbers to computing with words – from manipulation of measurements to manipulation of perceptions. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* 4 (1999) 105–119.
- [49] L.A. Zadeh and J. Kacprzyk (eds). *Computing with Words in Information/Intelligent Systems 1 & 2*. Physica-Verlag, New York, 1999.
- [50] G.M. Mendez and O. Castillo. Interval type-2 TSK fuzzy logic systems using hybrid learning algorithm. In: *Proceedings of IEEE FUZZ Conference*, Reno, NV, 2005, pp. 230–235.
- [51] C.-F. Hwang and C.-H. Rhee. An interval type-2 fuzzy spherical shells algorithm. In: *Proceedings of IEEE FUZZ Conference*, Budapest, Hungary, 2004.
- [52] C.-F. Hwang and C.-H. Rhee. Uncertain fuzzy clustering: interval type-2 fuzzy approach to c-means. *IEEE Trans. Fuzzy Syst.* 15 (2007) 107–120.
- [53] F.C.-H. Rhee. Uncertain fuzzy clustering: Insights and recommendations. *IEEE Comput. Intell. Mag.* 2 (2007) 44–56.
- [54] F.C.-H. Rhee and C. Hwang. A type-2 fuzzy c-means clustering algorithm. In: *Proceedings of IEEE FUZZ Conference*, Melbourne, Australia, 2001, pp. 1926–1929.
- [55] Q. Liang and J.M. Mendel. Decision feedback equalizer for nonlinear time-varying channels using type-2 fuzzy adaptive filters. In: *Proceedings of IEEE-FUZZ Conference*, San Antonio, TX, 2000.
- [56] Q. Liang and J.M. Mendel. Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters. *IEEE Trans. Fuzzy Syst.* 8 (2000) 551–563.
- [57] J.M. Mendel. Uncertainty, fuzzy logic, and signal processing. *Signal Process. J.* 80 (2000) 913–933.
- [58] Q. Liang and J.M. Mendel. Overcoming time-varying co-channel interference using type-2 fuzzy adaptive filter. *IEEE Trans. Circuits Syst.* 47 (2000) 1419–1428.
- [59] Q. Liang and L. Wang. Sensed signal strength forecasting for wireless sensors using interval type-2 fuzzy logic system. In: *Proceedings of IEEE FUZZ Conference*, Reno, NV, 2005, pp. 25–30.
- [60] H. Shu and Q. Liang. Wireless sensor network lifetime analysis using interval type-2 fuzzy logic systems. In: *Proceedings of IEEE FUZZ Conference*, Reno, NV, 2005, pp. 19–24.
- [61] O. Castillo, G. Huesca, and F. Valdez. Evolutionary computing for optimizing type-2 fuzzy systems in intelligent control of non-linear dynamic plants. In: *Proceedings North American Fuzzy Information Processing Society (NAFIPS)*, Ann Arbor, MI, 2005, pp. 247–251.
- [62] O. Castillo, P. Melin, and N. Cazarez. Design of stable type-2 fuzzy logic controllers based on a fuzzy Lyapunov approach. In: *Proceedings of IEEE-FUZZ Conference*, Vancouver, CA, 2006, Paper # FUZZ-4123.
- [63] S. Coupland, M. Gongora, R.I. John, and K. Wills. A comparative study of fuzzy logic controllers for autonomous robots. In: *Proceedings of IPMU*, Paris, France, 2006, pp. 1332–1339.
- [64] J. Figueiroa, J. Posada, J. Soriano, M. Melgarejo, and S. Rojas. A type-2 fuzzy controller for tracking mobile objects in the context of robotic soccer games. In: *Proceedings of IEEE FUZZ Conference*, Reno, NV, 2005, pp. 359–364.

- [65] H. Hagras. A type-2 fuzzy logic controller for autonomous mobile robots. In: *Proceedings of IEEE FUZZ Conference*, Budapest, Hungary, 2004.
- [66] H. Hagras. A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Trans. Fuzzy Syst.* 12 (2004) 524–539.
- [67] H. Hagras. Type-2 FLCs: A new generation of fuzzy controllers. *IEEE Comput. Intell. Mag.* 2 (2007) 30–43.
- [68] K.C. Wu. Fuzzy interval control of mobile robots. *Comput. Electr. Eng.* 22 (1996) 211–229.
- [69] C.-H. Lee, H.-Y. Pan, H.-H. Chang, and B.-H. Wang. Decoupled adaptive type-2 fuzzy controller (DAT2FC) design for nonlinear TORA systems, In: *Proceedings of IEEE-FUZZ Conference*, Vancouver, CA, 2006, Paper # FUZZ-4305.
- [70] P.-Z. Lin, C.-F. Hsu, and T.-T. Lee. Type-2 fuzzy logic controller design for buck DC-DC converters. In: *Proceedings of IEEE FUZZ Conference*, Reno, NV, 2005, pp. 365–370.
- [71] C. Lynch, H. Hagras, and V. Callaghan. Embedded type-2 FLC for real-time speed control of marine and traction diesel engines. In: *Proceedings of IEEE FUZZ Conference*, Reno, NV, 2005, pp. 347–352.
- [72] P. Melin and O. Castillo. A new method for adaptive model-based control of non-linear plants using type-2 fuzzy logic and neural networks. In: *Proceedings of IEEE FUZZ Conference*, St. Louis, MO, 2003, pp. 420–425.
- [73] R. Sepulveda, O. Castillo, and P. Melin. A. Rodriguez-Diaz, O. Montiel, Integrated development platform for intelligent control based on type-2 fuzzy logic. In: *Proceedings of North American Fuzzy Information Processing Society*, NAFIPS, Ann Arbor, MI, 2005, pp. 607–610.
- [74] W.W. Tan and J. Lai. Development of a type-2 fuzzy proportional controller. In: *Proceedings of IEEE FUZZ Conference*, Budapest, Hungary, 2004.
- [75] D. Wu and W.W. Tan. A type-2 fuzzy logic controller for the liquid-level process. In: *Proceedings of IEEE FUZZ Conference*, Budapest, Hungary, 2004, pp. 953–958.
- [76] D.A. Chiang, L.-R. Chow, and N.-C. Hsien. Fuzzy information in extended fuzzy relational databases. *Fuzzy Sets Syst.* 92 (1997) 1–20.
- [77] A. Niewiadomski, J. Kacprzyk, J. Ochelska, and P.S. Szczepaniak. *Interval-Valued Linguistic Summaries of Databases*. Control Cybern. Syst. Res. Institute, Polish Academy of Science, Warsaw, Poland, 2005.
- [78] A. Niewiadomski and P.S. Szczepaniak. News generating based on interval type-2 linguistic summaries of databases. In: *Proceedings of IPMU*, Paris, France, 2006, pp. 1324–1331.
- [79] J.L. Chaneau, M. Gunaratne, and A.G. Altschaeffl. An application of type-2 sets to decision making in engineering. In: J. Bezdek (ed.), *Analysis of Fuzzy Information, vol. II: Artificial Intelligence and Decision Systems*. CRC, Boca Raton, FL, 1987.
- [80] R.R. Yager. Fuzzy subsets of type II in decisions. *J. Cybern.* 10 (1980) 137–159.
- [81] T. Ozen, J.M. Garibaldi, and S. Musikaswan. Preliminary investigations into modeling the variation in human decision making, In: *Proceedings of IPMU*, Perugia, Italy, 2004, pp. 641–648.
- [82] F. Doctor, H. Hagras, and V. Callaghan. A type-2 fuzzy embedded agent for ubiquitous computing environment. In: *Proceedings of IEEE FUZZ Conference*, Budapest, Hungary, 2004.
- [83] F. Doctor, H. Hagras, and V. Callaghan. A type-2 fuzzy embedded agent to realise ambient intelligence in ubiquitous computing environments. *Inf. Sci.* 171 (2005) 309–334.
- [84] F. Doctor, H. Hagras, and V. Callaghan. Life long learning approach for type-2 fuzzy embedded agents in ambient intelligent environments. In: *Proceedings of IEEE-FUZZ Conference*, Vancouver, CA, 2006, Paper # FUZZ-4145.
- [85] H. Hagras, F. Doctor, V. Callaghan, and A. Lopez. An incremental adaptive life long learning approach for type-2 fuzzy embedded agents in ambient intelligent environments. *IEEE Trans. Fuzzy Syst.* 15 (2007) 41–55.
- [86] L. Di Lascio, A. Gisolfi, and A. Nappi. Medical differential diagnosis through type-2 fuzzy sets. In: *Proceedings of IEEE FUZZ Conference*, Reno, NV, 2005, pp. 371–376.
- [87] R.I. John and P.R. Innocent. Modeling uncertainty in clinical diagnosis using fuzzy logic. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 35 (2005) 1340–1350.
- [88] K. Wills, R.I. John, and S. Lake. Combining categories in nursing assessment using interval valued fuzzy sets. In: *Proceedings of IPMU*, Perugia, Italy, 2004.
- [89] J. Zeng and Z.-Q. Liu. Type-2 fuzzy hidden Markov models and their application to speech recognition. *IEEE Trans. Fuzzy Syst.* 14 (2006) 454–467.
- [90] N.N. Karnik and J.M. Mendel. Applications of type-2 fuzzy logic systems: Handling the uncertainty associated with surveys. In: *Proceedings of FUZZ-IEEE Conference*, Seoul, Korea, 1999.
- [91] F.C.-H Rhee and C. Hwang. An interval type-2 fuzzy perceptron. In: *Proceedings of IEEE FUZZ Conference*, Honolulu, HI, 2002.
- [92] S. Auephanwiriyakul and S. Dhompongsa. An investigation of a linguistic perceptron in a nonlinear decision boundary problem. In: *Proceedings of IEEE FUZZ Conference*, Vancouver, CA, 2006.

- [93] X. Chen, R. Harrison, Y.-Q. Zhang, and Y. Qiu. A multi-SVM fusion model using type-2 FLS. In: *Proceedings of IEEE-FUZZ Conference*, Vancouver, CA, 2006, Paper # FUZZ-4489.
- [94] Q. Liang and J.M. Mendel. MPEG VBR video traffic modeling and classification using fuzzy techniques. *IEEE Trans. Fuzzy Syst.* 9 (2000) 183–193.
- [95] H. Wu and J.M. Mendel. Classification of battlefield ground vehicles using acoustic features and fuzzy logic rule-based classifiers. *IEEE Trans. Fuzzy Syst.* 15 (2007) 56–72.
- [96] F.C.-H. Rheu and C. Hwang. An interval type-2 fuzzy K -nearest neighbor. In: *Proceedings of IEEE FUZZ Conference*, Honolulu, HI, 2002, pp. 802–807.
- [97] J. Zeng and Z.-Q. Liu. Type-2 fuzzy sets for handling uncertainty in pattern recognition. In: *Proceedings of IEEE-FUZZ Conference*, Vancouver, CA, 2006, Paper # FUZZ-4382.
- [98] P. Melin and O. Castillo. A new approach for quality control of sound speakers combining type-2 fuzzy logic and the fractal dimension. In: *Proceedings of International Conference of NAFIPS 2003*, Chicago, IL, 2003, pp. 20–25.
- [99] R.I. John. Type-2 inferencing and community transport scheduling. In: *Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing, EUFIT'96*, Aachen, Germany, 1996, pp. 1369–1372.
- [100] O. Castillo and P. Melin. Adaptive noise cancellation using type-2 fuzzy logic and neural networks. In: *Proceedings of IEEE FUZZ Conference*, Budapest, Hungary, 2004.
- [101] R.I. John, P.R. Innocent, and M.R. Barnes. Type-2 fuzzy Sets and neuro-fuzzy clustering for radiographic tibia images. In: *Proceedings of Sixth International Conference on Fuzzy Systems*, Barcelona, Spain, 1997, pp. 1375–1380; also, in *Proceedings of IEEE International Conference on Fuzzy Systems*, Anchorage, AK, 1998, pp. 1373–1376.
- [102] N.N. Karnik and J.M. Mendel. Applications of type-2 fuzzy logic systems to forecasting of time-series. *Inf. Sci.* 120 (1999) 89–111.
- [103] M. de los Angeles Hernandez Medina and G.M. Mendez. Modeling and prediction of the MXNUSD exchange rate using interval singleton type-2 fuzzy logic systems. *IEEE Comput. Intell. Mag.* 2 (2007) 5–8.
- [104] G.M. Mendez and M. de los Angeles Hernandez. Modelling and prediction of the MXNUSD exchange rate using interval singleton type-2 fuzzy logic systems. In: *Proceedings of IEEE-FUZZ Conference*, Vancouver, CA, 2006, Paper # FUZZ-4090.
- [105] U. Pareek and I.N. Kar. Estimating compressor discharge pressure of gas turbine power plant using type-2 fuzzy logic systems. In: *Proceedings of IEEE-FUZZ Conference*, Vancouver, CA, 2006, Paper # FUZZ-4171.
- [106] P. Fisher. What is where? type-2 fuzzy sets for geographical information. *IEEE Comput. Intell. Mag.* 2 (2007) 9–14.
- [107] S. Rahimi, M. Cobb, A. Zhou, D. Ali, H. Yang, and F.E. Petry. An inexact inferencing strategy for spatial objects with determined and indeterminate boundaries. In: *Proceedings of IEEE FUZZ Conference*, St. Louis, MO, 2003, pp. 778–783.
- [108] Q. Liang, N.N. Karnik, and J.M. Mendel. Connection admission control in ATM networks using survey-based type-2 fuzzy logic systems. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* 30 (2000) 329–339.

26

Design of Intelligent Systems with Interval Type-2 Fuzzy Logic

Oscar Castillo and Patricia Melin

26.1 Introduction

Uncertainty affects decision making and appears in a number of different forms. The concept of information is fully connected with the concept of uncertainty. The most fundamental aspect of this connection is that the uncertainty involved in any problem-solving situation is a result of some information deficiency, which may be incomplete, imprecise, fragmentary, not fully reliable, vague, contradictory, or deficient in some other way [1]. Uncertainty is an attribute of information [2]. The general framework of fuzzy reasoning allows handling much of this uncertainty; fuzzy systems employ type-1 fuzzy sets, which represent uncertainty by numbers in the range $[0, 1]$. When something is uncertain, like a measurement, it is difficult to determine its exact value, and of course type-1 fuzzy sets make more sense than using sets [2–4]. However, it is not reasonable to use an accurate membership function for something uncertain, so in this case what we need is another type of fuzzy sets, those which are able to handle these uncertainties, the so-called type-2 fuzzy sets [5, 6]. So, the amount of uncertainty in a system can be reduced by using type-2 fuzzy logic because it offers better capabilities to handle linguistic uncertainties by modeling vagueness and unreliability of information [7–9].

Recently, we have seen the use of type-2 fuzzy sets in fuzzy logic systems (FLS) in different areas of application [8, 10–13]. A novel approach for realizing the vision of ambient intelligence in ubiquitous computing environments (UCEs) is based on embedding intelligent agents that use type-2 fuzzy systems which are able to handle the different sources of uncertainty and imprecision in UCEs to give a good response [14]. There are also papers with emphasis on the implementation of type-2 FLS [15, 16] and in others, it is explained how type-2 fuzzy sets let us model and minimize the effects of uncertainties in rule-base FLS [17, 18]. There is also a paper that provides mathematical formulas and computational flowcharts for computing the derivatives that are needed to implement steepest descent parameter tuning algorithms for type-2 FLSs [19]. Some research works are devoted to solve real-world applications in different areas; for example, in signal processing, type-2 fuzzy logic is applied in prediction of the Mackey–Glass chaotic time series with uniform noise presence [20, 21]. In medicine, an expert system was developed for solving the problem of umbilical acid–base (UAB) assessment [22]. In industry, type-2 fuzzy logic and neural networks were used in the control of non-linear dynamic plants [23–25]; also we can find interesting studies in the field of mobile robots [26, 27].

In this chapter we deal with the application of interval type-2 fuzzy control to non-linear dynamic systems. It is a well-known fact that in the control of real systems, the instrumentation elements

(instrumentation amplifier, sensors, digital-to-analog, analog-to-digital converters, etc.) introduce some sort of unpredictable values in the information that has been collected [11, 12]. So, the controllers designed under idealized conditions tend to behave in an inappropriate manner [28]. Since, uncertainty is inherent in the design of controllers for real-world applications, we are presenting how to deal with this problem using type-2 fuzzy logic controller (FLC) to reduce the effects of imprecise information. We are supporting this statement with experimental results, qualitative observations, and quantitative measures of errors. For quantifying the errors, we utilized three widely used performance criteria, which are integral of square error (ISE), integral of the absolute value of the error (IAE), and integral of the time multiplied by the absolute value of the error (ITAE) [28]. We also consider the application of interval type-2 fuzzy logic to the problem of forecasting chaotic time series [29].

26.2 Fuzzy Logic Systems

In this section, a brief overview of type-1 and type-2 fuzzy systems is presented. This overview is considered as necessary to understand the basic concepts needed to understand the methods and algorithms presented later in the chapter.

26.2.1 Type-1 Fuzzy Logic Systems

In the 1940s and 1950s, many researchers showed that dynamic systems could be mathematically modeled using differential equations. In these works we have the foundations of the control theory, which, in addition with the transform theory (Laplace's theory), provided an extremely powerful means of analyzing and designing control systems [30]. These theories were developed until the seventies, when the area was called systems theory to indicate its definitiveness [31].

Soft computing techniques have become an important research topic, which can be applied in the design of intelligent controllers [10, 11]. These techniques have tried to avoid the above-mentioned drawbacks, and they allow us to obtain efficient controllers, which utilize the human experience in a more natural form than the conventional mathematical approach [3, 4, 32, 33]. In the cases in which a mathematical representation of the controlled system is difficult to obtain, the process operator has the knowledge and the experience to express the relationships existing in the process behavior.

An FLS, described completely in terms of type-1 fuzzy sets, is called a type-1 FLS. It is composed by a knowledge base, which comprises the information given by the process operator in form of linguistic control rules. A fuzzification interface has the effect of transforming crisp data into fuzzy sets. An inference system uses the fuzzy sets in conjunction with the knowledge base to make inferences by means of a reasoning method. Finally, a defuzzification interface translates the fuzzy control action so obtained to a real control action using a defuzzification method [30].

In this chapter, the implementation of the fuzzy controller in terms of type-1 fuzzy sets has two input variables, which are the error $e(t)$, the difference between the reference signal and the output of the process, as well as the error variation $\Delta e(t)$,

$$e(t) = r(t) - y(t), \quad (1)$$

$$\Delta e(t) = e(t) - e(t - 1), \quad (2)$$

so the control system can be represented as in Figure 26.1.

26.2.2 Type-2 Fuzzy Logic Systems

If for a type-1 membership function, as in Figure 26.2, we blur it to the left and to the right, as illustrated in Figure 26.3, then a type-2 membership function is obtained. In this case, for a specific value x' , the membership function (u') takes on different values, which are not all weighted the same, so we can assign an amplitude distribution to all of those points.

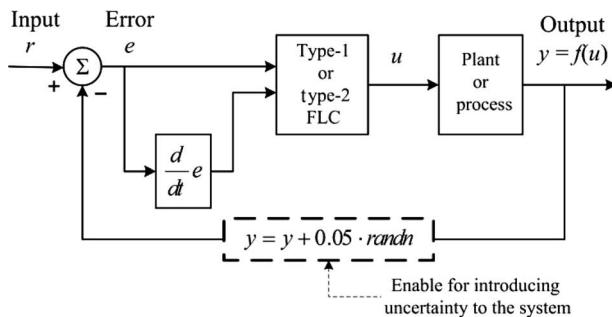


Figure 26.1 System used for obtaining the experimental results for the class I experiments

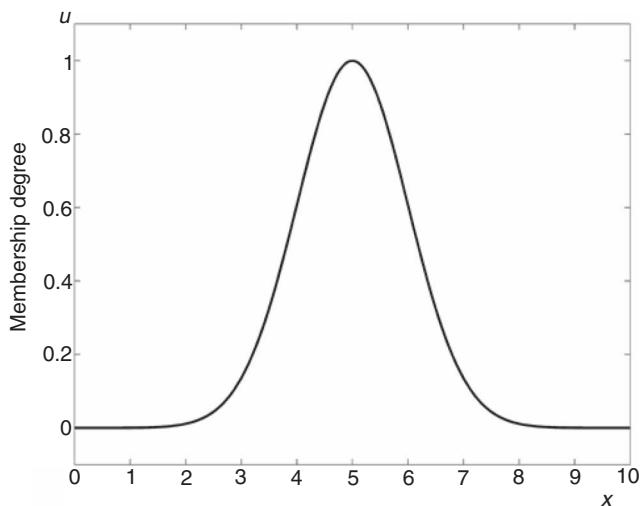


Figure 26.2 Type-1 membership function

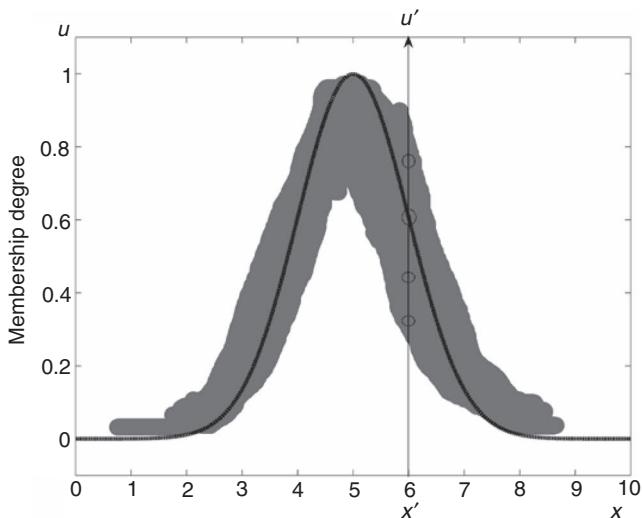


Figure 26.3 Blurred type-1 membership function

Doing this for all $x \in X$, we create a three-dimensional membership function – a type-2 membership function – that characterizes a type-2 fuzzy set [5, 34]. A type-2 fuzzy set \tilde{A} is characterized by the membership function:

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) \mid \forall x \in X, \forall u \in J_x \subseteq [0, 1]\}, \quad (3)$$

in which $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$. Another expression for \tilde{A} is

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad J_x \subseteq [0, 1], \quad (4)$$

where \iint denotes the union over all admissible input variables x and u . For discrete universes of discourse, \int is replaced by \sum [35]. In fact, $J_x \subseteq [0, 1]$ represents the primary membership of x and $\mu_{\tilde{A}}(x, u)$ is a type-1 fuzzy set known as the secondary set. Hence, a type-2 membership grade can be any subset in $[0, 1]$, the primary membership, and corresponding to each primary membership, there is a secondary membership (which can also be in $[0, 1]$) that defines the possibilities for the primary membership [18, 36]. Uncertainty is represented by a region, which is called the footprint of uncertainty (FOU). When $\mu_{\tilde{A}}(x, u) = 1, \forall u \in J_x \subseteq [0, 1]$, we have an interval type-2 membership function, as shown in Figure 26.4. The uniform shading for the FOU represents the entire interval type-2 fuzzy set and it can be described in terms of an upper membership function $\bar{\mu}_{\tilde{A}}(x)$ and a lower membership function $\underline{\mu}_{\tilde{A}}(x)$. Please note that here we are using crisp values for the bounds of the interval type-2 membership function, but it is also possible for the bounds to be fuzzy rather than crisp, as this would give another degree of approximation to model uncertainty. However, in this case we will analyze here the case of crisp bounds.

An FLS described using at least one type-2 fuzzy set is called a type-2 FLS. Type-1 FLSs are unable to directly handle rule uncertainties, because they use type-1 fuzzy sets that are certain [34]. On the other hand, type-2 FLSs are very useful in circumstances where it is difficult to determine an exact membership function, and there are measurement uncertainties [5, 17, 18].

It is known that type-2 fuzzy sets enable modeling and minimizing the effects of uncertainties in rule-base FLS. Unfortunately, type-2 fuzzy sets are more difficult to use and understand than type-1 fuzzy sets; hence, their use is not widespread yet. As a justification for the use of type-2 fuzzy sets, in [17] are

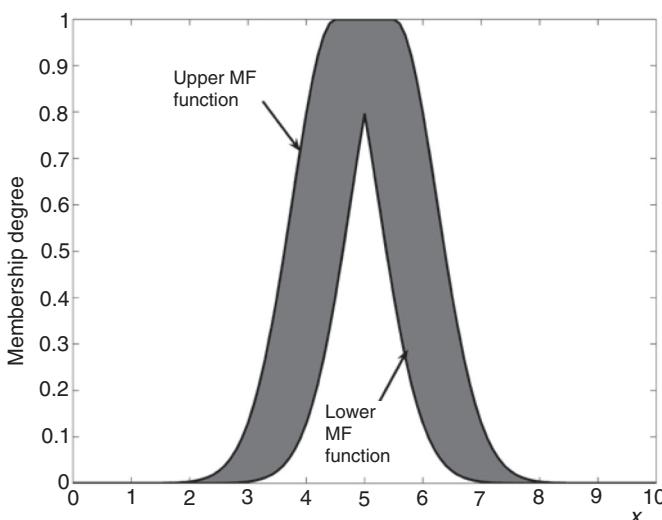


Figure 26.4 Interval type-2 membership function

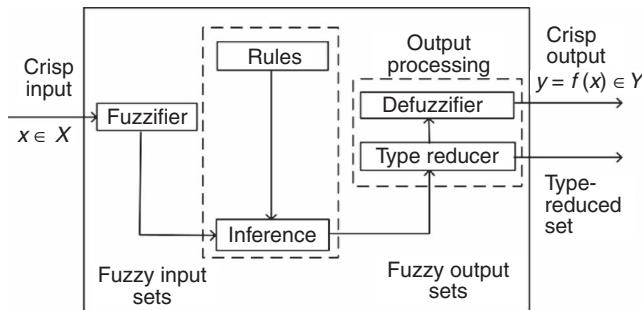


Figure 26.5 Type-2 fuzzy logic system

mentioned at least four sources of uncertainties not considered in type-1 FLSs:

1. The meanings of the words that are used in the antecedents and consequents of rules can be uncertain. (Words mean different things to different people.)
2. Consequents may have histogram of values associated with them, especially when knowledge is extracted from a group of experts who do not all agree.
3. Measurements that activate a type-1 FLS may be noisy and therefore uncertain.
4. The data used to tune the parameters of a type-1 FLS may also be noisy.

All of these uncertainties translate into uncertainties about fuzzy set membership functions. Type-1 fuzzy sets are not able to directly model such uncertainties because their membership functions are totally crisp. On the other hand, type-2 fuzzy sets are able to model such uncertainties because their membership functions are themselves fuzzy. A type-1 fuzzy set is a special case of a type-2 fuzzy set; its secondary membership function is a subset with only one element, unity.

A type-2 FLS is again characterized by IF-THEN rules, but its antecedent or consequent sets are now of type 2. Type-2 FLSs can be used when the circumstances are too uncertain to determine exact membership grades such as when the training data are corrupted by noise. Similar to a type-1 FLS, a type-2 FLS includes a fuzzifier, a rule base, fuzzy inference engine, and an output processor, as we can see in Figure 26.5. The output processor includes type reducer and defuzzifier; it generates a type-1 fuzzy set output (from the type reducer) or a crisp number (from the defuzzifier) [7, 35–37]. Now we will explain each of the blocks of Figure 26.5.

Type-2 membership functions can be formed by using the information provided by a group of experts that do not fully agree on the parameter values of membership functions. This difference of opinion gives rise to ranges of values for the membership function's parameters.

26.2.3 Fuzzifier

The fuzzifier maps a crisp point $\mathbf{x} = (x_1, \dots, x_p)^T \in X_1 \times X_2 \times \dots \times X_p \equiv \mathbf{X}$ into a type-2 fuzzy set \tilde{A}_x in \mathbf{X} [5], interval type-2 fuzzy sets in this case. We will use type-2 singleton fuzzifier; in a singleton fuzzification, the input fuzzy set has only a single point on non-zero membership [17]. \tilde{A}_x is a type-2 fuzzy singleton if $\mu_{\tilde{A}_x}(x) = 1/1$ for $\mathbf{x} = \mathbf{x}'$ and $\mu_{\tilde{A}_x}(x) = 1/0$ for all other $\mathbf{x} \neq \mathbf{x}'$ [27].

26.2.4 Rules

The structure of rules in a type-1 FLS and a type-2 FLS is the same, but in the latter the antecedents and the consequents will be represented by type-2 fuzzy sets. So for a type-2 FLS with p inputs $x_1 \in X_1, \dots, x_p \in X_p$ and one output $y \in Y$, multiple input single output (MISO), if we assume that there are M rules, the l th rule in the type-2 FLS can be written as follows [5]:

$$R^l: \text{IF } x_1 \text{ is } \tilde{F}_1^l \text{ and } \dots \text{ and } x_p \text{ is } \tilde{F}_p^l, \text{ THEN } y \text{ is } \tilde{G}^l \quad l = 1, \dots, M. \quad (5)$$

26.2.5 Inference

In the type-2 FLS, the inference engine combines rules and gives a mapping from input type-2 fuzzy sets to output type-2 fuzzy sets. It is necessary to compute the join \sqcup , (unions), and the meet \sqcap (intersections), as well as extended sup star compositions (sup star compositions) of type-2 relations [27]. If $\tilde{F}_1^l \times \cdots \times \tilde{F}_p^l = \tilde{A}^l$, equation (5) can be rewritten as

$$R^l : \tilde{F}_1^l \times \cdots \times \tilde{F}_p^l \rightarrow \tilde{G}^l = \tilde{A}^l \rightarrow \tilde{G}^l, \quad l = 1, \dots, M. \quad (6)$$

R^l is described by the membership function $\mu_{R^l}(\mathbf{x}, y) = \mu_{R^l}(x_1, \dots, x_p, y)$, where

$$\mu_{R^l}(\mathbf{x}, y) = \mu_{\tilde{A}^l \rightarrow \tilde{G}^l}(\mathbf{x}, y) \quad (7)$$

can be written as [27]

$$\begin{aligned} \mu_{R^l}(\mathbf{x}, y) &= \mu_{\tilde{A}^l \rightarrow \tilde{G}^l}(\mathbf{x}, y) = \mu_{\tilde{F}_1^l}(x_1) \sqcap \cdots \sqcap \mu_{\tilde{F}_p^l}(x_p) \sqcap \mu_{\tilde{G}^l}(y) \\ &= [\sqcap_{i=1}^p \mu_{\tilde{F}_i^l}(x_i)] \sqcap \mu_{\tilde{G}^l}(y). \end{aligned} \quad (8)$$

In general, the p -dimensional input to R^l is given by the type-2 fuzzy set $\tilde{A}_{\bar{x}}$ whose membership function is

$$\mu_{\tilde{A}_{\bar{x}}}(\mathbf{x}) = \mu_{\bar{x}_1}(x_1) \sqcap \cdots \sqcap \mu_{\bar{x}_p}(x_p) = \sqcap_{i=1}^p \mu_{\bar{x}_i}(x_i), \quad (9)$$

where $\bar{X}_i (i = 1, \dots, p)$ are the labels of the fuzzy sets describing the inputs. Each rule R^l determines a type-2 fuzzy set $\tilde{B}^l = \tilde{A}_{\bar{x}} \circ R^l$, such that [5]

$$\mu_{\tilde{B}^l}(y) = \mu_{\tilde{A}_{\bar{x}} \circ R^l} = \sqcup_{x \in \mathbf{x}} [\mu_{\tilde{A}_{\bar{x}}}(\mathbf{x}) \sqcap \mu_{R^l}(\mathbf{x}, y)] \quad y \in Y \quad l = 1, \dots, M. \quad (10)$$

This equation is the input/output relation in Figure 26.5 between the type-2 fuzzy set that activates one rule in the inference engine and the type-2 fuzzy set at the output of that engine [5].

In the FLS we used interval type-2 fuzzy sets and meet under product t -norm, so the result of the input and antecedent operations, which are contained in the firing set $\sqcap_{i=1}^p \mu_{\tilde{F}_{ii}}(x'_i \equiv F^l(\mathbf{x}'))$, is an interval type-1 set [5],

$$F^l(\mathbf{x}') = [\underline{f}^l(\mathbf{x}'), \bar{f}^l(\mathbf{x}')] \equiv [\underline{f}^l, \bar{f}^l], \quad (11)$$

where

$$\underline{f}^l(\mathbf{x}') = \underline{\mu}_{\tilde{F}_1^l}(x'_1) * \cdots * \underline{\mu}_{\tilde{F}_p^l}(x'_p), \quad (12)$$

and

$$\bar{f}^l(\mathbf{x}') = \bar{\mu}_{\tilde{F}_1^l}(x'_1) * \cdots * \bar{\mu}_{\tilde{F}_p^l}(x'_p), \quad (13)$$

where $*$ is the product operation.

26.2.6 Type Reducer

The type reducer generates a type-1 fuzzy set output, which is then converted into a crisp output through the defuzzifier. This type-1 fuzzy set is also an interval set; for the case of our FLS, we used center of

sets (cos) type reduction, Y_{cos} , which is expressed as [5]

$$Y_{\text{cos}}(\mathbf{x}) = [y_l, y_r] = \int_{y^1 \in [y_l^1, y_r^1]} \cdots \int_{y^M \in [y_l^M, y_r^M]} \int_{f^1 \in [f^1, \bar{f}^1]} \cdots \int_{f^M \in [f^M, \bar{f}^M]} 1 \Bigg/ \frac{\sum_{i=1}^M f^i y^i}{\sum_{i=1}^M f^i}. \quad (14)$$

This interval set is determined by its two end points, y_l and y_r , which corresponds to the centroid of the type-2 interval consequent set \tilde{G}^i [5],

$$C_{\tilde{G}^i} = \int_{\theta_1 \in J_{y^1}} \cdots \int_{\theta_N \in J_{y^N}} 1 \Bigg/ \frac{\sum_{i=1}^N y_i \theta_i}{\sum_{i=1}^N \theta_i} = [y_l^i, y_r^i]. \quad (15)$$

Before the computation of $Y_{\text{cos}}(\mathbf{x})$, we must evaluate equation (15) and its two end points, y_l and y_r . If the values of f_i and y_i that are associated with y_l are denoted f_l^i and y_l^i , respectively, and the values of f_i and y_i that are associated with y_r are denoted f_r^i and y_r^i , respectively, from equation (14), we have [5]:

$$y_l = \frac{\sum_{i=1}^M f_l^i y_l^i}{\sum_{i=1}^M f_l^i}, \quad (16)$$

$$y_r = \frac{\sum_{i=1}^M f_r^i y_r^i}{\sum_{i=1}^M f_r^i}. \quad (17)$$

26.2.7 Defuzzifier

From the type reducer we obtain an interval set Y_{cos} . To defuzzify it we use the average of y_l and y_r , so the defuzzified output of an interval singleton type-2 FLS is [35]

$$y(\mathbf{x}) = \frac{y_l + y_r}{2}. \quad (18)$$

In this chapter, we are simulating the fact that the instrumentation elements (instrumentation amplifier, sensors, digital-to-analog, analog-to-digital converters, etc.) are introducing some sort of unpredictable values in the collected information. In the case of the implementation of the type-2 FLC, we have the same characteristics as in type-1 FLC, but we use type-2 fuzzy sets as membership functions for the inputs and for the output.

26.2.8 Performance Criteria

For evaluating the transient closed-loop response of a computer control system we can use the same criteria that are normally used for adjusting constants in PID (proportional integral derivative) controllers [38]. These are as follows [28]:

1. Integral of square error (ISE):

$$\text{ISE} = \int_0^\infty [e(t)]^2 dt. \quad (19)$$

2. Integral of the absolute value of the error (IAE):

$$\text{IAE} = \int_0^\infty |e(t)| dt. \quad (20)$$

3. Integral of the time multiplied by the absolute value of the error (ITAE):

$$\text{ITAE} = \int_0^{\infty} t|e(t)|dt. \quad (21)$$

The selection of the criteria depends on the type of response desired; the errors will contribute differently for each criterion, so we have that large errors will increase the value of ISE more heavily than to IAE. ISE will favor responses with smaller overshoot for load changes, but ISE will give longer settling time. In ITAE, time appears as a factor, and therefore, ITAE will penalize heavily errors that occur late in time, but virtually ignores errors that occur early in time. Designing using ITAE will give us the shortest settling time, but it will produce the largest overshoot among the three criteria considered. Designing considering IAE will give us an intermediate result; in this case, the settling time will not be so large than using ISE nor so small than using ITAE, and the same applies for the overshoot response. The selection of a particular criterion is depending on the type of desired response.

26.3 Experimental Results for Intelligent Control

The experimental results are devoted to show comparisons in the system's response in a feedback controller when using a type-1 FLC or a type-2 FLC. A set of five experiments is described in this section. The first two experiments were performed in ideal conditions, i.e., without any kind of disturbance. In the last three experiments, Gaussian noise was added to the feedback loop with the purpose of simulating, in a global way, the effects of uncertainty from several sources.

Figure 26.1 shows the feedback control system that was used for obtaining the simulation results. The complete system was simulated in the Matlab programming language, and the controller was designed to follow the input as closely as possible. The plant is a non-linear system that is modeled using equation (22)

$$y(i) = 0.2 \cdot y(i-3) + 0.07y(i-2) + 0.9 \cdot y(i-1) + 0.05 \cdot u(i-1) + 0.5 \cdot u(i-2). \quad (22)$$

To illustrate the dynamics of this non-linear system, two different inputs are applied, first the input indicated by equation (23), which is shown in Figure 26.6, and whose system's response is in Figure 26.7.

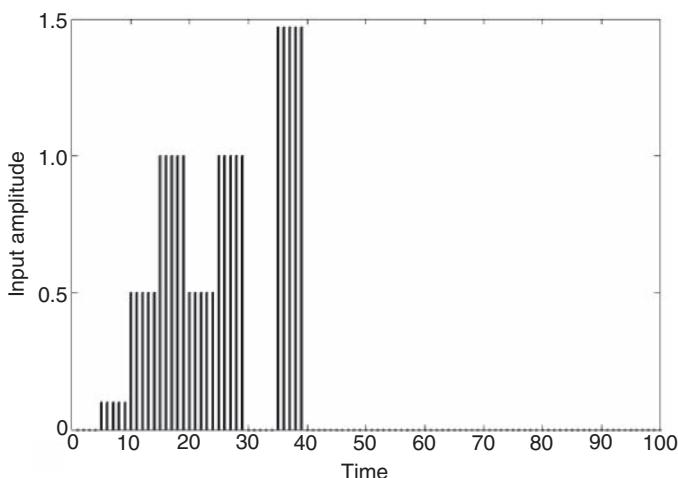


Figure 26.6 Test sequence applied to the model of the plant given in equation (23)

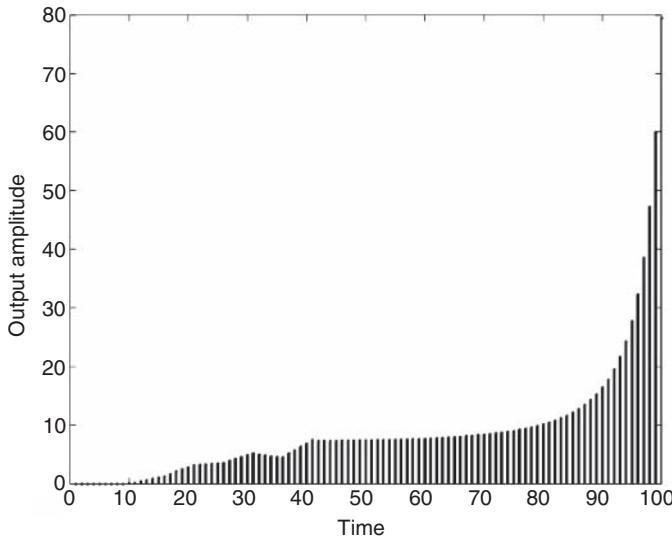


Figure 26.7 System response for the inputs given in equation (23), which is illustrated in Figure 26.6

$$u(i) = \begin{cases} 0 & 1 \leq i < 5 \\ .1 & 5 \leq i < 10 \\ .5 & 10 \leq i < 15 \\ 1 & 15 \leq i < 20 \\ .5 & 20 \leq i < 25 \\ 1 & 25 \leq i < 30 \\ 0 & 30 \leq i < 35 \\ 1.47 & 35 \leq i < 40 \end{cases} \quad (23)$$

Now, for a slightly different input given by equation (24), see Figure 26.8; we have the corresponding system's response in Figure 26.9.

$$u(i) = \begin{cases} 0 & 1 \leq i < 5 \\ .1 & 5 \leq i < 10 \\ .5 & 10 \leq i < 15 \\ 1 & 15 \leq i < 20 \\ .5 & 20 \leq i < 25 \\ 1 & 25 \leq i < 30 \\ 0 & 30 \leq i < 35 \\ 1.4 & 35 \leq i < 40 \end{cases} \quad (24)$$

Going back to the control problem, this system given by equation (22) was used in Figure 26.1, under the name of plant or process; in this figure we can see that the controller's output is applied directly to the plant's input. Since we are interested in comparing the performance between type-1 and type-2 FLC systems, the controller was tested in two ways:

1. One is considering the system as ideal, i.e., not introducing in the modules of the control system any source of uncertainty (experiments 1 and 2).
2. The other one is simulating the effects of uncertain modules (subsystems) response introducing some uncertainty (experiments 3, 4, and 5).

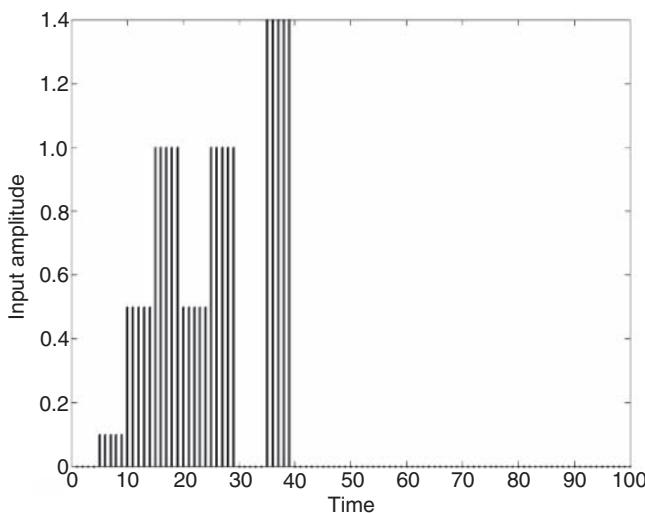


Figure 26.8 A second input to the model for testing the plant response

For both cases, as it is shown in Figure 26.1, the system's output is directly connected to the summing junction, but in the second case, the uncertainty was simulated introducing random noise with normal distribution (the dashed square in Figure 26.1). We added noise to the system's output $y(i)$ using the Matlab's function 'randn,' which generates random numbers with Gaussian distribution. The signal and the added noise in turn were obtained with the programmer's expression (25); the result $y(i)$ was introduced to the summing junction of the controller system. Note that in expression (25) we are using the value 0.05 for experiments 3 and 4 but in the set of tests for experiment 5, we varied this value to obtain different SNR values:

$$y(i) = y(i) + 0.05 \cdot \text{randn}. \quad (25)$$

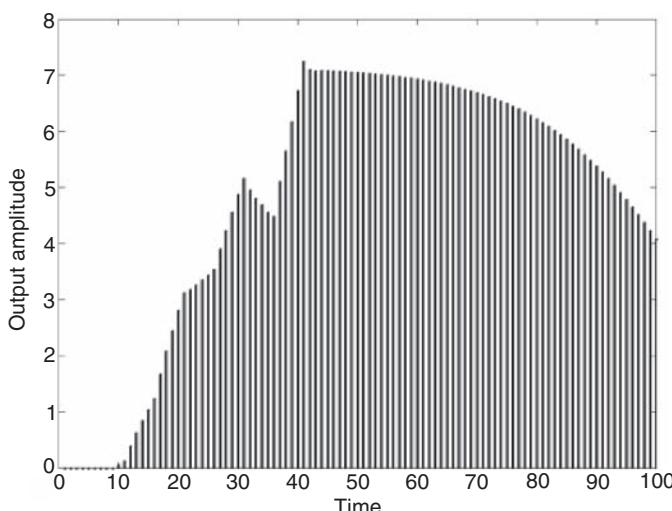


Figure 26.9 Output of the plant when we applied the input given by equation (24), which is illustrated in Figure 26.8

The system was tested using as input, a unit step sequence free of noise, $r(i)$. For evaluating the system's response and comparing between type 1 and type 2 fuzzy controllers, the performance criteria ISE, IAE, and ITAE were used. In Table 26.3, we summarized the values obtained in an ideal system for each criterion considering 400 units of time. For calculating ITAE, a sampling time of $T_s = 0.1$ s was considered.

For all experiments the reference input r is stable and noisy free. In experiments 3 and 4, although the reference appears clean, the feedback at the summing junction is noisy since noise for simulating the overall existing uncertainty in the system was introduced deliberately; in consequence, the controller's inputs $e(t)$ (error) and $\Delta e(t)$ contain uncertainty in the data.

In experiment 5, we tested the systems, type-1 and type-2 FLCs, introducing different values of noise η ; this was done by modifying the signal-to-noise ratio (SNR) [33]:

$$SNR = \frac{\sum |s|^2}{\sum |\eta|^2} = \frac{P_{signal}}{P_{noise}}. \quad (26)$$

Because many signals have a very wide dynamic range [39], SNRs are usually expressed in terms of the logarithmic decibel scale, SNR(db),

$$SNR(db) = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right). \quad (27)$$

In Table 26.4, we show, for different values of SNR(db), the behavior of ISE, IAE, and ITAE for type-1 and type-2 FLCs. In all the cases the results for type-2 FLC are better than type-1 FLC.

In the type-1 FLC, Gaussian membership functions (Gaussian MFs) for the inputs and for the output were used. A Gaussian MF is specified by two parameters $\{c, \sigma\}$:

$$\mu_A(x) = e^{-\frac{1}{2}(\frac{x-c}{\sigma})^2}. \quad (28)$$

c represents the MFs center and σ determines the MFs standard deviation.

For each of the inputs of the type-1 FLC, $e(t)$ and $\Delta e(t)$, three type-1 fuzzy Gaussian MFs were defined as negative, zero, and positive. The universe of discourse for these membership functions is in the range $[-10, 10]$; their centers are $-10, 0$, and 10 respectively, and their standard deviations is 4.2466 as is illustrated in Figures 26.10 and 26.11.

For the output of the type-1 FLC, we have five type-1 fuzzy Gaussian MFs: NG, N, Z, P, and PG. They are in the interval $[-10, 10]$, their centers are $-10, -5, 0, 5$, and 10 , respectively, and their standard

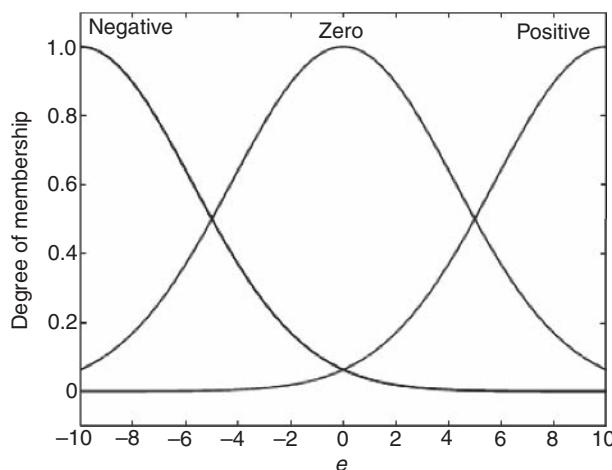


Figure 26.10 Input e membership functions for the type-1 FLC

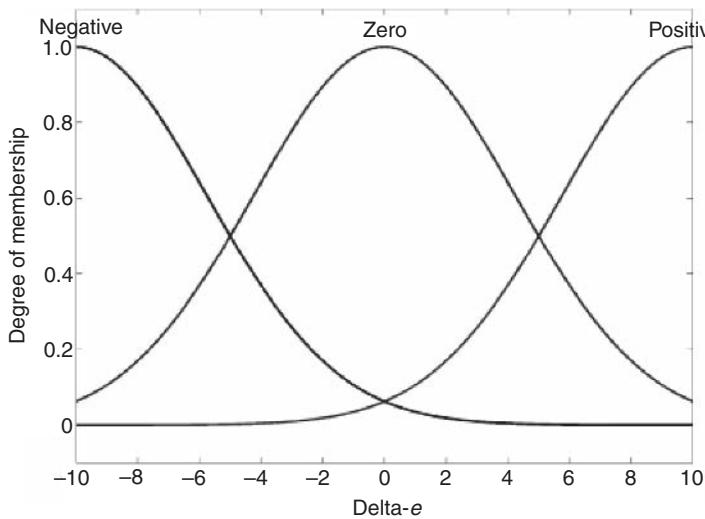


Figure 26.11 Input Δe membership functions for the type-1 FLC

deviation is 2.1233 as can be seen in Figure 26.12. Table 26.1 illustrates the characteristics of the MFs of the inputs and output of the type-1 FLC.

In experiments 2, 4, and 5, for the type-2 FLC, as in type-1 FLC, we also selected Gaussian MFs for the inputs and for the output, but in this case we have interval type-2 Gaussian MFs with a fixed center, c , and an uncertain standard deviation, σ ; i.e.,

$$\mu_A(x) = e^{-\frac{1}{2}(\frac{x-c}{\sigma})^2}. \quad (29)$$

In terms of the upper and lower membership functions, we have for $\bar{\mu}_{\tilde{A}}(x)$,

$$\bar{\mu}_{\tilde{A}}(x) = N(c, \sigma_2; x) \quad (30)$$

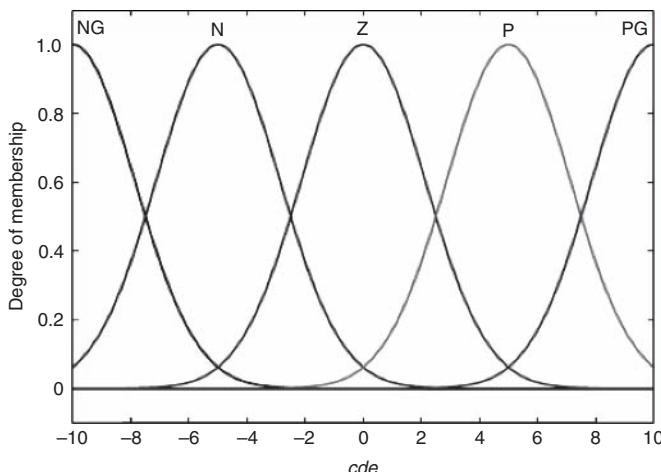


Figure 26.12 Output cde membership functions for the type-1 FLC

Table 26.1 Characteristics of the inputs and outputs of the type-1 FLC

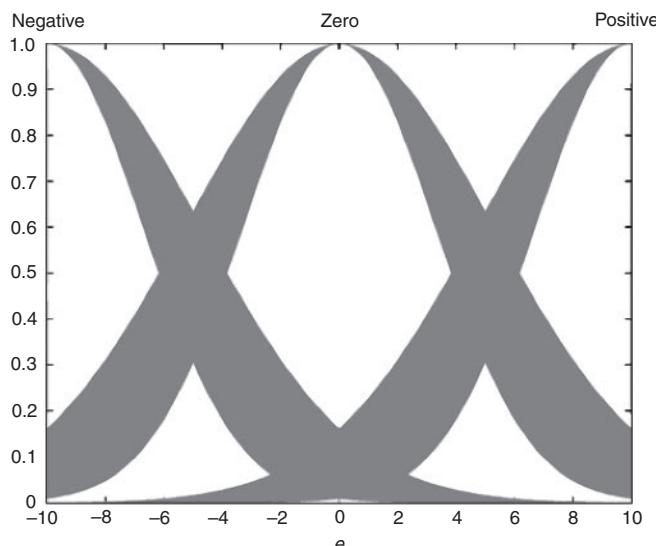
Variable	Term	Center c	Standard deviation σ
Input e	Negative	-10	4.2466
	Zero	0	4.2466
	Positive	10	4.2466
Input Δe	Negative	-10	4.2466
	Zero	0	4.2466
	Positive	10	4.2466
Output cde	NG	-10	2.1233
	N	-5	2.1233
	Z	0	2.1233
	P	5	2.1233
	PG	10	2.1233

and for the lower membership function $\underline{\mu}_{\tilde{A}}(x)$,

$$\underline{\mu}_{\tilde{A}}(x) = N(c, \sigma_1, x), \quad (31)$$

where $N(c, \sigma_2, x) \equiv e^{-\frac{1}{2}(\frac{x-c}{\sigma_2})^2}$ and $N(c, \sigma_1, x) \equiv e^{-\frac{1}{2}(\frac{x-c}{\sigma_1})^2}$ [5]. Hence, in the type-2 FLC, for each input we defined three-interval type-2 fuzzy Gaussian MFs: negative, zero, and positive in the interval $[-10, 10]$, as illustrated in Figures 26.13 and 26.14. For computing the output we have five-interval type-2 fuzzy Gaussian MFs, which are NG, N, Z, P, and PG, in the interval $[-10, 10]$, as can be seen in Figure 26.15. Table 26.2 shows the characteristics of the inputs and output of the type-2 FLC.

For type-2 FLC we used, basically, the software is available online for type-2 fuzzy logic [40]. In all experiments, we have a dash-dot line for illustrating the system's response and behavior of type-1 FLC, in the same sense, a continuous line for type-2 FLC. The reference input r is shown with a dot line.

**Figure 26.13** Input e membership functions for the type-2 FLC

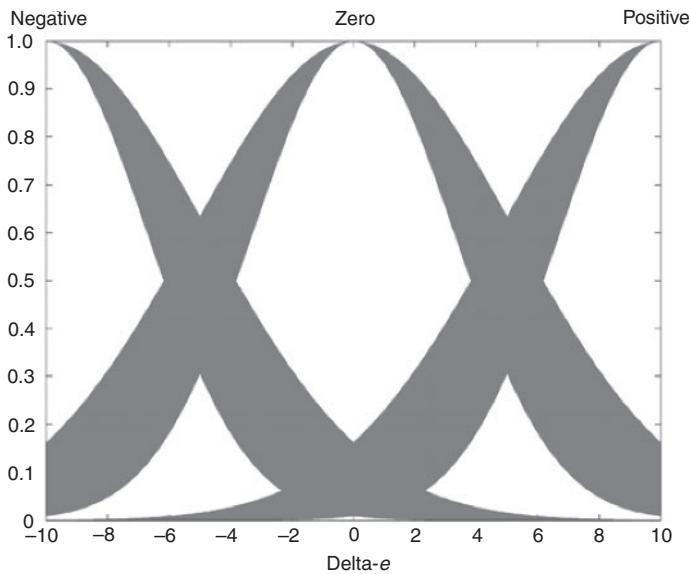


Figure 26.14 Input Δe membership functions for the type-2 FLC

Experiment 1. Simulation of an ideal system with a type-1 FLC.

In this experiment, uncertainty data were not added to the system and the system's response is illustrated in Figure 26.16. Note that the settling time is of about 140 units of time; i.e., the system tends to stabilize with time and the output will follow accurately the input. In Table 26.3, we listed the obtained values of ISE, IAE, and ITAE for this experiment. In Figures 26.17, 26.18, and 26.19, the ISE, IAE, and ITAE behaviors of this experiment are shown.

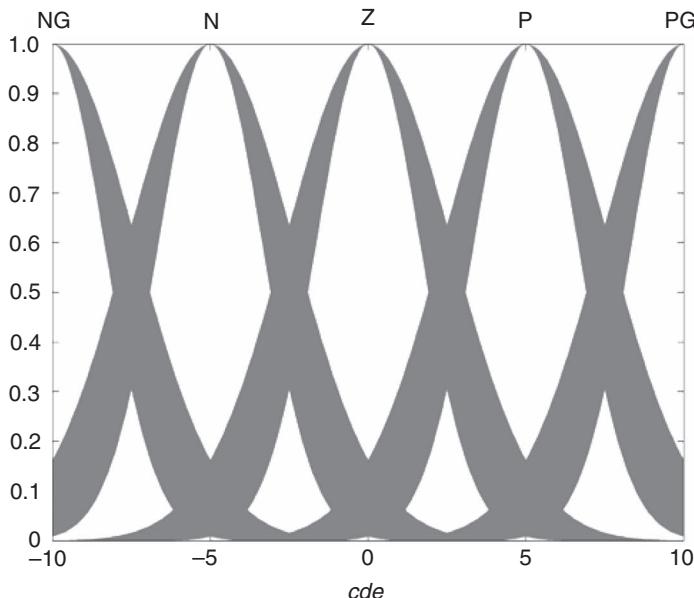
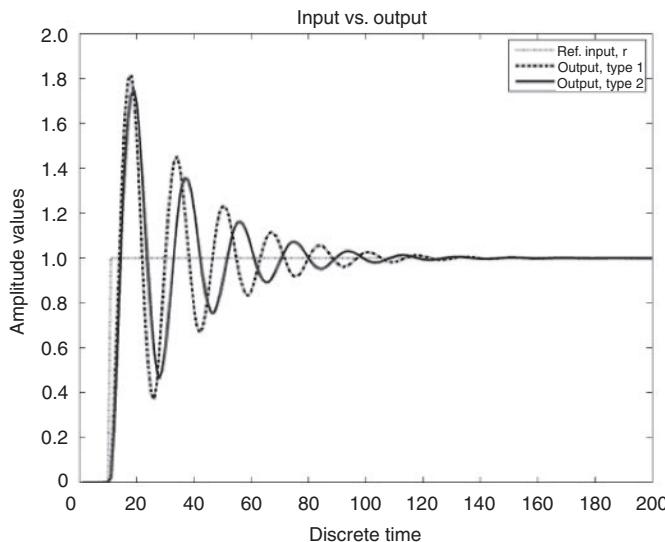


Figure 26.15 Output cde membership functions for the type-2 FLC

Table 26.2 Characteristics of the inputs and output of the type-2 FLC

Variable	Term	Center c	Standard deviation σ_1	Standard deviation σ_2
Input e	Negative	-10	5.2466	3.2466
	Zero	0	5.2466	3.2466
	Positive	10	5.2466	3.2466
Input Δe	Negative	-10	5.2466	3.2466
	Zero	0	5.2466	3.2466
	Positive	10	5.2466	3.2466
Output cde	NG	-10	2.6233	1.6233
	N	-5	2.6233	1.6233
	Z	0	2.6233	1.6233
	P	5	2.6233	1.6233
	PG	10	2.6233	1.6233

**Figure 26.16** This graphic shows the system's response to a unit step sequence. The input reference r is shown with pointed line, for the type-1 the systems' output $y(i)$ is shown with dash-dot line, and for type-2, the system's output $y(i)$ with continuous line**Table 26.3** Comparison of performance criteria for type-1 and type-2 fuzzy logic controllers for 20-db signal-to-noise ratio^a

Performance criteria	Type-1 FLC		Type-2 FLC	
	Ideal system	System with uncertainty	Ideal system	System with uncertainty
ISE	7.65	19.4	6.8	18.3
IAE	17.68	49.5	16.4	44.8
ITAE	62.46	444.2	56.39	402.9

^aValues obtained after 200 samples.

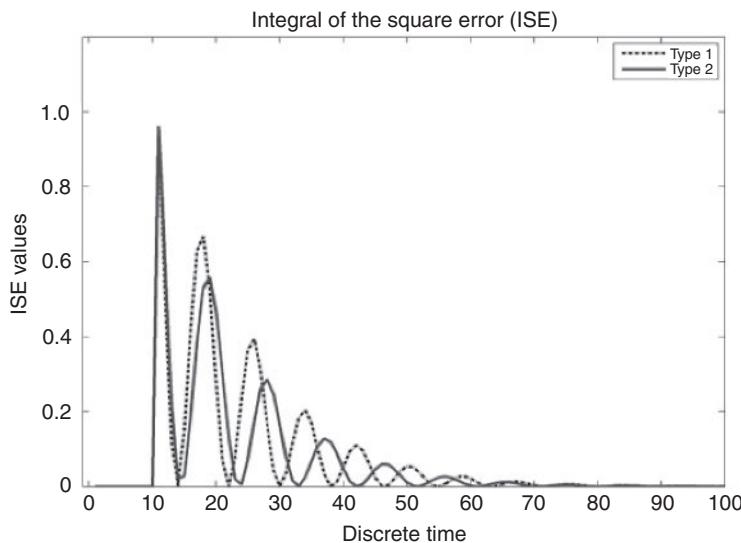


Figure 26.17 In uncertainty absence, the ISE values are very similar for type-1 and type-2 FLCs

Experiment 2. Simulation of an ideal system using the type-2 FLC.

Here, the same test conditions of experiment 1 were used, but in this case, we implemented the controller's algorithm with type-2 fuzzy logic. The output sequence is illustrated in Figure 26.16, and the corresponding performance criteria are listed in Table 26.3, and we can observe that using a type-2 FLC we obtained the lower errors. By visual inspection, we can observe that the output system's response of the experiment 1, and this one, are similar as it is shown in Figures 26.17, 26.18, and 26.19.

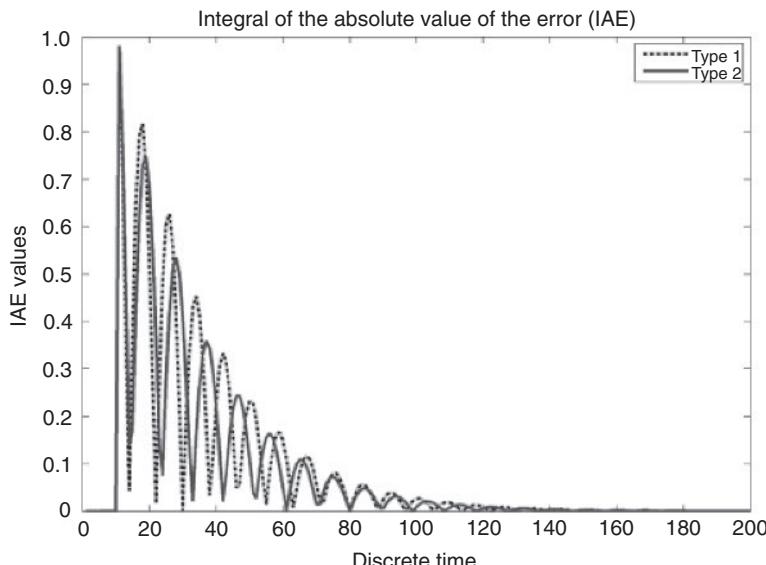


Figure 26.18 In uncertainty absence, the IAE values obtained at the plant's output are very similar for type-1 and type-2 FLCs here it is more evident that a type-1 FLC works a little better than in Figure 26.17

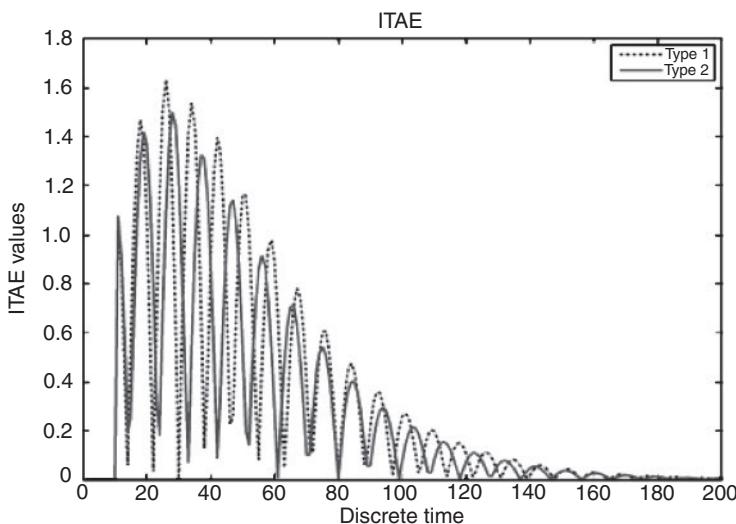


Figure 26.19 In uncertainty absence, the ITAE values obtained at the plant's output are similar for type-1 and type-2 FLCs, in accordance with Figure 26.18, it is evident that a type-1 FLC works a little better

Experiment 3. System with uncertainty using a type-1 FLC.

In this case, equation (25) was used to simulate the effects of uncertainty introduced to the system by transducers, amplifiers, and any other element that in real-world applications affects expected values. In this experiment the noise level was simulated in the range of 20 db of SNR ratio. Figure 26.20 shows the system's response output. In Figures 26.21, 26.22, and 26.23, the performance criteria ISE, IAE, and ITAE are represented graphically.

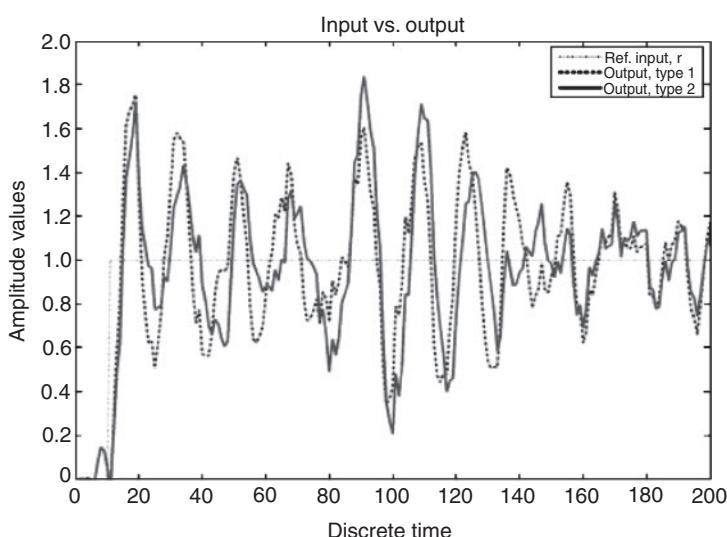


Figure 26.20 This graphic was obtained with uncertainty presence; compare the system's outputs produced by type-1 and type-2 FLCs. Note that quite the opposite to Figure 26.16, a type-2 FLC works much better than a type-1 FLC when the system has uncertainty. The overshoot error is lower for a type-2 FLC

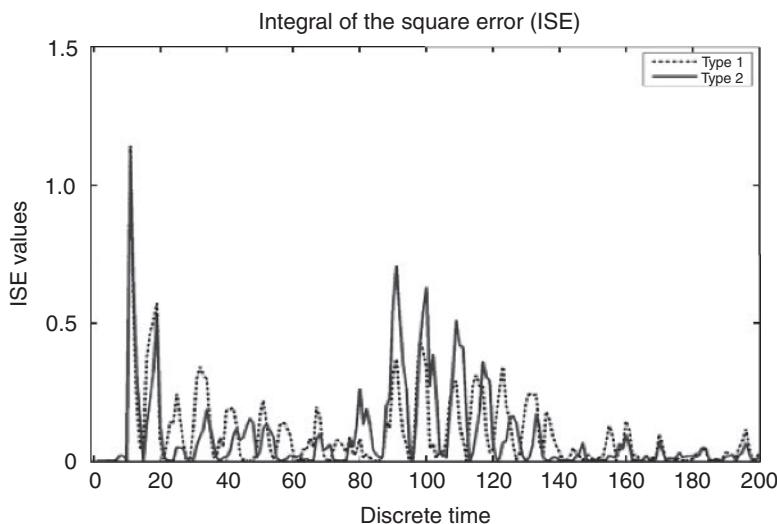


Figure 26.21 We can see that a type-2 FLC produces lower overshoot errors; quantitatively the ISE overall error of using type-2 is 18.3 against 19.4 of the overall error produced by the type-1 FLC

Experiment 4. System with uncertainty using a type-2 FLC.

In this experiment, uncertainty was introduced in the system, in the same way as in experiment 3. In this case, a type-2 FLC was used and the results obtained with a type-1 FLC (experiment 3) were improved. We can appreciate from Figure 26.20 that the lower overshoot and the best settling times are reached using a type-2 FLC. In Figures 26.21 and 26.22, we can see that with a type-2 FLC the overshoot error decreases very quickly and it remains lower than using a type-1 FLC. In Figure 26.23, we can observe that through time the lower errors are obtained using a type-2 FLC.

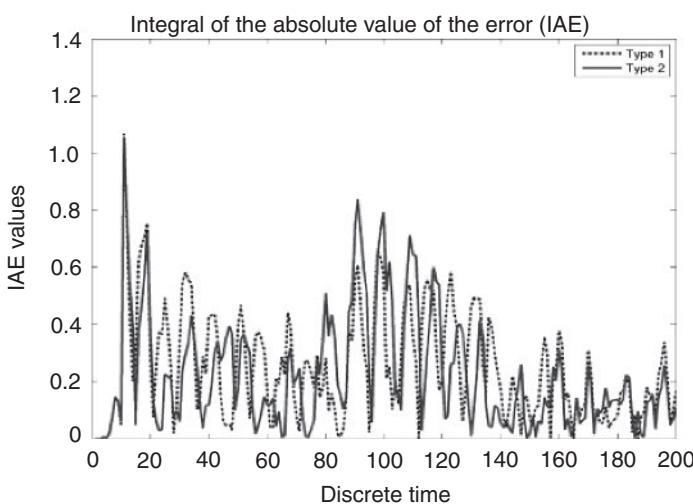


Figure 26.22 In accordance with Figure 26.20, IAE confirms that we obtained the best system response using a type-2 FLC with uncertainty presence. Moreover, the error of the settling time and steady state is lower using a type-2 FLC

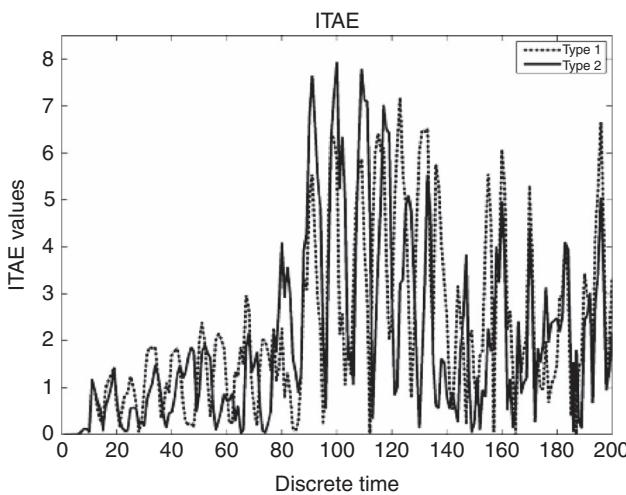


Figure 26.23 Here we can see that the steady-state error of the system produced by a type-2 FLC is lower than the error produced by a type-1 FLC with uncertainty present. ITAE will punish heavily all those errors produced with time

Experiment 5. Varying the SNR in type-1 and type-2 FLCs.

To test the robustness of the type-1 and type-2 FLCs, we repeated experiments 3 and 4, giving different noise levels, going from 30 db to 8 db of SNR ratio in each experiment. In Table 26.4, we summarized the values for ISE, IAE, and ITAE considering 200 units of time with a P_{signal} of 22.98 db in all cases. As it can be seen in Table 26.4, in presence of different noise levels, the behavior of type-2 FLC is in general better than type-1 FLC.

From Table 26.4, considering two examples, the extreme cases, we have for an SNR ratio of 8 db, in type-1 FLC the following performance values ISE = 321.1, IAE = 198.1, ITAE = 2234.1; and for

Table 26.4 Behavior of type-1 and type-2 fuzzy logic controllers after variation of signal noise ratio^a

SNR (db)	SNR	Noise variation		Type-1 FLC			Type-2 FLC		
		SumNoise	SumNoise (db)	ISE	IAE	ITAE	ISE	IAE	ITAE
8	6.4	187.42	22.72	321.1	198.1	2234.1	299.4	194.1	2023.1
10	10.058	119.2	20.762	178.1	148.4	1599.4	168.7	142.2	1413.5
12	15.868	75.56	18.783	104.7	114.5	1193.8	102.1	108.8	1057.7
14	25.135	47.702	16.785	64.1	90.5	915.5	63.7	84.8	814.6
16	39.883	30.062	14.78	40.9	72.8	710.9	40.6	67.3	637.8
18	63.21	18.967	12.78	27.4	59.6	559.1	26.6	54.2	504.4
20	100.04	11.984	10.78	19.4	49.5	444.2	18.3	44.8	402.9
22	158.54	7.56	8.78	14.7	42	356.9	13.2	37.8	324.6
24	251.3	4.77	6.78	11.9	36.2	289	10.3	32.5	264.2
26	398.2	3.01	4.78	10.1	31.9	236.7	8.5	28.6	217.3
28	631.5	1.89	2.78	9.1	28.5	196.3	7.5	25.5	180.7
30	1008	1.19	0.78	8.5	25.9	164.9	7	23.3	152.6

^aValues obtained for 200 samples.

the same case, in type-2 FLC, we have ISE = 299.4, IAE = 194.1, ITAE = 2023.1. For 30 db of SNR ratio, we have for the type-1 FLC, ISE = 8.5, IAE = 25.9, ITAE = 164.9, and for the type-2 FLC, ISE = 7, IAE = 23.3, ITAE = 152.6. These values indicate a better performance of the type-2 FLC than type-1 FLC, because they are a representation of errors, and as the error increases the performance of the system goes down.

26.4 Experimental Results for Time-Series Prediction

In this section, we illustrate the application of interval type-2 fuzzy logic to the problem of time-series prediction [13, 41, 42]. The Mackey–Glass time series [30] is used to compare the results of interval type-2 fuzzy logic with the results of other intelligent methods. In particular, a comparison is made with type-1 fuzzy systems, neural networks, neuro-fuzzy systems, and neuro-genetic and fuzzy-genetic approaches.

26.4.1 Prediction with a Neural Network

In this case, we did find a neural network model for this time series using five time delays and six periods. The architecture of the neural network used was 4-12-1 with 150 epochs, and 500 data points for training/500 data points for validation. The mean squared error for prediction is 0.0043 (see Figure 26.24).

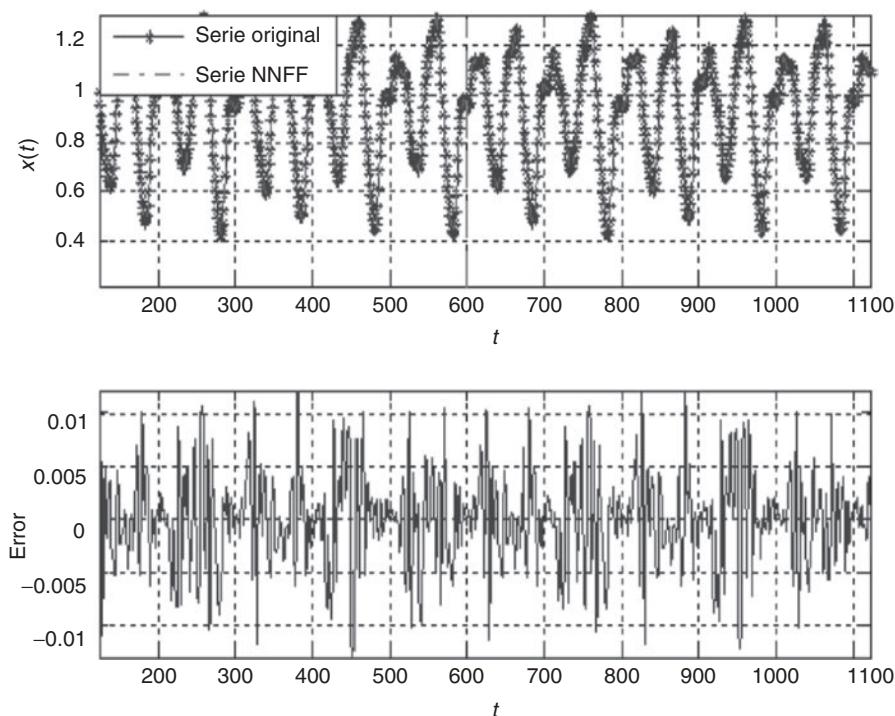


Figure 26.24 Forecasting the Mackey–Glass time series with a neural network. On top it is shown the comparison of the prediction with the neural network and the original time series. Below it is shown the forecasting error

26.4.2 Prediction with an Adaptive Neuro-Fuzzy Inference System

To find the adaptive neuro-fuzzy inference system (ANFIS) model an analysis was made of the data, and the decision was to use a five time delay with six periods. The time series was divided into 500 data points for training and 500 data points for validation. The ANFIS model was designed with four inputs (two membership functions each) and one output with 16 linear functions, giving a total of 16 fuzzy rules. The training was of 50 epochs and the mean squared error of prediction is of 0.0016 (see Figure 26.25).

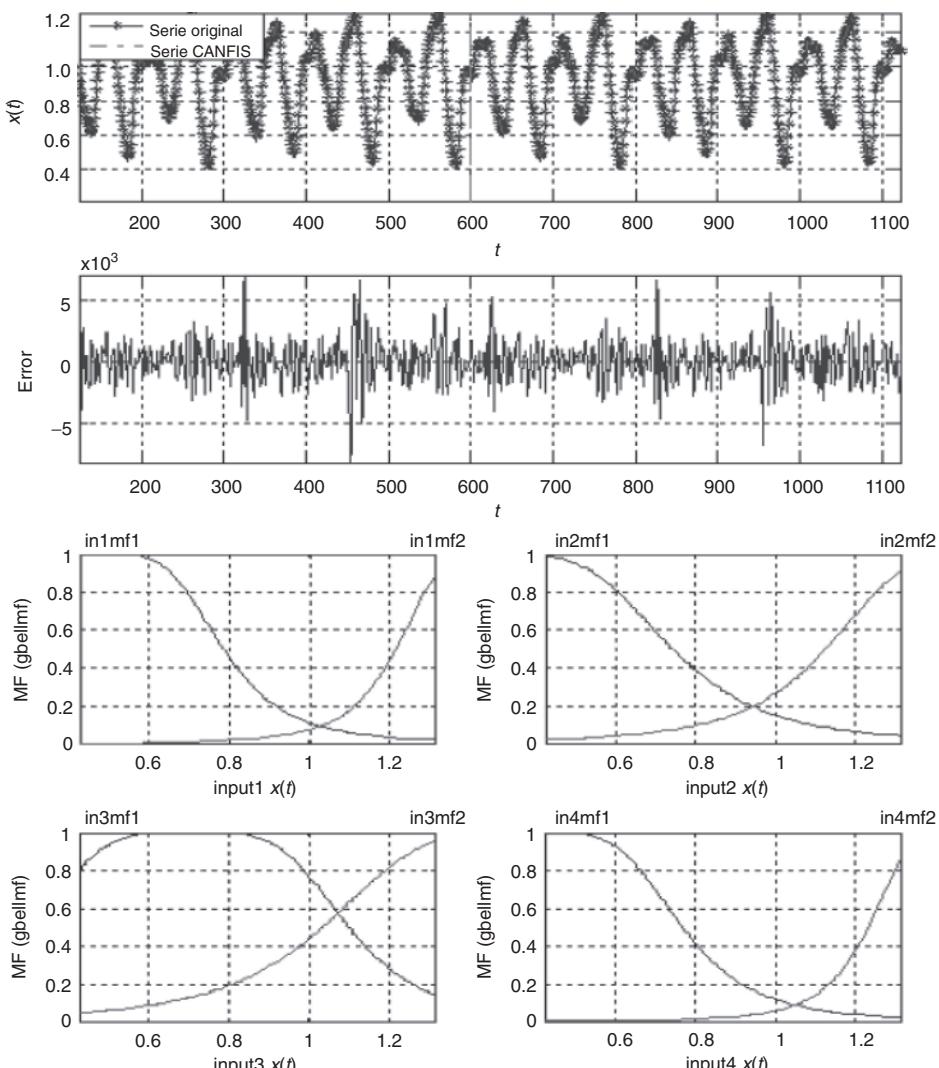


Figure 26.25 Forecasting the Mackey–Glass time series with ANFIS. The figure on top shows the comparison of the predicted values and the original time series. The following figures show the error plot and the membership functions obtained with ANFIS

26.4.3 Prediction with an Interval Type-2 TSK Fuzzy Model

To find the type-2 fuzzy model an analysis was made of the data, and the decision was to use a five time delay with six periods. The time series was divided into 500 data points for training and 500 data points for validation. The model was designed with four inputs (with two interval type-2 (igbellmtype2) membership functions each) and one output with 16 interval linear functions, giving a total of 16 fuzzy rules. The training was of 50 epochs and the mean squared error of prediction is of 0.00023 (see Figure 26.26).

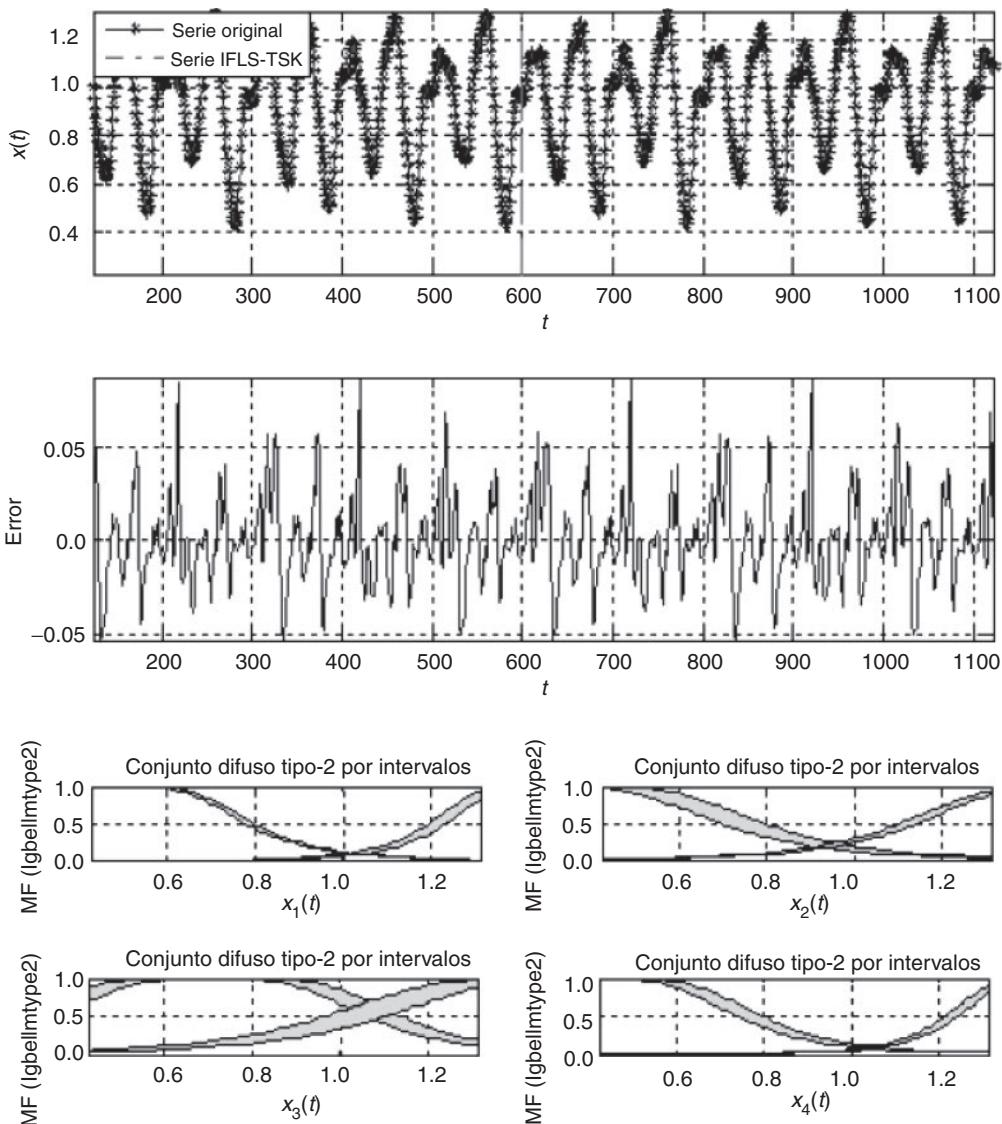


Figure 26.26 Forecasting the Mackey–Glass time series with an interval type-2 TSK fuzzy model. The figure on top shows the comparison of the predicted values and the original time series. The following figures show the error plot and the interval type-2 membership functions obtained

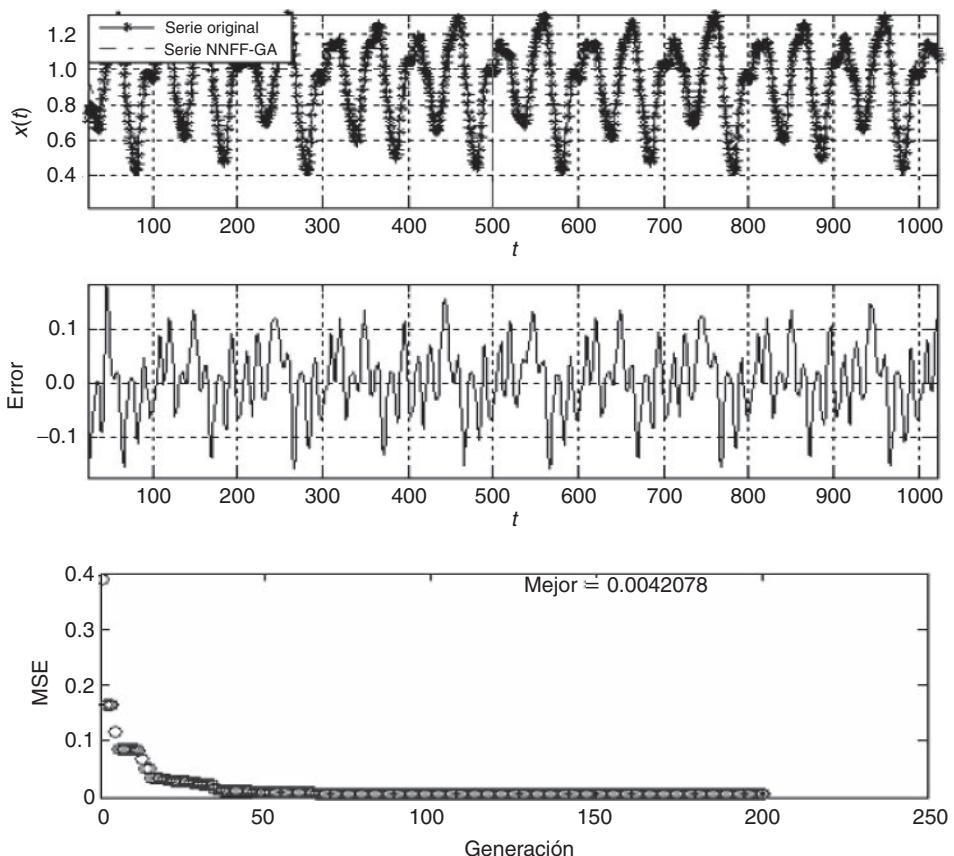


Figure 26.27 Forecasting the Mackey–Glass time series with a neuro-genetic model. The figure on top shows the comparison of the predicted values and the original time series. The figure on the middle shows the error plot. The figure on the bottom shows the evolution of the mean squared error as the generations increase up to 250

26.4.4 Prediction with a Neuro-Genetic Model

In this case, we use a genetic algorithm for training a neural network model for this time series using five time delays and six periods. The architecture of the neural network used was 4-12-1 with 200 generations, and 500 data points for training/500 data points for validation. The mean squared error for prediction is 0.00640 (see Figure 26.27). The genetic parameters are population size = 30, crossover ratio = 0.75, and mutation ratio = 0.01.

26.4.5 Prediction with Type-1 Fuzzy Models Optimized with Genetic Algorithms

In this section, we show prediction results of type-1 fuzzy models that were optimized using genetic algorithms. In all cases, we have four inputs (with two membership functions) and one output with 16 functions. The parameters of the genetic algorithms are the same as in the previous section. We show in Figure 26.28 the results of a TSK model (mean squared error of 0.0647) and in Figure 26.29 the results of a Mamdani model (mean squared error of 0.0692).

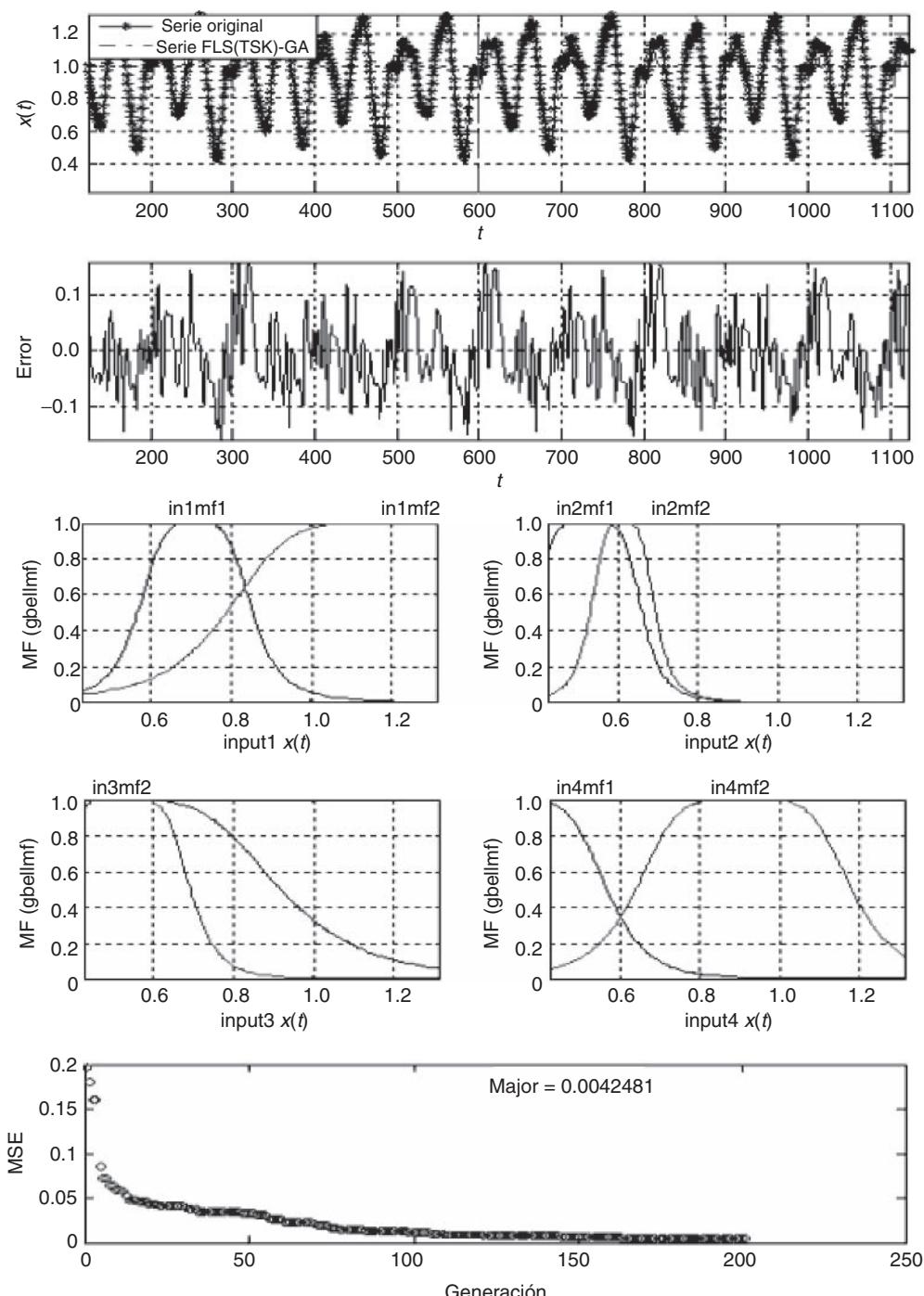


Figure 26.28 Forecasting the Mackey-Glass time series with a TSK fuzzy model optimized using a genetic algorithm

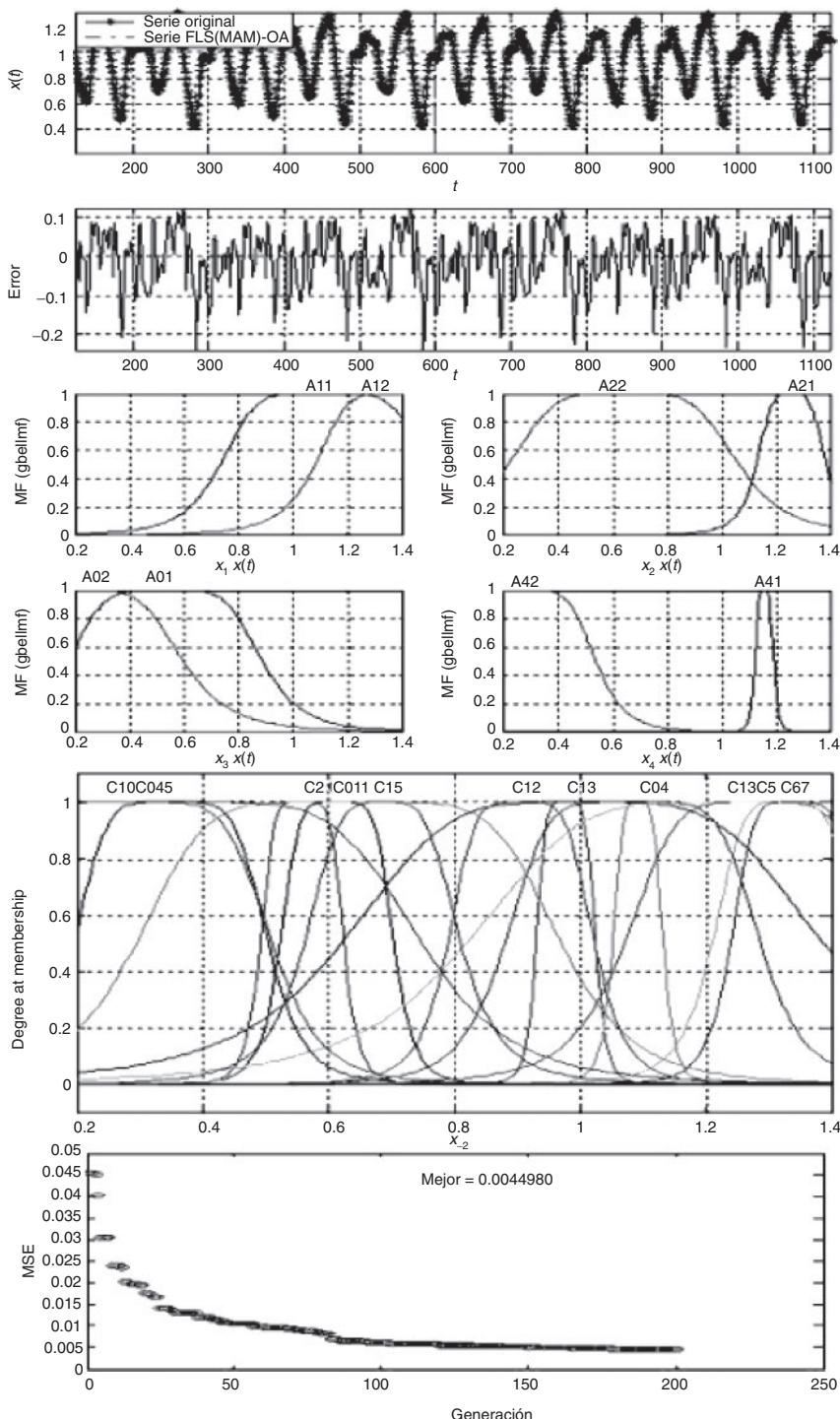


Figure 26.29 Forecasting the Mackey-Glass time series with a Mamdani fuzzy model optimized using a genetic algorithm

Table 26.5 Summary of results for the Mackey–Glass Time series

Method	RMSE	Data training/checking	Epochs or generations
NNFF (Figure 26.24)	0.0043	500/500	150
CANFIS (Figure 26.25)	0.0016	500/500	50
IT2FLS(TSK) (Figure 26.26)	0.00023	500/500	1
NNFF-GA (Figure 26.27)	0.0640	500/500	150
FLS(TSK)-GA (Figure 26.28)	0.0647	500/500	200
FLS(MAM)-GA (Figure 26.29)	0.0693	500/500	200

The Mackey–Glass time series shows chaotic behavior and for this reason has been chosen many times as a benchmark problem for prediction methods. We show in Table 26.5 a summary of the results using the methods mentioned previously. Based on mean squared error (RMSE) of forecasting, we can conclude that the interval type-2 fuzzy model (IT2 FLS on Table 26.5) is the best one to predict future values of this time series. Also, based on the training required, we can conclude that the interval type-2 fuzzy model is the best one because it requires only one epoch.

26.5 Conclusion

We have presented the study of the controllers' design for non-linear control systems using type-1 and type-2 fuzzy logic. We presented five experiments where we simulated the systems' responses with and without uncertainty presence. In the experiments, a quantification of errors was achieved and documented in tables for different criteria such as ISE, IAE, and ITAE; it was shown that the lower overshoot errors and the best settling times were obtained using a type-2 FLC. Based on the experimental results, we can say that the best results are obtained using type-2 fuzzy systems. In our opinion, this is because type-2 fuzzy sets that are used in type-2 fuzzy systems can handle uncertainties in a better way because they provide us with more parameters and more design degrees of freedom. We also presented simulation results of time-series forecasting in which the interval type-2 fuzzy model outperforms other intelligent methods for the same application.

References

- [1] G.J. Klir, B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, NJ, 1995.
- [2] L.A. Zadeh. Toward a generalized theory of uncertainty (GTU) – an outline. *Inf. Sci.* 172 (2005) 1–40.
- [3] L. Zadeh. The concept of a linguistic variable and its application to approximate reasoning, Part 1. *Inf. Sci.* 8 (3) (1975) 199–249.
- [4] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning, Part II. *Inf. Sci.* 8 (4) 301–357.
- [5] J.M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall, NJ, 2001.
- [6] M. Mizumoto and K. Tanaka. Some properties of fuzzy sets of type-2. *Inf. Control* 31 (1976) 312–340.
- [7] N.N. Karnik and J.M. Mendel. Operations on type-2 fuzzy sets. *Int. J. Fuzzy Sets Syst.* 122 (2) (2001) 327–348.
- [8] M. Wagenknecht and K. Hartmann. Application of fuzzy sets of type 2 to the solution of fuzzy equations systems. *Fuzzy Sets Syst.* 25 (1988) 183–190.
- [9] R.R. Yager. Fuzzy subsets of type II in decisions. *J. Cybern.* 10 (1980) 137–159.
- [10] O. Castillo and P. Melin. *Soft Computing for Control of Non-Linear Dynamical Systems*. Springer-Verlag, Heidelberg, Germany, 2001.
- [11] O. Castillo and P. Melin. *Soft Computing and Fractal Theory for Intelligent Manufacturing*. Springer-Verlag, Heidelberg, Germany, 2003.
- [12] O. Castillo and P. Melin. A new approach for plant monitoring using type-2 fuzzy logic and fractal theory. *Int. J. Gen. Syst.* 33 (2) (2004) 305–319.

- [13] A. Mencattini, M. Salmeri, S. Bertazzoni, R. Lojacono, E. Pasero, and W. Monaci. Local meteorological forecasting by type-2 fuzzy systems time series prediction. In: *Computational Intelligence for Measurement Systems and Applications, Cimsa. 2005*, July, 20–22, pp. 75–80.
- [14] F. Doctor, H. Hagras, and V. Callaghan. A type-2 fuzzy embedded agent to realize ambient intelligence in ubiquitous computing environments. *Inf. Sci.* 171 (2005) 309–334.
- [15] N.N. Karnik, J.M. Mendel, and Q. Liang. Type-2 fuzzy logic systems. *IEEE Trans. Fuzzy Syst.* 7 (6) (1999) 643–658.
- [16] J.M. Mendel. Type-2 fuzzy logic systems: Type-reduction. In: *IEEE System, Man, and Cybernetics Conference*. San Diego, CA, 1998, pp. 2046–2051.
- [17] J.M. Mendel and R.I. John. Type-2 fuzzy sets made simple. *IEEE Trans. Fuzzy Syst.* 10 (2) (2002) 117–127.
- [18] H. Wu and J.M. Mendel. Introduction to uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems, In: *Proceeding of FUZZ-IEEE 2001*, Melbourne, Australia, 2001, pp. 662–665.
- [19] J.M. Mendel. Computing derivatives in interval type-2 fuzzy logic systems. *IEEE Trans. Fuzzy Syst.* 12 (2004) 84–98.
- [20] N.N. Karnik and J.M. Mendel. Applications of type-2 fuzzy logic systems to forecasting of time-series. *Inf. Sci.* 120 (1–4) (1999) 89–111.
- [21] J.M. Mendel. Uncertainty, fuzzy logic, and signal processing. *Signal Process. J.* 80 (2000) 913–933.
- [22] T. Ozen and J.M. Garibaldi. Investigating adaptation in type-2 fuzzy logic systems applied to umbilical acid-base assessment. In: *Proceeding of the European Symposium on Intelligent Technologies, Hybrid Systems and Their Implementation on Smart Adaptive Systems (EUNITE 2003)*, Oulu, Finland, 2003.
- [23] P. Melin and O. Castillo. Intelligent control of non-linear dynamic plants using type-2 fuzzy logic and neural networks. In: *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society*. New Orleans, LA, 2002, pp. 22–27.
- [24] P. Melin and O. Castillo. A new method for adaptive model-based control of non-linear dynamic plants using type-2 fuzzy logic and neural networks. In: *Proceeding of the 12th IEEE International Conference on Fuzzy Systems*. St. Louis, MO, 2003, pp. 420–425.
- [25] P. Melin and O. Castillo. A new method for adaptive control of non-linear plants using type-2 fuzzy logic and neural networks. *Int. J. Gen. Syst.* 33 (2) (2004) 289–304.
- [26] L. Astudillo, O. Castillo, and L.T. Aguilar. Intelligent control of an autonomous mobile robot using type-2 fuzzy logic, In: *Proceeding of the Conference on Artificial Intelligence, ICAI'06*, Las Vegas, NV, 2006, pp. 565–570.
- [27] H.A. Hagras. Hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Trans. Fuzzy Syst.* 12 (4) (2004) 524–539.
- [28] R. Sepúlveda, O. Castillo, P. Melin, A. Rodríguez-Díaz, and O. Montiel. Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic. *Inf. Sci.* 177 (10) (2007) 2023–2048.
- [29] G.E.P. Box and G.M. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden Day, San Francisco, 1976.
- [30] J.-S.R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence, Matlab Curriculum Series*. Prentice Hall, NJ, 1997.
- [31] E.H. Mamdani. Twenty years of fuzzy control: Experiences gained and lessons learned. In: *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*. San Francisco, CA, 1993, pp. 339–344.
- [32] L.A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Syst. Man Cybern.* 3 (1973) 28–44.
- [33] L.A. Zadeh. Similarity relations and fuzzy ordering. *Inf. Sci.* 3 (1971) 177–206.
- [34] J.M. Mendel and G.C. Mouzouris. Type-2 fuzzy logic systems, *IEEE Trans. Fuzzy Syst.* 7 (1999) 643–658.
- [35] J.M. Mendel and H. Wu. Properties of the centroid of an interval type-2 fuzzy set, including the centroid of a fuzzy granule. In: *Proceeding of FUZZ-IEEE 2005*, Reno, NV, 2005, pp. 341–346.
- [36] J.M. Mendel. On a 50% savings in the computation of the centroid of a symmetrical interval type-2 fuzzy set. *Inf. Sci.* 172 (3–4) (2005) 417–430.
- [37] N.N. Karnik and J.M. Mendel. Centroid of a type-2 fuzzy set. *Inf. Sci.* 132 (1–4) (2001) 195–220.
- [38] W.C. Messner and D.M. Tilbury. *Control Tutorials for MatLab and Simulink, a Web Based Approach*. Prentice Hall, NJ, 1999.
- [39] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing Principles, Algorithms, and Applications*, 3rd ed. Prentice Hall, NJ, 1996.
- [40] N.N. Karnik, Q. Liang, and J.M. Mendel. Type-2 fuzzy logic software. Available at: <http://sipi.usc.edu/mendel/software/>, 2001.
- [41] K. Huaring and H. Yu. A Type 2 fuzzy time series model for stock index forecasting. *Phys. A Stat. Mech. Appl.* 353 (2005) 445–462.
- [42] N.N. Karnik and J.M. Mendel. Applications of type-2 fuzzy logic systems to forecasting of time-series. *Inf. Sci.* 120 (1–4) (1999) 89–111.

27

Theoretical Aspects of Shadowed Sets

Gianpiero Cattaneo and Davide Ciucci

27.1 Introduction

A well-known, and deeply studied, generalization of classical (Boolean) sets are fuzzy sets. First introduced by L. Zadeh in 1965 [1], fuzzy sets are formally defined as follows.

Definition 1. Let X be a set of objects, called the universe of discourse. A fuzzy set on X is any mapping $f : X \rightarrow [0, 1]$.

In the sequel, we denote the collection of all fuzzy sets on X by $[0, 1]^X$ or sometimes simply by $\mathcal{F}(X)$, and by \mathbf{k} , for any fixed $k \in [0, 1]$, the constant fuzzy set $\forall x \in X, \mathbf{k}(x) = k$. Moreover, let us recall that on $\mathcal{F}(X)$, it is possible to define a *pointwise order relation*. For any two fuzzy sets $f, g \in \mathcal{F}(X)$,

$$f \leq g \quad \text{iff} \quad \forall x \in X \ f(x) \leq g(x).$$

This order induces on $\mathcal{F}(X)$ a lattice structure, where the meet and join operators are, respectively, defined as

$$\begin{aligned} (f \wedge g)(x) &:= \min\{f(x), g(x)\}, \\ (f \vee g)(x) &:= \max\{f(x), g(x)\}. \end{aligned}$$

This lattice is complete and distributive, bounded by the minimum element $\mathbf{0}$ and by the maximum element $\mathbf{1}$.

Fuzzy sets are a powerful tool to describe and reason with vague concepts; indeed, they permit to express vagueness with an infinite degree of accuracy, i.e., any value in the range $[0, 1]$. However, this quality in precision may not always be needed or even worst may lead to high computational costs. Thus, some ‘simplifications’ of fuzzy sets have been proposed, for instance, level fuzzy sets and α -cuts, which will be briefly discussed in Section 27.2. More recently, Pedrycz introduced shadowed sets [2–4] as a new tool to deal with vagueness in a simpler, and in his opinion more realistic, way. According to him, the intention was ‘to introduce a model which does not lend itself to precise numerical membership values but relies on basic concepts of truth values (yes–no) and on entire [open] unit interval perceived as a zone of ‘uncertainty’ [2]. Formally, a shadowed set can be defined as follows.

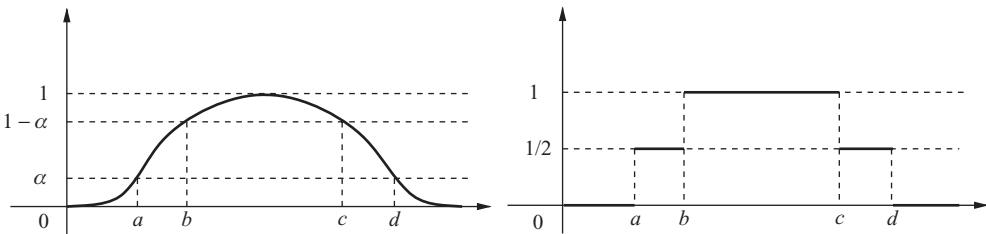


Figure 27.1 A fuzzy set and its corresponding shadowed set

Definition 2. Let X be a set of objects, called the universe. A shadowed set on X is any mapping $s : X \rightarrow \{0, 1, (0, 1)\}$.

We denote the collection of all shadowed sets on X as $\{0, 1, (0, 1)\}^X$, or sometimes simply by $S(X)$. Further, in order to simplify the notation, we will indicate $(0, 1)$ by the value $\frac{1}{2}$. This is only a change from a syntactical point of view, without any loss from the semantic point of view. Indeed, if 1 corresponds to truth, 0 to falseness, then $\frac{1}{2}$ is halfway between true and false; i.e., it represents a really uncertain situation, as it was with the original syntax of $(0, 1)$. Given a shadowed set $s \in S(X)$, its *shadow* is the collection of all the points x from the universe for which $s(x) \neq 0, 1$.

Fuzzy and shadowed sets are strictly linked since from any fuzzy set it is possible to obtain a corresponding shadowed set. In order to define such a mapping, it is sufficient to fix a value $\alpha \in [0, \frac{1}{2}]$. Then, for a given fuzzy set f , the membership values $f(x)$ which are less than or equal to α are set to 0 and those greater than or equal to $(1 - \alpha)$ are set to 1. The remaining ones, i.e., the membership values belonging to $(\alpha, 1 - \alpha)$, are set to $\frac{1}{2}$ (in the original approach to $(0, 1)$), since they are characterized by a great uncertainty or lack of knowledge and they are consequently considered the ‘shadow’ of the induced shadowed set. In Figure 27.1, a fuzzy set and its induced shadowed set are represented.

More formally, let $\alpha \in (0, \frac{1}{2})$ be a fixed value, the α -approximation function (also, α -shadow representation) of a fuzzy set f , denoted by $s_\alpha(f)$, is defined as the following shadowed set:

$$s_\alpha(f)(x) := \begin{cases} 0 & f(x) \leq \alpha \\ 1 & f(x) \geq 1 - \alpha \\ \frac{1}{2} & \text{otherwise.} \end{cases} \quad (1)$$

In this chapter, shadowed sets are compared with other methods of granular computing. In Section 27.2, we analyze the relationship among shadowed sets and other ways of approximating fuzzy sets, intuitionistic fuzzy sets, and variable-precision rough sets. Then, in Section 27.3, an algebraic approach to shadowed sets, as well as to the other introduced paradigms, is given. To this scope BZMV algebra, pre-BZMV algebras, and ŁΠ algebras are introduced.

27.2 Shadowed Sets in the Context of Granular Computing

In this section, we analyze the relationship among shadowed sets and other tools of knowledge granulation. As a first step, we introduce a rough approximation of a fuzzy set and see how it is related to a shadowed set generated from the same fuzzy set.

27.2.1 Relationship with Fuzzy Sets and Their Rough Approximations

Let $f \in \mathcal{F}(X)$ be a given fuzzy set on the domain X . It is possible to introduce the two operators of *necessity* $v(f)$ and *possibility* $\mu(f)$ of f (the use of these names will be clear in Section 27.3) defined

for any point $x \in X$ respectively as

$$\nu(f)(x) = \begin{cases} 1 & f(x) = 1 \\ 0 & f(x) \neq 1, \end{cases} \quad (2a)$$

$$\mu(f)(x) = \begin{cases} 0 & f(x) = 0 \\ 1 & f(x) \neq 0. \end{cases} \quad (2b)$$

For any fuzzy set $f \in \mathcal{F}(X)$, its *necessity domain* $A_1(f)$ and *possibility domain* $A_p(f)$ are the two subsets of the universe defined as

$$\begin{aligned} A_1(f) &:= \{x \in X : f(x) = 1\}, \\ A_p(f) &:= \{x \in X : f(x) \neq 0\}. \end{aligned}$$

For any subset A of the universe X , let us denote by χ_A the *characteristic functional* of A defined as $\chi_A(x) = 1$ if $x \in A$, and equal to 0 otherwise. Using the above-introduced domains of a fuzzy set f , one immediately obtains that

$$\nu(f) = \chi_{A_1(f)} \quad \text{and} \quad \mu(f) = \chi_{A_p(f)}.$$

These two operators can be used to define a *rough approximation* of a fuzzy set as

$$r(f) := \langle \nu(f), \mu(f) \rangle \quad \text{with} \quad \nu(f) \leq f \leq \mu(f).$$

That is, to a given fuzzy set we associate the characteristic functional of its necessity and possibility domains. These are the best approximations on a fuzzy set f from the bottom and top by crisp sets. That is, $\nu(f)$ (resp. $\mu(f)$) is a crisp set such that $\nu(f) \leq f$ (resp. $f \leq \mu(f)$) and for any crisp set e , if $e \leq f$ (resp. $f \leq e$), then $e \leq \nu(f)$ (resp. $\mu(f) \leq e$).

Further, it is possible to define a rough approximation in an equivalent way through the pair of necessity–impossibility fuzzy sets. In order to introduce this approach, let us define the *standard negation* on fuzzy sets as the unary operator $\neg : \mathcal{F}(X) \mapsto \mathcal{F}(X)$, assigning to any fuzzy set f its fuzzy negation $\neg f$ defined for any $x \in X$ by the law

$$\neg f(x) := 1 - f(x). \quad (3)$$

Using this negation it is possible to introduce also the *impossibility* operator \sim as the ‘not possibility’ $\sim f = \neg \mu(f)$ defined for any $x \in X$ by

$$\sim f(x) := \begin{cases} 1 & f(x) = 0 \\ 0 & f(x) \neq 0. \end{cases} \quad (4)$$

Remark 1. Here we defined the impossibility negation given the possibility operator and the standard negation as primitive. Let us also note that the other way round is a possible choice to tackle the problem. That is, given the two primitive unary operators \neg and \sim defined according to (3) and (4), the necessity and possibility can be otherwise defined according to the following laws: $\nu(f)(x) := \sim \neg f(x)$ and $\mu(f)(x) := \neg \sim f(x)$. (And so, $\sim f = \neg \mu(f)$; i.e., the negation \sim is just the impossibility $\neg \mu$.)

Now, the orthopair of necessity–impossibility rough approximation of a fuzzy set is given by the pair $r_i(f) := \langle \nu(f), \sim f \rangle$, with $\nu(f) + \sim f \leq \mathbf{1}$. Due to equation (4), it is easy to see that $\sim f$ represents what is usually called the *exterior* of f , and it is the characteristic functional of the *impossibility domain* $A_0(f)$, defined as

$$A_0(f) := \{x \in X : f(x) = 0\}.$$

A fuzzy set $e \in \mathcal{F}(X)$ is said to be *exact* (also *sharp*) with respect to a rough approximation iff $r(e) = \langle e, e \rangle$, and this happens iff $\forall x \in X, v(e)(x) = \mu(e)(x) = e(x)$. Therefore, the collection of all exact sets is $\mathcal{F}_e(X) = \{0, 1\}^X = \{\chi_A : A \subseteq X\}$; i.e., the sharp elements are the characteristic function of subsets of the universe, which are the usual *crisp* sets of standard fuzzy set theory. Moreover, the mapping $\chi : \mathcal{P}(X) \mapsto \mathcal{F}_e(X)$ assigning to any subset A of X ($A \in \mathcal{P}(X)$) its characteristic functional $\chi_A \in \mathcal{F}_e(X)$ is a Boolean lattice isomorphism. Let us stress that characteristic functionals, as Boolean valued functions, are in particular shadowed sets and this situation is represented in the following diagram:

$$\begin{array}{ccc}
 & [0, 1]^X \ni \chi_A & \\
 & \downarrow \text{id} & \\
 & \{0, 1, 1/2\}^X \ni \chi_A & \\
 & \downarrow \text{id} & \\
 A \in \mathcal{P}(X) & \xrightarrow{\chi} & \{0, 1\}^X \ni \chi_A
 \end{array}$$

Let us now show how the abstract rough approximation of a fuzzy set f , as the pair $r_i(f) = \langle v(f), \sim f \rangle$, allows one to single out the 0-approximation-induced shadowed set $s_0(f)$, according to the (1) applied to the case $\alpha = 0$. In fact, $v(f) = \chi_{A_1(f)}$ is the characteristic function of the elements which have value 1 in the induced shadowed set and $\sim f = \chi_{A_0(f)}$ is the characteristic function of the elements which have value 0. The other elements of the universe X represents the shadow of the shadowed set and they can be collected in $A_s(f) = \{x \in X : f(x) \neq 0, 1\} = X \setminus A_1(f) \cup A_0(f)$.

In order to define this relationship in an equational way, we introduce the Łukasiewicz norm and conorm [5] on fuzzy sets, given respectively by the following two mappings:

$$\begin{aligned}
 (f_1 \odot f_2)(x) &:= \max\{0, f_1(x) + f_2(x) - 1\}, \\
 (f \oplus g)(x) &:= \min\{1, f(x) + g(x)\}.
 \end{aligned}$$

Now, in the case of $\alpha = 0$, the shadowed set $s_0(f)$ defined in equation (1) can be obtained through a combination of the operators v and μ according to the following identity:

$$s_0(f) = \mu(f) \odot \left(v(f) \oplus \frac{1}{2} \right), \quad (5)$$

where we recall that $\frac{1}{2}$ is the fuzzy set identically equal to $\frac{1}{2}$. Indeed (and compare with (1) applied to the particular case of $\alpha = 0$) one immediately obtains

$$\forall x \in X, \quad \left[\mu(f) \odot \left(v(f) \oplus \frac{1}{2} \right) \right](x) = \begin{cases} 0 & f(x) = 0 \\ 1 & f(x) = 1 \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

Let us remark that $(\mathcal{F}(X), \odot, \oplus, \neg, \sim, 0, 1)$ is a structure characterizing fuzzy sets and able to cope with rough approximation and shadowed sets, as discussed above. This structure will be analyzed from an algebraic point of view in Section 27.3.

Of course, s_0 , the 0-shadow representation of f , gives only the induced shadowed set in the particular case of $\alpha = 0$, and it does not capture all the possible ones that can be obtained from a fuzzy set by equation (1). In order to consider all these possibilities, a generalization of the impossibility operator can

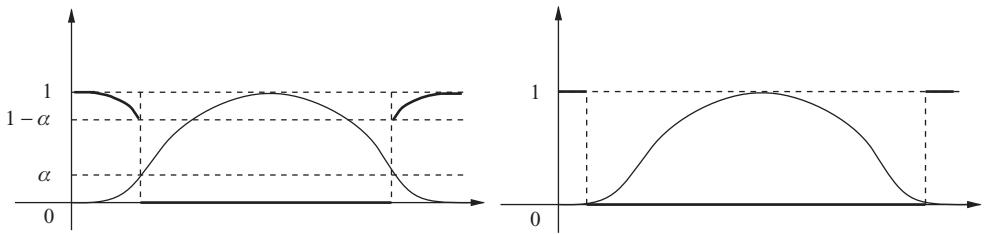


Figure 27.2 Generalized \sim_α : at the left it is represented the case $\alpha \in (0, 1/2)$ and at the right the case $\alpha = 0$

be defined as

$$\forall \alpha \in \left[0, \frac{1}{2}\right] \quad (\sim_\alpha f)(x) := \begin{cases} 1 - f(x) & \text{if } f(x) \leq \alpha, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Clearly, this is a generalization of \sim ; in fact, when $\alpha = 0$, we obtain $\sim_0 f = \sim f$. In Figure 27.2, it is represented a fuzzy set and its α -impossibility (i.e., \sim_α).

The derived operators μ_α and ν_α then become

$$\mu_\alpha(f)(x) := (\neg \sim_\alpha f)(x) = \begin{cases} f(x) & f(x) \leq \alpha \\ 1 & f(x) > \alpha, \end{cases} \quad (7)$$

$$\nu_\alpha(f)(x) := (\sim_\alpha \neg f)(x) = \begin{cases} f(x) & f(x) \geq (1 - \alpha) \\ 0 & f(x) < (1 - \alpha). \end{cases} \quad (8)$$

Let us introduce the shadowed set $s_\alpha(f)$, induced by the fuzzy set f and defined analogously to equation (5), as follows:

$$s_\alpha(f) := \mu_\alpha(f) \odot \left(\nu_\alpha(f) \oplus \frac{1}{2} \right).$$

This coincide with the shadowed set previously defined by the (1). Indeed, for arbitrary $x \in X$, one has

$$\left[\mu_\alpha(f) \odot \left(\nu_\alpha(f) \oplus \frac{1}{2} \right) \right](x) = \begin{cases} 0 & f(x) \leq \alpha \\ 1 & f(x) \geq 1 - \alpha \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

So, given a fuzzy set f , on one side we can obtain the rough approximation $r_\alpha(f) = \langle \nu_\alpha(f), \mu_\alpha(f) \rangle$ and on the other side we can induce the shadowed set $s_\alpha(f)$, formally expressed by the elements ν_α and μ_α of r_α . The relation between the two mappings r_α and s_α is given by the mapping $\sigma : \mathcal{F}(X) \times \mathcal{F}(X) \mapsto \mathcal{S}(X)$ which assigns to any pair of fuzzy sets $h, k \in \mathcal{F}(X)$ the shadowed set

$$\sigma(h, k) := h \odot \left(k \oplus \frac{1}{2} \right). \quad (9)$$

To be precise, we have the following identity between two mappings from $\mathcal{F}(X)$ into $\mathcal{S}(X)$:

$$\sigma \circ r_\alpha = s_\alpha.$$

Indeed, for any arbitrary fuzzy set $f \in \mathcal{F}(X)$, one has that

$$\sigma(r_\alpha(f)) = \sigma(\langle v_\alpha(f), \mu_\alpha(f) \rangle) = \mu_\alpha(f) \odot \left(v_\alpha(f) \oplus \frac{1}{2} \right) = s_\alpha(f).$$

In the following diagram all the three functions r_α , s_α , and σ are drawn, showing the relation among them:

$$\begin{array}{ccc} f & \xrightarrow{r_\alpha} & r_\alpha(f) \\ & \searrow s_\alpha & \downarrow \sigma \\ & & s_\alpha(f) \end{array}$$

Of course, the function σ is not a bijection as can be seen by considering the identically one shadowed set on the universe $[0, 1]$, $\forall x, \mathbf{1}(x) := 1$ and the shadowed set $k : [0, 1] \mapsto \{0, \frac{1}{2}, 1\}$ defined as

$$k(x) = \begin{cases} 1 & \text{if } x \in [0, 1/2], \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

These shadowed sets can be used to define two different pairs with the same image under σ : $\sigma(\mathbf{1}, k) = \mathbf{1}$ and $\sigma(\mathbf{1}, \mathbf{1}) = \mathbf{1}$.

However, we are interested only in the special situation of pairs of crisp sets $v(f)$ and $\mu(f)$, satisfying the further property $v(f) \leq \mu(f)$. In this case it can be proved that σ is a bijection (see Section 27.2.4).

27.2.2 Level Fuzzy Sets and α -Cuts

As pointed out also in [2], besides shadowed sets other methods to approximate fuzzy sets are known in literature. Here, we are going to consider level fuzzy sets and α -cuts. A γ -level fuzzy set [6] for $\gamma \in [0, 1]$ is formally obtained from a given fuzzy set f on the domain X through the following equation:

$$\forall x \in X \quad f_l^\gamma(x) := \begin{cases} f(x) & \text{if } f(x) \geq \gamma, \\ 0 & \text{otherwise.} \end{cases}$$

That is, a level fuzzy set is obtained from a fuzzy set by setting to zero the membership functions below the threshold value γ . Level fuzzy sets can be obtained by the operations on fuzzy sets previously defined. Indeed, the necessity v_α , as defined in equation (8), of a fuzzy set f is a $1 - \gamma$ level fuzzy set:

$$f_l^\gamma = v_{(1-\gamma)}(f).$$

Further, also the negation defined by Radecki [6] on level fuzzy sets

$$\forall x \in X \quad \neg_R(f_l^\gamma(x)) := \begin{cases} 0 & f(x) \geq \gamma \text{ and } f(x) > 1 - \gamma \\ 1 - f(x) & \gamma \leq f(x) \leq 1 - \gamma \\ 1 & f(x) < \gamma \end{cases}$$

can be recovered in a similar way:

$$\neg_R(f_l^\gamma) = v_{(1-\gamma)}(\neg f_l^\gamma).$$

We remark that this equation is well defined for all $\gamma \in [0, 1]$, and if $\gamma > 0.5$, then $\neg_R(f_l^\gamma(x)) \in \{0, 1\}$.

In a similar way, it is also possible to recover α -cuts. We recall that an α -cut (resp. *strong α -cut*) is obtained from a fuzzy set by setting to 1 the membership values greater than or equal to (resp. greater than) a fixed value α and to 0 the other ones [7]. Formally, let f be a fuzzy set defined over the domain X and $\alpha \in [0, 1]$, then the corresponding α -cut of f is

$$\forall x \in X \quad f_c^\alpha(x) := \begin{cases} 1 & \text{if } f(x) \geq \alpha, \\ 0 & \text{otherwise,} \end{cases}$$

and a strong α -cut is defined as

$$\forall x \in X \quad f_s^\alpha(x) := \begin{cases} 1 & \text{if } f(x) > \alpha, \\ 0 & \text{otherwise.} \end{cases}$$

So, we have that $v_\alpha(\mu_\alpha(f))$ is a $(1 - \alpha)$ -cut of f and $\mu_{(1-\alpha)}(v_{(1-\alpha)(f)})$ is a strong α -cut of f :

$$f_c^\alpha = v_{(1-\alpha)}(\mu_{(1-\alpha)}(f)) \quad f_s^\alpha = \mu_{(1-\alpha)}(v_{(1-\alpha)(f)}).$$

We remark that also in this case the parameter α can range in the whole unit interval $[0, 1]$.

Example 1. Let $X = [0, 1]$ and f the fuzzy set defined as $f(x) = x$. If we set $\gamma = 0.6$, then the $0.6 -$ level fuzzy set of f is

$$\forall x \in [0, 1] \quad f_l^{0.6}(x) = \begin{cases} x & \text{if } x \geq 0.6 \\ 0 & \text{otherwise.} \end{cases} \quad f_l^{0.6}(x) = v_{0.4}(f)(x)$$

On the other side, if $\alpha = 0.6$, then the α -cut and strong α -cut are, respectively,

$$\forall x \in [0, 1] \quad f_c^{0.6}(x) = \begin{cases} 1 & x \geq 0.6 \\ 0 & x < 0.6 \end{cases} \quad f_c^{0.6}(x) = v_{0.4}(\mu_{0.4}(f)),$$

$$\forall x \in [0, 1] \quad f_s^{0.6}(x) = \begin{cases} 1 & x > 0.6 \\ 0 & x \leq 0.6 \end{cases} \quad f_s^{0.6}(x) = \mu_{0.4}(v_{0.4}(f)).$$

Finally, if $\alpha = 0.4$, the shadowed set relative to α is

$$s_{0.4}(f) = \begin{cases} 0 & f(x) \leq 0.4 \\ \frac{1}{2} & 0.4 \leq f(x) < 0.6 \\ 1 & f(x) \geq 0.6. \end{cases}$$

27.2.3 Generalized Rough Approximations

Another approach to approximate a fuzzy set has been presented in [8] through a generalization of the operators v and μ . Indeed, the approximation of a fuzzy set given by the operators v and μ is a two-valued set. A shadowed set, on the other side, gives a three-valued approximation of a fuzzy set. Generalizing this idea, it is possible to define an n -valued approximation. Let us consider a fuzzy set $f : X \rightarrow [0, 1]$ and let r_1, \dots, r_n be rational numbers in $[0, 1]$, such that $0 < r_1 < r_2 < \dots < r_{n-1} < r_n < 1$. Then

- The n -internal approximation of f is a fuzzy set $\xi : X \rightarrow \{0, r_1, \dots, r_n, 1\}$ such that $\xi \leq f$ and for each $x \in X$, $\xi(x) = r_i$ iff $f(x) \in [r_i, r_{i+1})$.
- The n -outer approximation of f is a fuzzy set $\tau : X \rightarrow \{0, r_1, \dots, r_n, 1\}$ such that $f \leq \tau$ and for each $x \in X$, $\tau(x) = r_i$ iff $f(x) \in (r_{i-1}, r_i]$.

Table 27.1 A comparison among the approximations ξ_f, τ_f relative to the three-valued set $\{0, \frac{1}{2}, 1\}$ and the shadowed set s_α

$f(x)$	$\xi_f(x)$	$s_\alpha(x)$	$\tau_f(x)$
0	0	0	0
$(0, \alpha]$	0	0	$\frac{1}{2}$
$(\alpha, \frac{1}{2})$	0	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
$(\frac{1}{2}, 1 - \alpha)$	$\frac{1}{2}$	$\frac{1}{2}$	1
$[1 - \alpha, 1)$	$\frac{1}{2}$	1	1
1	1	1	1

Thus, the generalized rough approximation of a fuzzy set f is given by the pair $\langle \xi_f, \tau_f \rangle$. In this environment, the exact sets, i.e., the fuzzy sets which coincide with their approximations ξ and τ , are the fuzzy sets which assume values only in the set $\{0, r_1, \dots, r_n, 1\}$.

If we consider an approximation based on the set $\{0, \frac{1}{2}, 1\}$, we have that to any fuzzy set f the following two approximation mappings are associated:

$$\xi_f(x) = \begin{cases} 0 & x < \frac{1}{2} \\ \frac{1}{2} & x \in [\frac{1}{2}, 1] \\ 1 & x = 1 \end{cases}$$

$$\tau_f(x) = \begin{cases} 0 & x = 0 \\ \frac{1}{2} & x \in (0, \frac{1}{2}] \\ 1 & x \in (\frac{1}{2}, 1]. \end{cases}$$

Clearly, it holds that $\xi_f \leq f \leq \tau_f$. Thus, we have that all the different functions ξ_f, τ_f , and s_α approximate a fuzzy set through three values. Among these functions there exists the order relation $\xi_f \leq s_\alpha \leq \tau_f$, as can be seen in Table 27.1.

We underline that, on the contrary, there does not exist an order relation between a fuzzy set and its related shadowed set.

27.2.4 ICS and Shadowed Sets

Let us consider the collection of fuzzy sets $\mathcal{F}(X)$ and introduce the binary relation $\perp \subseteq \mathcal{F}(X) \times \mathcal{F}(X)$ defined as

$$f \perp g \quad \text{iff} \quad \forall x \in X : f(x) + g(x) \leq 1, \quad (10)$$

which turns out to be an *orthogonality* relation according to [9].

An *orthopair of fuzzy sets* (called IFS according to a standard term, see Remark 2) on the universe X is any pair of fuzzy sets $\langle f_A, g_A \rangle \in \mathcal{F}(X) \times \mathcal{F}(X)$, under the orthogonality condition $f_A \perp g_A$. The collection of all IFSs will be denoted by $\mathcal{IF}(X)$; this set is non-empty since it contains the particular elements $O := \langle \mathbf{0}, \mathbf{1} \rangle$, $I := \langle \mathbf{1}, \mathbf{0} \rangle$, and $H := \langle \frac{1}{2}, \frac{1}{2} \rangle$.

Remark 2. The acronym IFS takes its origin from *intuitionistic fuzzy sets*, the name given by Atanassov [10–12] to the orthopairs of fuzzy sets. However, there is an undergoing debate about the

theoretical correctness of this name (see [13–18]). Thus, we prefer to use the term orthopair or equivalently IFS as a name itself, without giving it a particular meaning as an acronym of something which as to do with intuitionism.

In this section we investigate a particular subclass of the class $\mathcal{IF}(X)$ of all IFSs on the universe X and its relationship to shadowed sets. To this aim, let us consider the collection, denoted by $\mathcal{IC}(X)$, of all orthopairs $\langle \chi_{A_1}, \chi_{A_0} \rangle$ of characteristic functions of subsets A_1, A_0 of X . Trivially, $\mathcal{IC}(X) \subseteq \mathcal{IF}(X)$ and

$$\chi_{A_1} \perp \chi_{A_0} \quad \text{iff} \quad A_1 \cap A_0 = \emptyset.$$

Therefore, $\mathcal{IC}(X)$ consists of pairs of *crisp* sets, also called ICS (as the crisp counterpart of fuzzy IFS), A_1 and A_0 under the orthogonality condition of their disjointness. So we can identify ICSs with pairs of mutually disjoint subsets of X

$$\langle \chi_{A_1}, \chi_{A_0} \rangle \iff \langle A_1, A_0 \rangle$$

and we denote their collection also by $\mathcal{IC}(X) = \{\langle A_1, A_0 \rangle \in \mathcal{P}(X) \times \mathcal{P}(X) : A_1 \cap A_0 = \emptyset\}$. The subset A_1 (resp. A_0) is the *certainty* (resp. *impossibility*) *domain* of the involved ICS $\langle A_1, A_0 \rangle$.

Remark 3. At the best of our knowledge, the concept of ICS has been introduced for the first time by M. Yves Gentilhomme in [19] (see also [20, 21]) in an equivalent way with respect to the one described here. Indeed, in these papers pairs of ordinary subsets of the universe X of the kind $\langle A_1, A_p \rangle$, under the condition $A_1 \subseteq A_p$ are considered. The mapping $\langle A_1, A_0 \rangle \rightarrow \langle A_1, (A_0)^c \rangle$ institute a one-to-one and onto correspondence which allows one to identify the two approaches.

The pairs $\langle A_1, A_0 \rangle$ are also considered in [22]. In this context they are called *classical preclusive propositions* and analyzed from the point of view of algebraic rough approximations. Indeed, to any ICS pair $\langle A_1, A_0 \rangle$, it is possible to assign a fuzzy set $f := \frac{1}{2}(\chi_{A_1} + \chi_{A_0^c})$, such that its rough approximation $r_i(f)$ coincides with the starting pair; i.e., $r_i(f) = \langle v(f), \sim f \rangle = \langle A_1, A_0 \rangle$.

Let us note that, later on, Çoker [23] introduced in an independent way the so-called *intuitionistic set* as a weakening of IFS to classical sets, whose definition exactly coincides with ICS.

Finally, we introduce for any ICS $\langle A_1, A_0 \rangle$ its *uncertainty domain* $A_u = X \setminus (A_1 \cup A_0)$. Then the mapping $\mathcal{IC}(X) \mapsto \mathcal{S}(X)$ defined by the law

$$\langle A_1, A_0 \rangle \rightarrow \chi_{A_1} + \frac{1}{2} \cdot \chi_{A_0} \quad \left[\chi_{A_1} + \frac{1}{2} \cdot \chi_{A_0} \right] (x) = \begin{cases} 0 & \text{if } x \in A_0 \\ 1 & \text{if } x \in A_1 \\ \frac{1}{2} & \text{if } x \in A_u \end{cases} \quad (11)$$

is a one-to-one and onto correspondence which allows one to identify ICS and shadowed sets. In this way all the algebraic structures of ICSs (see [18]) are automatically inherited by shadowed sets, in particular the one of Kleene algebra and the one of Heyting algebra. However, in Section 27.3 we are going to directly analyze this algebraic approach.

27.2.5 Variable-Precision Rough Sets and Shadowed Sets

An interesting subcase of ICS is represented by rough sets, where given a set A , A_1 is the lower approximation of A , A_0 the exterior region (i.e., the complement of the upper approximation) and A_u the boundary region. In particular, we consider the variable-precision rough set (VPRS) model, introduced by Ziarko in [24], which, as we will see, is strictly linked to shadowed sets.

VPRS are a generalization of Pawlak rough sets obtained by relaxing the definition of set membership (or equivalently the notion of subset). Let $\gamma = \{E_1, E_2, \dots, E_n\}$ be a partition of a universe X , usually

generated by an equivalence relation on X . For all $x \in X$, we denote by $E(x)$ the equivalence class $E \in \gamma$ such that $x \in E$. For any subset of objects $H \subseteq X$, the *rough membership function* is defined as $\mu_H^\gamma : X \mapsto [0, 1]$:

$$\mu_H^\gamma(y) := \frac{|E(y) \cap H|}{|E(y)|}.$$

Clearly, if $E(y) \subseteq H$, then $\mu_H^\gamma(y) = 1$ and y certainly belongs to the set H . Vice versa, if $E(y) \cap H = \emptyset$ (and thus $y \notin H$), then, coherently, $\mu_H^\gamma(y) = 0$.

This membership function can be used to define the variable-precision rough approximation.

Definition 3. Let $\gamma = \{E_1, E_2, \dots, E_n\}$ be a partition of a universe X . Given a set of objects $H \subseteq X$, the variable-precision lower approximation of H is defined for any fixed $\alpha \in [0, \frac{1}{2}]$ as

$$L_\alpha(H) = \{y \in X : \mu_H^\gamma(y) \geq 1 - \alpha\}.$$

On the other hand, the variable-precision upper approximation of H is defined as

$$U_\alpha(H) = \{y \in X : \mu_H^\gamma(y) > \alpha\}.$$

It is easy to see that if $\alpha = 0$, the lower and upper variable-precision approximations coincide with the standard Pawlak rough approximations.

Remark 4. The original definition of Ziarko was based on the notion of *relative classification error*. Let H and K be two sets, then, the relative classification error $c(H, K)$ of H with respect to K is defined as

$$c(H, K) = \begin{cases} 1 - \frac{|H \cap K|}{|H|} & |H| > 0 \\ 0 & |H| = 0. \end{cases}$$

As stated by Ziarko in [24], $c(H, K)$ is ‘the relative degree of misclassification of set H with respect to set K .’ Thus, we correctly have that if $H \subseteq K$, then $c(H, K) = 0$; i.e., there is no error in classifying all the elements of a set H as elements of a super set K .

Using this measure of error, and once fixed $\beta \in [0, \frac{1}{2}]$, the lower and upper approximations are respectively defined as

$$\begin{aligned} L_\beta(H) &= \{x \in X : c(E(x), H) \leq \beta\}, \\ U_\beta(H) &= \{x \in X : c(E(x), H) < 1 - \beta\}. \end{aligned}$$

It can easily be seen that this definition is equivalent to the one given above.

By Definition 3, it follows that the exterior and boundary regions are defined as $\mathcal{E}_\alpha(H) = \{y \in X : \mu_H^\gamma(y) \leq \alpha\}$ and $B_\alpha(H) = \{y \in X : \alpha < \mu_H^\gamma(y) < 1 - \alpha\}$. Now, given a set H , we can define its rough approximation as the pair $(L_\alpha(H), U_\alpha(H))$ or, equivalently, as $(L_\alpha(H), \mathcal{E}_\alpha(H))$, which is a pair of disjoint sets, i.e., an ICS.

Thus, using equation (11) we obtain the mapping $X \mapsto \mathcal{S}(X)$ defined as

$$y \mapsto \begin{cases} 0 & \text{if } \mu_H^\gamma(y) \leq \alpha, \\ 1 & \text{if } \mu_H^\gamma(y) \geq 1 - \alpha, \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

By comparing this equation with (1), we see that it is the α -shadowed set generated by the rough membership function μ_H^γ . So, a variable-precision rough set can be seen as the α -shadowed set of the fuzzy set μ_H^γ .

27.2.6 Balance of Vagueness

We have seen different ways to approximate a fuzzy set through a simpler construct. A peculiarity of the shadowed sets approach is that it gives a method to compute an optimal threshold value α , according to the following statement [4]:

The primary goal we intend to accomplish in this way is the one of localizing and balancing the factor of uncertainty that is inherently associated with any fuzzy set.

Indeed, a shadowed set localizes uncertainty by moving it from the intervals $[0, \alpha)$ and $(1 - \alpha, 1]$ to the interval $(\alpha, 1 - \alpha)$, i.e., to the ‘shadow.’ In order to balance this exchange of information, Pedrycz proposes the following method.

For a given fuzzy set $f : X \mapsto [0, 1]$ and for every $\alpha \in [0, 1/2]$, let us divide the universe X in three regions:

$$\begin{aligned}\mathcal{Q}_1(\alpha) &= \{x : f(x) \in (0, \alpha)\}, \\ \mathcal{Q}_2(\alpha) &= \{x : f(x) \in (1 - \alpha, 1]\}, \\ \mathcal{Q}_3(\alpha) &= \{x : f(x) \in (\alpha, 1 - \alpha)\}.\end{aligned}$$

Then, the optimal α is the one minimizing (if possible, setting to zero) the following quantity [4, 25]:

$$V_f(\alpha) := \left| \int_{\mathcal{Q}_1(\alpha)} f(x)dx + \int_{\mathcal{Q}_2(\alpha)} (1 - f(x))dx - \int_{\mathcal{Q}_3(\alpha)} dx \right|. \quad (12)$$

With respect to this definition, in the case of triangular fuzzy sets the optimal α is $\sqrt{2} - 1$, and in the case of Gaussian-shaped fuzzy set is 0.395.

Let us note that instead of equation (12), sometimes (see, for instance, [2, 3]) it is used as a very different quantity written in the following form:

$$V'_f(\alpha) := \left| \int_{\mathcal{Q}_1(\alpha)} f(x)dx + \int_{\mathcal{Q}_2(\alpha)} (1 - f(x))dx - \int_{\mathcal{Q}_3(\alpha)} f(x)dx \right|. \quad (13)$$

We think that the more suitable is the one given by equation (12). Indeed it corresponds to the semantic given to the shadow ‘where there are no specific membership values assigned, but we admit the entire unit interval as feasible membership grades’ [25]. Furthermore, if on the other hand equation (13) is used, the values of α given before cannot be recovered, since in the case of triangular (resp. Gaussian) fuzzy sets we obtain $\alpha = 0.366$ (resp. $\alpha = 0.335$).

The possibility to compare different values of α and choosing the best one is clearly useful in applications, among which we mention fuzzy clustering, image processing, and fuzzy decision making (for a more detailed discussion, see [2]). Also for this reason, shadowed sets ‘can be regarded as a concise and operationally appealing vehicle of processing fuzzy sets’ and they can play a significant role in granular computing leading ‘to a full-fledged calculus of information granules’ [25].

27.3 Algebraic Approach

In this section, we will give an algebraic framework to the paradigms introduced in the previous section. First, BZMV algebras are used to characterize fuzzy and shadowed sets and the particular case of s_0 (see equations (4) and (5)). In order to generalize this approach to all s_α , pre-BZMV algebras are introduced

and studied. Finally, it is described how ŁΠ algebras can characterize generalized rough approximations of Section 27.2.3.

27.3.1 BZMV Algebras and Equivalent Structures

In this section we propose BZMV^{dM} algebras [26, 27] as an algebraic approach to describe both fuzzy and shadowed sets.

Definition 4. A de Morgan Brouwer Zadeh many-valued (BZMV^{dM}) algebra is a system $\mathbf{A} = \langle A, \oplus, \neg, \sim, 0 \rangle$, where A is a non-empty set, \oplus is a binary operation, \neg and \sim are unary operations, and 0 is a constant, obeying the following axioms:

$$(\text{BZMV1}) (a \oplus b) \oplus c = (b \oplus c) \oplus a.$$

$$(\text{BZMV2}) a \oplus 0 = a.$$

$$(\text{BZMV3}) \neg(\neg a) = a.$$

$$(\text{BZMV4}) \neg(\neg a \oplus b) \oplus b = \neg(a \oplus \neg b) \oplus a.$$

$$(\text{BZMV5}) \sim a \oplus \sim \sim a = \neg 0.$$

$$(\text{BZMV6}) a \oplus \sim \sim a = \sim \sim a.$$

$$(\text{BZMV7}) \sim \neg[(\neg(a \oplus \neg b) \oplus b)] = \neg(\sim \sim a \oplus \neg \sim \sim b) \oplus \neg \sim \sim b.$$

On the basis of the primitive notions of a BZMV^{dM} algebra, it is possible to define the following further derived operations:

$$a \odot b := \neg(\neg a \oplus \neg b). \quad (14a)$$

$$a \vee b := \neg(\neg a \oplus b) \oplus b = a \oplus (\neg a \odot b). \quad (14b)$$

$$a \wedge b := \neg(\neg(a \oplus \neg b) \oplus \neg b) = a \odot (\neg a \oplus b). \quad (14c)$$

It is easy to prove that operations \vee and \wedge are the *join* and *meet* binary connectives of a *distributive lattice*, which can be considered as algebraic realization of logical disjunction and conjunction; in particular, they are idempotent operations. On the other hand, the \oplus and \odot are not idempotent and they are the well-known MV disjunction and MV conjunction connectives of the Chang approach [28, 29]. That is, let $\langle A, \oplus, \neg, \sim, 0 \rangle$ be a BZMV^{dM} algebra, then, the substructure $\langle A, \oplus, \neg, 0 \rangle$ is an MV algebra.

An interesting strengthening of BZMV^{dM} algebras are three-valued BZMV algebras, first introduced in [26].

Definition 5. A BZMV^3 algebra is a BZMV^{dM} algebra \mathbf{A} , where axiom (BZMV5) is replaced by the following condition:

$$(s\text{BZMV5}) \sim a \oplus (a \oplus a) = 1.$$

The importance of BZMV^3 algebras is due to the fact that, as showed in [26], they are categorically equivalent to three-valued Łukasiewicz algebras [30] and to MV³ algebras [31], which permit an algebraic semantic characterization for a three-valued logic. Indeed, the paradigmatic model of BZMV^3 algebras is the three-element set $\{0, \frac{1}{2}, 1\}$ endowed with the following operations:

$$a \oplus b = \min\{a + b, 1\},$$

$$\neg a = 1 - a,$$

$$\sim a = \begin{cases} 1 & \text{if } a = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Let us note that BZMV^{dM} algebras are equivalent to other well-known structures. Without entering into details (see [32]) we mention:

- HW algebras, a pasting of Heyting and Wajsberg algebras [33];
- MV_Δ algebras, obtained by adding a Baaz's operator Δ [34, 35] to MV algebras;
- Stonean MV algebra, a particular class of MV algebras introduced by Belluce [36].

Thus, with BZMV^{dM} algebras it is possible to recover several other approaches. In particular, we also underline that Heyting algebras are a substructure of BZMV^{dM} algebra. So, there exists a strong relation with the intuitionistic approach to algebraic logic. Indeed, it can be shown that \sim is an intuitionistic (Brouwer) negation; i.e., the following are satisfied:

- (B1) $a \leq \sim(\sim a)$.
- (B2) $\sim(a \vee b) = \sim a \wedge \sim b$.
- (B3) $a \wedge \sim a = 0$.

Further, also the dual de Morgan law, ' $\sim(a \wedge b) = \sim a \vee \sim b$ ', and the contraposition law, ' $a \leq b$ implies $\sim b \leq \sim a$ ' hold, but, as required by intuitionism, the excluded middle law ' $a \vee \sim a = 1$ ' is in general not valid.

On the other hand, the unary operator \neg behaves as a Kleene (fuzzy, Zadeh) negation. That is, \neg satisfies the following properties:

- (K1) $\neg(\neg a) = a$.
- (K2) $\neg(a \vee b) = \neg a \wedge \neg b$.
- (K3) $a \wedge \neg a \leq b \vee \neg b$.

Also this one is clearly a non-standard negation, since there exist elements which do not satisfy the non-contradiction law ' $a \wedge \neg a = 0$ ' and the excluded middle law ' $a \vee \neg a = 1$ '. The dual de Morgan law, ' $\neg(a \wedge b) = \neg a \vee \neg b$ ', is on the contrary satisfied, as well as the contraposition law, ' $a \leq b$ implies $\neg b \leq \neg a$ '.

Through the interaction of these two negation it is possible to define two new operators $v : A \mapsto A$, $v(a) := \sim \neg a$, and $\mu : A \mapsto A$, $\mu(a) := \neg \sim a$, which have interesting modal and topological properties.

Proposition 1. *In any BZMV^{dM} algebra, the operators v and μ can, respectively, be interpreted as necessity and possibility operators on a S5-modal system; i.e., the following conditions hold:*

1. $v(1) = 1$ (*N principle*).
2. $v(a) \leq a \leq \mu(a)$ (*T principle*).
3. $a \leq v(\mu(a))$ (*B principle*).
4. $v(v(a)) = v(a)$ $\mu(\mu(a)) = \mu(a)$ (*S₄ principle*).
5. $\mu(a) = v(\mu(a))$ $v(a) = \mu(v(a))$ (*S₅ principle*).

Let us note that these are non-standard modalities due to the following peculiarities:

1. The operators v and μ satisfy all S5-modal system properties but are defined on a Kleene lattice instead of a Boolean algebra.
2. They satisfy, contrary to the classical case, the distributivity laws $v(a) \vee v(b) = v(a \vee b)$ and $\mu(a) \wedge \mu(b) = \mu(a \wedge b)$.

Moreover, from a topological point of view, v and μ can be interpreted as an interior and a closure operator, and they can be used to define a rough approximation through clopen (exact) elements.

Proposition 2. [26, 37] Let $\langle A, \oplus, \neg, \sim, 0 \rangle$ be a $BZMV^{dM}$ algebra. Then, the map $v : A \rightarrow A$ such that $v(a) := \sim \neg a$ is a (additive) topological interior operator; i.e., the following conditions hold:

(I_0)	$1 = v(1)$	(normalized).
(I_1)	$v(a) \leq a$	(decreasing).
(I_2)	$v(a) = v(v(a))$	(idempotent).
(I_3)	$v(a \wedge b) = v(a) \wedge v(b)$	(multiplicativity).
(I_4)	$v(a) \vee v(b) = v(a \vee b)$	(additivity).
(I_5)	$a \wedge v(\neg a) = 0$	(v -interconnection).

Properties (I_0) – (I_3) qualify the mapping I as a topological interior operator [38, 39], whereas properties (I_4) and (I_5) are specific conditions satisfied in this particular case. In the context of topological interior operators the elements which coincide with their interior are called *open* elements and the collection of all these open elements is denoted as

$$\mathbb{O}(A) = \{a \in A : a = v(a)\}.$$

Proposition 3. [26, 37] Let $\langle A, \oplus, \neg, \sim, 0 \rangle$ be a $BZMV^{dM}$ algebra. Then, the map $\mu : A \rightarrow A$ such that $\mu(a) := \neg \sim a$ is a (multiplicative) topological closure operator; i.e., the following conditions hold:

(C_0)	$0 = \mu(0)$	(normalized).
(C_1)	$a \leq \mu(a)$	(increasing).
(C_2)	$\mu(a) = \mu(\mu(a))$	(idempotent).
(C_3)	$\mu(a) \vee \mu(b) = \mu(a \vee b)$	(additivity).
(C_4)	$\mu(a \wedge b) = \mu(a) \wedge \mu(b)$	(multiplicativity).
(C_5)	$a \wedge \neg\mu(a) = 0$	(μ -interconnection).

Dually to the previous case, properties (C_0) – (C_3) qualify the mapping C as a topological closure operator [38, 39], whereas properties (C_4) and (C_5) are specific conditions satisfied in this particular case. Further, the elements which coincide with their closure are called *closed* elements and their collection is denoted as

$$\mathbb{C}(A) = \{a \in A : a = \mu(a)\}.$$

In a generic structure equipped with an interior and a closure operation, an element which is both open and closed is said to be *clopen* and the collection of all such clopen elements is denoted by $\mathbb{CO}(A)$. However, in a $BZMV^{dM}$ algebra the set of open elements coincide with the set of closed elements; thus, we have that

$$\mathbb{CO}(A) = \mathbb{O}(A) = \mathbb{C}(A).$$

This subset of clopen elements is not empty; indeed, both $0, 1 \in \mathbb{CO}(A)$.

Following [40], in any de Morgan lattice, i.e., a lattice with a de Morgan negation $'$, and once given an interior (equiv., closure) operator it is possible to define an abstract approximation space $\langle A, \mathbb{CO}(A), \mathbb{CO}(A) \rangle$ according to [41]. In this structure, A is the lattice of approximable elements, $\mathbb{CO}(A)$ the sublattice of lower and upper definable elements, and v and μ are the approximation operators. Thus, it can be shown that the following properties are satisfied:

- (L1) $v(a)$ is sharp ($v(a) \in \mathbb{CO}(A)$).
- (L2) $v(a)$ is an *inner* (*lower*) approximation of a ($v(a) \leq a$).
- (L3) $v(a)$ is the *best* inner approximation of a by sharp elements:
let $e \in \mathbb{CO}(A)$ be such that $e \leq a$, then $e \leq v(a)$.

Analogously,

- (U1) $\mu(a)$ is sharp ($\mu(a) \in \mathbb{CO}(A)$).
- (U2) $\mu(a)$ is an outer (upper) approximation of a ($a \leq \mu(a)$).
- (U3) $\mu(a)$ is the best outer approximation of a by sharp elements:
let $f \in \mathbb{CO}(A)$ be such that $a \leq f$, then $\mu(a) \leq f$.

That is, $v(a)$ (resp. $\mu(a)$) turns out to be the best approximation from the bottom (resp. top) of a by sharp elements. Consequently, given an element a of a $BZMV^{dM}$ algebra, its rough approximation by exact elements is given by the pair

$$r(a) := \langle v(a), \mu(a) \rangle \quad \text{with } v(a) \leq a \leq \mu(a),$$

which is the image of the element a under the *rough approximation mapping* $r : A \mapsto \mathbb{CO}(A) \times \mathbb{CO}(A)$. Finally, an equivalent way to define a rough approximation is given by the mapping $r_i : A \mapsto \mathbb{CO}(A) \times \mathbb{CO}(A)$, defined as $r_i(a) := \langle v(a), \sim a \rangle$.

27.3.1.1 Fuzzy and Shadowed Sets

We now come back to the concrete cases of fuzzy and shadowed sets, and we show how it is possible to give them in a canonical way the structure of $BZMV^{dM}$ algebras.

Proposition 4. *Let $\mathcal{F}(X) = [0, 1]^X$ be the collection of fuzzy sets on the universe X . Let us define the operators:*

$$\begin{aligned} (f \oplus g)(x) &:= \min\{1, f(x) + g(x)\} \\ \neg f(x) &:= 1 - f(x) \\ \sim f(x) &:= \begin{cases} 1 & \text{if } f(x) = 0, \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

and the identically zero fuzzy set: $\forall x \in X, \mathbf{0}(x) := 0$. Then, the structure $\langle \mathcal{F}(X), \oplus, \neg, \sim, \mathbf{0} \rangle$ is a $BZMV^{dM}$ algebra such that $\mathbf{1} = \sim \mathbf{0} = \neg \mathbf{0}$ is the identically one fuzzy set: $\forall x \in X, \mathbf{1}(x) = 1$.

The structure $\langle \mathcal{F}(X), \oplus, \neg, \sim, \mathbf{0} \rangle$ is not a $BZMV^3$ algebra; indeed, it is sufficient to apply property (sBZMV5) to the element $\frac{1}{3}$.

Similarly, it is possible to give the structure of $BZMV^3$ algebra to the collection of shadowed sets $\mathcal{S}(X) = \{0, \frac{1}{2}, 1\}^X$ on the universe X . The operations syntactically are exactly as in Proposition 4, but they are defined on the domain of shadowed sets.

Proposition 5. *Let $\mathcal{S}(X) = \{0, \frac{1}{2}, 1\}^X$ be the collection of shadowed sets on the universe X . Then, the structure $\langle \mathcal{S}(X), \oplus, \neg, \sim, \mathbf{0} \rangle$, where $\oplus, \neg, \sim, \mathbf{0}$ are defined as in Proposition 4, is a $BZMV^3$ algebra.*

The lattice operators induced from a $BZMV^{dM}$ algebra according to (14b) and (14c) can be consequently defined both in $\mathcal{F}(X)$ and in $\mathcal{S}(X)$. They turn out to coincide with the usual min–max operators, i.e., Gödel t -norm and t -conorm on fuzzy sets:

$$\begin{aligned} (f_1 \vee f_2)(x) &= \max\{f_1(x), f_2(x)\}, \\ (f_1 \wedge f_2)(x) &= \min\{f_1(x), f_2(x)\}. \end{aligned}$$

Analogously, the MV conjunction \odot defined in (14a) is exactly the Łukasiewicz t -norm:

$$(f_1 \odot f_2)(x) = \max\{0, f_1(x) + f_2(x) - 1\}.$$

Finally, the topological (modal) operators ν and μ coincide with the ones introduced in equations (2).

Now, we have all the instruments in order to define the mapping s_0 of equation (5). In this algebraic context, the mapping $s_0 : \mathcal{F}(X) \rightarrow \mathcal{S}(X)$, $f \mapsto s_0(f)$ is a mapping between the $BZMV^{dM}$ algebra of fuzzy sets and the $BZMV^{dM}$ algebra of shadowed set. However, it is not a bijection nor a homomorphism between $BZMV^{dM}$ algebras, as can be seen in the following counterexample.

Example 2. Let us consider the fuzzy sets $f_1, f_2 : [0, 1] \mapsto [0, 1]$ defined as

$$f_1(x) := \begin{cases} 0.2 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \quad f_2(x) := \begin{cases} 0.3 & \text{if } x = 0 \\ 0 & \text{otherwise.} \end{cases}$$

So, $f_1 \neq f_2$ but

$$s_0(f_1) = s_0(f_2) = \begin{cases} 0.5 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

and this proves that s_0 is not a bijection.

Furthermore, stressing with symbols \oplus_S and \oplus_F the MV disjunction (also ‘truncated’ sum) operation acting on $\mathcal{S}(X)$ and $\mathcal{F}(X)$, respectively, we have

$$[s_0(f_1) \oplus_S s_0(f_2)](x) = \begin{cases} 1 & f(x) = 0 \\ 0 & \text{otherwise} \end{cases} \neq \begin{cases} \frac{1}{2} & f(x) = 0 \\ 0 & \text{otherwise} \end{cases} = [s_0(f_1 \oplus_F f_2)](x)$$

and so s_0 is neither a $BZMV^{dM}$ algebras homomorphism.

When considering also the generalization from s_0 to s_α , we have to substitute \sim with \sim_α . However, if we consider the structure $\langle [0, 1]^X, \oplus, \neg, \sim_\alpha, 0 \rangle$, the system $\langle [0, 1]^X, \oplus, \neg, \sim_\alpha, 0 \rangle$ is no more a $BZMV^{dM}$ algebra. In fact, for instance, axiom $BZMV6$ is not satisfied. Given a fuzzy set f , we have

$$f(x) \oplus (\sim_\alpha \sim_\alpha f)(x) = \begin{cases} 1 & f(x) > \alpha \\ f(x) & f(x) \leq \alpha \end{cases} \neq \begin{cases} 1 & f(x) > \alpha \\ 0 & f(x) \leq \alpha \end{cases} = (\sim_\alpha \sim_\alpha f)(x).$$

Section 27.3.2 will be devoted to the study of an algebraization of the structure $\langle \mathcal{F}(X), \oplus, \neg, \sim_\alpha, \mathbf{0} \rangle$ containing this new operator \sim_α .

27.3.2 Pre- $BZMV^{dM}$ Algebra

We, now, introduce an algebra obtained as a weakening of $BZMV^{dM}$ algebras. The advantage of this new structure is that it admits as a model the collection of fuzzy sets endowed with the operator \sim_α .

Definition 6. A structure $\mathbf{A} = \langle \mathcal{A}, \oplus, \neg, \sim_w, 0 \rangle$ is a pre- $BZMV^{dM}$ algebra if the following are satisfied:

1. The substructure $\langle \mathcal{A}, \oplus, \neg, 0 \rangle$ is an MV algebra, whose induced lattice operations are defined as

$$\begin{aligned} a \vee b &:= \neg(\neg a \oplus b) \oplus b \\ a \wedge b &:= \neg(\neg(a \oplus \neg b) \oplus \neg b) \end{aligned}$$

and, as usual, the partial order is $a \leq b$ iff $a \wedge b = a$ (iff $\neg a \oplus b = a \rightarrow_L b = 1$).

2. The following properties are satisfied:

- (a) $a \oplus \sim_w \sim_w a = \neg \sim_w a$.
- (b) $\sim_w a \wedge \sim_w b \leq \sim_w (a \vee b)$.

- (c) $\sim_w a \vee \sim_w b = \sim_w (a \wedge b)$.
(d) $\sim_w \neg a \leq \sim_w \neg \sim_w \neg a$.

In general, it is possible to show that any $BZMV^{dM}$ algebra is a pre- $BZMV^{dM}$ algebra and that the vice versa does not hold [42].

Proposition 6. [42] Let $\langle \mathcal{A}, \oplus, \neg, \sim_w, 0 \rangle$ be a pre- $BZMV^{dM}$ algebra, then it is a $BZMV^{dM}$ algebra iff the following interconnection rule holds:

$$\forall a \in \mathcal{A}, \sim_w \sim_w a = \neg \sim_w a.$$

The unary operator \sim_w satisfies some typical properties of a negation, in particular both de Morgan laws and the contraposition law:

1. $\sim_w (a \wedge b) = \sim_w a \vee \sim_w b$ (\vee de Morgan law).
2. $\sim_w (a \vee b) = \sim_w a \wedge \sim_w b$ (\wedge de Morgan law).
3. If $a \leq b$, then $\sim_w b \leq \sim_w a$ (contraposition law).

However, it is not an intuitionistic negation; in fact, in general, it satisfies neither the non-contradiction law (property (B3)) nor the weak double negation law (property (B1)).

Further, the link between the two negations is established by the following rule:

$$\sim_w a \leq \neg a$$

and the following boundary condition is satisfied: $\sim_w 0 = \neg 0$.

Anyway, also in a pre- $BZMV^{dM}$ algebra, it is possible to introduce modal operators of necessity, $\nu_w(a) := \sim_w \neg a$ and possibility $\mu_w(a) := \neg \sim_w a$. However, in this structure ν_w and μ_w do not have an S_5 -like behavior but only an S_4 -like one (always based on a Kleene lattice instead of on a Boolean one).

Proposition 7. [42] Let $\langle \mathcal{A}, \oplus, \neg, \sim_w, 0 \rangle$ be a pre- $BZMV^{dM}$ algebra. Then, for every $a \in \mathcal{A}$ the following properties are satisfied:

1. $\nu_w(a) \leq a \leq \mu_w(a)$ (T principle).
2. $\nu_w(\nu_w(a)) = \nu_w(a)$ $\mu_w(\mu_w(a)) = \mu_w(a)$ (S_4 principle).

In general the following properties are not satisfied by these weak modalities:

3. $a \leq \nu_w(\mu_w(a))$ (B principle).
4. $\mu_w(a) = \nu_w(\mu_w(a)), \nu_w(a) = \mu_w(\nu_w(a))$ (S_5 principle).

Even if the necessity and possibility mappings have a weaker modal behavior in pre- $BZMV^{dM}$ algebras than in $BZMV^{dM}$ algebras, they can still be used to define a lower and upper approximation, and it turns out that ν_w is an (additive) topological interior operator and μ_w is a (multiplicative) topological closure operator.

Proposition 8. [42] Let $\langle \mathcal{A}, \oplus, \neg, \sim_w, 0 \rangle$ be a pre- $BZMV^{dM}$ algebra. Then the map $\nu_w : \mathcal{A} \rightarrow \mathcal{A}$ such that $\nu_w(a) := \sim_w \neg a$ is an (additive) topological interior operator; i.e.,

- | | | |
|---------|--|---------------------|
| (I_0) | $1 = \nu_w(1)$ | (normalized). |
| (I_1) | $\nu_w(a) \leq a$ | (decreasing). |
| (I_2) | $\nu_w(a) = \nu_w(\nu_w(a))$ | (idempotent). |
| (I_3) | $\nu_w(a \wedge b) = \nu_w(a) \wedge \nu_w(b)$ | (multiplicativity). |
| (I_4) | $\nu_w(a) \vee \nu_w(b) = \nu_w(a \vee b)$ | (additivity). |

Dually, the map $\mu_w : \mathcal{A} \rightarrow \mathcal{A}$ such that $\mu_w(a) := \neg \sim_w a$ is a (multiplicative) topological closure operator. That is, the following are satisfied:

- | | | |
|---------|--|---------------------|
| (C_0) | $0 = \mu_w(0)$ | (normalized). |
| (C_1) | $a \leq \mu_w(a)$ | (increasing). |
| (C_2) | $\mu_w(a) = \mu_w(\mu_w(a))$ | (idempotent). |
| (C_3) | $\mu_w(a) \vee \mu_w(b) = \mu_w(a \vee b)$ | (additivity). |
| (C_4) | $\mu_w(a \wedge b) = \mu_w(a) \wedge \mu_w(b)$ | (multiplicativity). |

The collection of *open* and *closed* elements with respect to these operators are $\mathbb{O}_w(\mathcal{A}) = \{a \in \mathcal{A} : a = v_w(a)\}$ and $\mathbb{C}_w(\mathcal{A}) = \{a \in \mathcal{A} : a = \mu_w(a)\}$.

Comparing these results with the ones of Propositions 2 and 3, we note that in this case the interconnection rule does not hold for both interior and closure operators. Further, in general, in a pre-BZMV^{dM} algebra, the subsets of \mathcal{A} of open and closed elements do not coincide, $\mathbb{C}_w(\mathcal{A}) \neq \mathbb{O}_w(\mathcal{A})$; neither one is a subset of the other. So, it is worthwhile to consider also the set of all *clopen* elements, i.e., elements which are both closed and open:

$$\mathbb{CO}_w(\mathcal{A}) = \mathbb{C}_w(\mathcal{A}) \cap \mathbb{O}_w(\mathcal{A}).$$

The collection of clopen elements is a Boolean algebra; to be precise, the following proposition holds.

Proposition 9. [42] Let $\mathbf{A} = \langle \mathcal{A}, \oplus, \neg, \sim_w, 0 \rangle$ be a pre-BZMV^{dM} algebra. Then the collection $\mathbb{CO}_w(\mathcal{A})$ of all its clopen elements satisfies the following properties:

1. The lattice connectives coincide with the MV ones:

$$\forall e, f \in \mathbb{CO}_w(\mathcal{A}), \quad e \wedge f = e \odot f, \quad \text{and} \quad e \vee f = e \oplus f.$$

2. The two negation connectives (Kleene and Brouwer) coincide:

$$\forall e \in \mathbb{CO}_w(\mathcal{A}), \quad \neg e = \sim_w e.$$

3. The structure $\mathbf{A}_e = \langle \mathbb{CO}_w(\mathcal{A}), \wedge, \vee, \neg, 0 \rangle$ is a Boolean lattice (algebra), which is the largest pre-BZMV^{dM} subalgebra of \mathbf{A} that is at the same time a Boolean algebra with respect to the same operations $\wedge (= \odot)$, $\vee (= \oplus)$, and $\neg (= \sim)$.

As in the case of BZMV^{dM} algebras, the presence of an interior and a closure operator enables the introduction of an abstract approximation space generated by a pre-BZMV^{dM} algebra as the structure $\langle \mathcal{A}, \mathbb{O}_w(\mathcal{A}), \mathbb{C}_w(\mathcal{A}) \rangle$, where

- \mathcal{A} is the set of approximable elements.
- $\mathbb{O}_w(\mathcal{A}) \subseteq \mathcal{A}$ is the set of *innerdefinable* elements, such that 0 and $1 \in \mathbb{O}_w(\mathcal{A})$.
- $\mathbb{C}_w(\mathcal{A}) \subseteq \mathcal{A}$ is the set of *outerdefinable* elements, such that 0 and $1 \in \mathbb{C}_w(\mathcal{A})$.
- $v_w : \mathcal{A} \rightarrow \mathbb{O}_w(\mathcal{A})$ is the *inner approximation map*.
- $\mu_w : \mathcal{A} \rightarrow \mathbb{C}_w(\mathcal{A})$ is the *outer approximation map*.

For any element $a \in \mathcal{A}$, its *rough approximation* is defined as the pair:

$$r_w(a) := \langle v_w(a), \mu_w(a) \rangle, \quad \text{with } v_w(a) \leq a \leq \mu_w(a).$$

This approximation is the best approximation by open (resp. closed) elements that it is possible to define on a pre-BZMV^{dM} structure; i.e., there hold properties similar to (L1)–(L3) and (U1)–(U3) and the only difference is that here we have to distinguish between open-exact and closed-exact elements.

27.3.2.1 Fuzzy Sets

The collection of all fuzzy sets can be equipped with a structure of pre-BZMV^{dM} algebra, according to the following result.

Proposition 10. [42] Let $\mathcal{F}(X)$ be the collection of fuzzy sets based on the universe X and let $\alpha \in [0, \frac{1}{2}]$. Once defined the standard \oplus and \neg operators on \mathcal{F} , and the \sim_α negation as in equation (6), then the structure $\mathbb{F}_\alpha = \langle \mathcal{F}(X), \oplus, \neg, \sim_\alpha, \mathbf{0} \rangle$ is a pre-BZMV^{dM} algebra, which is not a BZMV^{dM} algebra.

We now give an example of the fact that \mathbb{F}_α is not a BZMV^{dM} algebra.

Example 3. Let us consider the structure $\mathbb{F}_{0.4}$ with $X = \mathbb{R}$ and define the fuzzy set $f(x) = 0.3$ for all $x \in \mathbb{R}$. Then, $\sim_\alpha f(x) \oplus \sim_\alpha \sim_\alpha f(x) = 0.7$ for all x . So, axiom (BZMV5) is not satisfied.

In this context the modal operators of necessity ν_w and possibility μ_w are defined as in equations (7) and (8). As expected they do not satisfy the B and S_5 principles of Proposition 1.

Example 4. Let us consider the algebra $\mathbb{F}_{0.4}$, with $X = [0, 1]$, and define the fuzzy set

$$f(x) = \begin{cases} 0.3 & \text{if } x < \frac{1}{2}, \\ 0.7 & \text{otherwise.} \end{cases}$$

We have

$$\begin{aligned} \mu_\alpha(f(x)) &= \begin{cases} 0.3 & x < \frac{1}{2} \\ 1 & x \geq \frac{1}{2} \end{cases} \neq \begin{cases} 0 & x < \frac{1}{2} \\ 1 & x \geq \frac{1}{2} \end{cases} = \nu_\alpha(\mu_\alpha(f(x))). \\ \nu_\alpha(f(x)) &= \begin{cases} 0 & x < \frac{1}{2} \\ 0.7 & x \geq \frac{1}{2} \end{cases} \neq \begin{cases} 0 & x < \frac{1}{2} \\ 1 & x \geq \frac{1}{2} \end{cases} = \mu_\alpha(\nu_\alpha(f(x))). \end{aligned}$$

Finally, $f(x)$ is incomparable with $\nu_\alpha(\mu_\alpha(f(x)))$.

In the pre-BZMV^{dM} algebraic context of fuzzy sets discussed in Proposition 10, the collection of closed and open elements are, respectively,

$$\begin{aligned} \mathbb{C}_\alpha(\mathcal{F}(X)) &= \{f \in \mathcal{F}(X) : f(x) > \alpha \quad \text{iff} \quad f(x) = 1\}. \\ \mathbb{O}_\alpha(\mathcal{F}(X)) &= \{f \in \mathcal{F}(X) : f(x) < 1 - \alpha \quad \text{iff} \quad f(x) = 0\}. \end{aligned}$$

The clopen sets are the 0–1 valued fuzzy sets, $\mathbb{CO}_\alpha(\mathcal{F}(X)) = \{0, 1\}^X$.

Example 5. In the universe $[0, 1]$, once set $\alpha = 0.4$, an example of open element is the fuzzy set

$$f_1(x) = \begin{cases} 0 & \text{if } x < \frac{1}{2}, \\ 0.7 & \text{otherwise,} \end{cases}$$

and an example of closed set is the fuzzy set

$$f_2(x) = \begin{cases} 0.3 & \text{if } x < \frac{1}{2}, \\ 1 & \text{otherwise.} \end{cases}$$

The fuzzy sets f_1 and f_2 are drawn in Figure 27.3.

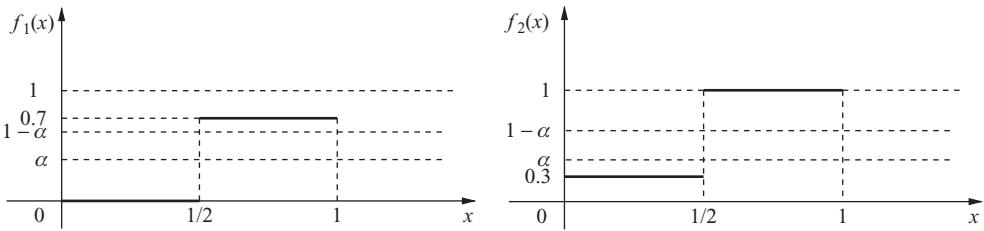


Figure 27.3 Example of open fuzzy set, f_1 , and closed fuzzy set, f_2

When considering the mapping $s_\alpha : \mathcal{F}(X) \mapsto \mathcal{S}(X)$, as defined in equation (1), we see that it satisfies the following properties:

$$\begin{aligned} s_\alpha(f) &= s_\alpha(s_\alpha(f)) && \text{(idempotent).} \\ f_1 \leq f_2 &\text{ implies } s_\alpha(f_1) \leq s_\alpha(f_2) && \text{(monotone).} \end{aligned}$$

Further, if f is a shadowed set, i.e., a fuzzy set which assumes only three values, $f : X \mapsto \{0, \frac{1}{2}, 1\}$, then it also holds

$$s_\alpha(f) = f.$$

We can also enrich the collection of shadowed sets $\{0, \frac{1}{2}, 1\}^X$ with the operation \sim_α in order to equip it with a pre-BZMV^{dM} structure. However, it can easily be proved that in this case \sim_α is equal to \sim_0 for all $\alpha \in [0, \frac{1}{2})$ and so we again obtain a BZMV^d algebra.

Finally, we note that also α -cuts and level fuzzy sets can be obtained by pre-BZMV formulas, since as we have seen in Section 27.2.2, they can be expressed through the combination of the v_w and μ_w operators.

27.3.3 $\mathbb{L}\Pi_{\frac{1}{2}}$ Algebra

In this section we introduce $\mathbb{L}\Pi$ algebras in order to give a theoretical approach to generalized rough approximations, as defined in Section 27.2.3. $\mathbb{L}\Pi$ algebras are obtained joining MV algebras with product algebras [35], and here we consider the particular subclass of $\mathbb{L}\Pi_{\frac{1}{2}}$ algebras, i.e., $\mathbb{L}\Pi$ algebras with a constant element $\frac{1}{2}$ such that $\frac{1}{2} = -\frac{1}{2}$. Let us note that $\mathbb{L}\Pi$ algebras are a stronger structure than BZMV^{dM} (and thus also of pre-BZMV^{dM}) algebras.

Definition 7. An $\mathbb{L}\Pi_{\frac{1}{2}}$ algebra is a structure $\langle A, \oplus, \neg, \otimes, \rightarrow_p, 0, 1, \frac{1}{2} \rangle$, such that once defined

$$\begin{aligned} \neg_p a &:= a \rightarrow_p 0 \\ \Delta a &:= \neg_p \neg a \\ a \odot b &:= \neg(\neg a \oplus \neg b) \\ a \ominus b &:= a \odot \neg b \\ a \wedge b &:= a \ominus (a \ominus b) \\ a \vee b &:= a \oplus (b \ominus a) \\ a \rightarrow_L b &:= \neg a \oplus b \\ a \leftrightarrow_L b &:= (a \rightarrow_L b) \odot (b \rightarrow_L a) \end{aligned}$$

the following hold:

- (LP1) $\langle A, \oplus, \neg, 0, 1 \rangle$ is an MV algebra.
- (LP2) Δ is a Baaz operators [34]; i.e., it satisfies the following axioms:

- ($\Delta 1$) $\Delta a \vee \neg \Delta a = 1$
- ($\Delta 2$) $\Delta(a \vee b) \leq \Delta a \vee \Delta b$
- ($\Delta 3$) $\Delta a \leq a$
- ($\Delta 3$) $\Delta a \leq \Delta \Delta a$
- ($\Delta 4$) $\Delta a \odot \Delta(a \rightarrow_L b) \leq \Delta b$
- ($\Delta 5$) $\Delta 1 = 1$.

(LP3) $\langle A, \otimes, 1 \rangle$ is a commutative monoid.

(LP4) The following axioms are satisfied:

- (a) $a \otimes (b \ominus c) = (a \otimes b) \ominus (a \otimes c)$
- (b) $\Delta(a \leftrightarrow_L b) \wedge \Delta(c \leftrightarrow_L u) \leq ((a \otimes c) \leftrightarrow_L (b \otimes u))$
- (c) $\Delta(a \leftrightarrow_L b) \wedge \Delta(c \leftrightarrow_L u) \leq ((a \rightarrow_p c) \leftrightarrow_L (b \rightarrow_p u))$
- (d) $a \wedge \neg_p a = 0$
- (e) $\Delta(a \rightarrow_L b) \leq (a \rightarrow_p b)$
- (f) $\Delta(b \rightarrow_L a) \leq (a \otimes (a \rightarrow_p b) \leftrightarrow_L b)$
- (g) $\frac{1}{2} = \neg \frac{1}{2}$

It is easy to show that the collection of all fuzzy sets on a given domain have the structure of an $\text{Ł}\Pi_{\frac{1}{2}}$ algebra once endowed with proper operators.

Proposition 11. Let $\mathcal{F}(X) = [0, 1]^X$ be the collection of fuzzy sets on the universe X . Let us define the operators

$$\begin{aligned} (f \oplus g)(x) &:= \min\{1, f(x) + g(x)\}, \\ (f \otimes g)(x) &:= f(x) \cdot g(x), \\ (f \rightarrow_p g)(x) &:= \begin{cases} 1 & \text{if } f(x) \leq g(x) \\ \frac{g(x)}{f(x)} & \text{otherwise,} \end{cases} \\ \neg f(x) &:= 1 - f(x). \end{aligned}$$

Then, the structure $\langle \mathcal{F}(X), \oplus, \neg, \otimes, \rightarrow_p, \mathbf{0}, \mathbf{1}, \frac{1}{2} \rangle$ is a $\text{Ł}\Pi_{\frac{1}{2}}$ algebra.

In particular, let us consider the case of $X = [0, 1]^k$ with $k \in \mathbb{N}$. Then, the free $\text{Ł}\Pi$ algebra over k generators defined as the structure $\text{Free}(k) = \langle \mathcal{F}([0, 1]^k), \oplus, \neg, \otimes, \rightarrow_p, \mathbf{0}, \mathbf{1}, \frac{1}{2} \rangle$ is of great interest. Indeed, it is well known that there exists a bijection between Boolean algebras and Boolean logic as well as a strict connection with classical sets theory. When considering weaker algebraic structures, and the corresponding many-valued logics, we have that to each of them it can be associated a collection of fuzzy sets. For instance, formulas on MV algebras, and hence on Łukasiewicz logic, define those fuzzy sets which are McNaughton functions [43], i.e., functions $f : [0, 1]^k \mapsto [0, 1]$ which are continuous and piecewise linear.

In our case, $\text{Ł}\Pi$ algebra is the Lindembaum–Tarski algebra of $\text{Ł}\Pi$ logic (we are not going into details here, see for instance [44]) and the following result about fuzzy sets holds.

Theorem 1. [45] A function $f : [0, 1]^k \rightarrow [0, 1]$ is a truth table of a formula of $\text{Ł}\Pi_{\frac{1}{2}}$ iff there is a finite partition of $[0, 1]^k$ such that each block of the partition is a \mathbb{Q} -semialgebraic set, and f restricted to each block is a fraction of two polynomials with rational coefficients.

Now, we are going to define a lower and an upper approximation map on $\text{Free}(k)$, i.e., on fuzzy sets definable in $\text{Ł}\Pi$ -logic according to Theorem 1, in such a way that the effect on a fuzzy set coincide with the generalized approximations of Section 27.2.3. First of all, we consider the lower approximation, i.e., a mapping $v_n : \mathcal{F}([0, 1]^k) \mapsto \mathcal{F}([0, 1]^k)$, such that for any fuzzy set $f : [0, 1]^k \rightarrow [0, 1]$ and for any set of rationals $R = \{0 < r_1 < \dots < r_n < 1\}$, $v_n(f)$ is the approximation of f through the rationals in R .

Thus, let f be a fuzzy subset of the k -cube $[0, 1]^k$. Then, $v_n(f_\varphi)$ is the n -inner approximation defined on the basis of n as follows:

- The 0-inner approximation of f is the fuzzy set $v_0(f) = \Delta f$.
- The 1-inner approximation of f is

$$\Delta f \vee \mathbf{r} \otimes \Delta(\mathbf{r} \rightarrow f).$$

- In general, the n -inner approximation of f is

$$\Delta f \vee \left(\bigvee_{i=1}^n \mathbf{r}_i \otimes \Delta(\mathbf{r}_i \rightarrow f) \right),$$

with $0 < r_1 < r_2 < \dots < r_n < 1$.

Proposition 12. [8] For each $n \in \mathbb{N}$, let $R = \{0 < r_1 < \dots < r_n < 1\}$ be a set of rational numbers. Let $v_n : \mathcal{F}([0, 1]^k) \rightarrow \mathcal{F}([0, 1]^k)$ be such that $v_n(f)$ is the n -inner approximation of f with respect to R , as defined above. Then v_n is a topological interior operator; that is, the following hold:

- (I1) $v_n(\mathbf{1}) = \mathbf{1}$.
- (I2) $v_n(f) \leq f$.
- (I3) $v_n(f) = v_n(v_n(f))$.
- (I4) $v_n(f_1 \wedge f_2) = v_n(f_1) \wedge v_n(f_2)$.

Further, v_n also satisfies the following properties:

- (I5) $v_n(f_1 \vee f_2) = v_n(f_1) \vee v_n(f_2)$.
- (I6) $f \odot v_n(\neg f) = \mathbf{0}$.

With respect to this interior operator, it is possible to define the subset $\mathbb{O}_n(k) \subseteq \mathcal{F}_k(\mathbb{L}\Pi^{\frac{1}{2}})$ of open elements as

$$\mathbb{O}_n(k) = \{f \in \mathcal{F}([0, 1]^k) : v_n(f) = f\}.$$

As already stated in Section 27.2.3, if the interior is relative to the set of rationals $R = \{0 < r_1 < \dots < r_n < 1\}$, then the elements of $\mathbb{O}_n(k)$ are those fuzzy sets which assume only values in R ; that is, for all $\underline{x} \in [0, 1]^k$, $f(\underline{x}) \in R$.

In a dual way, it is possible to define an outer n -approximation $\mu_n : \mathcal{F}([0, 1]^k) \mapsto \mathcal{F}([0, 1]^k)$ for a given set of rationals R . Let f be a fuzzy subset of the k -cube, then

- The 0-outer approximation of f is given by ∇f .
- The 1-outer approximation of f is given by

$$\nabla f \wedge \neg(\neg \mathbf{r} \otimes \Delta(f \rightarrow \mathbf{r}))$$

with $0 < r < 1$.

- In general, if $0 < r_1 < r_2 < \dots < r_n < 1$, the n -outer approximation of f is given by

$$\nabla f \wedge \left(\bigwedge_{i=1}^n \neg(\neg \mathbf{r}_i \otimes \Delta(f \rightarrow \mathbf{r}_{i+1})) \right).$$

The map $\mu_n : \mathcal{F}([0, 1]^k) \mapsto \mathcal{F}([0, 1]^k)$ is a topological closure operator; i.e., the following proposition holds.

Proposition 13. [8] For each $n \in \mathbb{N}$, let $R = \{0 < r_1 < \dots < r_n < 1\}$ be a set of rational numbers. Let $\mu_n : \mathcal{F}([0, 1]^k) \mapsto \mathcal{F}([0, 1]^k)$ be such that $\mu_n(f)$ is the n -outer approximation of f with respect to R as defined above. Then μ_n is a topological closure operator:

- (O1) $\mu_n(\mathbf{0}) = \mathbf{0}$.
- (O2) $\mu_n(f) \geq f$.
- (O3) $\mu_n(\mu_n(f)) = \mu_n(f)$.
- (O4) $\mu_n(f_1 \vee f_2) = \mu_n(f_1) \vee \mu_n(f_2)$.

Further, μ_n also satisfies the following properties:

- (O5) $\mu_n(f_1 \wedge f_2) = \mu_n(f_1) \wedge \mu_n(f_2)$.
- (O6) $f \odot \mu_n(\neg f) = \mathbf{0}$.

Thus, the collection of closed elements $\mathbb{C}_n(k) \subseteq \mathcal{F}([0, 1]^k)$ is the following set:

$$\mathbb{C}_n(k) = \{f \in \mathcal{F}([0, 1]^k) : \mu_n(f) = f\}.$$

It is easily seen that if a fuzzy set is open with respect to the interior operator relative to the set of rationals $R = \{0 < r_1 < \dots < r_n < 1\}$, then f is also closed with respect to the closure operator based on the same set R . Vice versa, any closed fuzzy set is also open. In conclusion, the set of clopen elements $\mathbb{CO}_n(k) := \mathbb{O}_n(k) \cap \mathbb{C}_n(k)$ coincide with both the set of closed and open elements.

Having an interior and a closure operator, also in this context, it is possible (once fixed the set of rationals used to generate the approximations) to define a rough approximation space where

- $\mathcal{F}_k(\mathbb{L}\Pi^{\frac{1}{2}})$ is the set of *approximable elements*.
- $\mathbb{CO}_n(k)$ is the set of *exact elements*.
- $\forall f \in \mathcal{F}_k(\mathbb{L}\Pi^{\frac{1}{2}})$, $v_n(f)$ is the *lower approximation* of f and $\mu_n(f)$ the *upper approximation* of f .

In this context a rough approximation of a fuzzy set f is just the pair

$$r_n(f) := (v_n(f), \mu_n(f)) \in \mathbb{CO}_n(k) \times \mathbb{CO}_n(k) \quad \text{with} \quad v_n(f) \leq f \leq \mu_n(f).$$

Clearly, this is the best approximation by exact elements; i.e., v_n satisfies properties (L1)–(L3) (see Section 27.3.1) and μ_n properties (U1)–(U3).

27.4 Conclusion

Shadowed sets have been analyzed in their theoretical aspects. First of all, the relationship among shadowed sets, fuzzy sets, and rough approximations has been outlined. For a given fuzzy set f , the induced rough approximation is just one of the possible shadowed sets obtainable by f , precisely the one with $\alpha = 0$. On the other hand, two different generalized rough approximations have been introduced. The first one, r_n , depends on a set of n rational values and the second one, r_w , on a threshold value α . For both approximations, their relationship with shadowed sets has been outlined. Further, analogies and differences among shadowed sets and level fuzzy sets, α -cuts, ICS, and VPRS are explained.

A review of all the above-mentioned paradigms has been made from a theoretical point of view. In particular, an algebraic approach to shadowed set, as well as to fuzzy sets and rough approximations, has been given in the form of BZMV algebras. Further, $\mathbb{L}\Pi^{\frac{1}{2}}$ (resp. pre-BZMV) algebras are used to properly treat generalized rough approximations r_n (resp. r_w). Let us remark that the lower (resp. upper) approximation generated in the VPRS model is not a topological operator, since in general it is not decreasing (resp. increasing), as showed in [24].

Acknowledgment

This work has been supported by MIUR\PRIN project ‘Automata and Formal Languages: mathematical and application-driven studies.’

References

- [1] L.A. Zadeh. Fuzzy sets. *Inf. Control* 8 (1965) 338–353.
- [2] W. Pedrycz. Shadowed sets: Representing and processing fuzzy sets. *IEEE Trans. Syst. Man Cybern.* 28(1) (1998) 103–109.
- [3] W. Pedrycz. Shadowed sets: Bridging fuzzy and rough sets. In: S. Pal and A. Skowron (eds), *Rough Fuzzy Hybridization*. Springer-Verlag, Singapore, 1999, pp. 179–199.
- [4] W. Pedrycz and G. Vukovich. Granular computing with shadowed sets. *Int. J. Intell. Syst.* 17 (2002) 173–197.
- [5] E.P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*. Kluwer Academic, Dordrecht, 2000.
- [6] T. Radecki. Level fuzzy sets. *J. Cybern.* 7 (1977) 189–198.
- [7] D. Dubois and H. Prade. *Fuzzy Sets and Systems. Theory and Applications*. Academic Press, New York, 1980.
- [8] D. Ciucci and T. Flaminio. Generalized rough approximations in $\mathbb{L}\Pi^{\frac{1}{2}}$. *Int. J. Approx. Reason.* 2007. doi:10.1016/j.ijar.2007.10.2006.
- [9] G. Cattaneo and A. Manià. Abstract orthogonality and orthocomplementation. *Proc. Camb. Phil. Soc.* 76 (1974) 115–132.
- [10] K.T. Atanassov and S. Stoeva. Intuitionistic fuzzy sets. In: *Polish Symposium on Interval & Fuzzy Mathematics*, Poznan, August 1983, pp. 23–26.
- [11] K.T. Atanassov. Intuitionistic fuzzy sets. *Fuzzy Sets Syst.* 20 (1986) 87–96.
- [12] K.T. Atanassov. More on intuitionistic fuzzy sets. *Fuzzy Sets Syst.* 33 (1989) 37–45.
- [13] G. Cattaneo and D. Ciucci. Generalized negations and intuitionistic fuzzy sets. A criticism to a widely used terminology. In: *Proceedings of International Conference in Fuzzy Logic and Technology (EUSFLAT03)*, University of Applied Sciences of Zittau–Goerlitz, Zittau, 2003, pp. 147–152.
- [14] G. Cattaneo and D. Ciucci. Intuitionistic fuzzy sets or orthopair fuzzy sets? In: *Proceedings of International Conference in Fuzzy Logic and Technology (EUSFLAT03)*, University of Applied Sciences of Zittau–Goerlitz, Zittau, 2003, pp. 153–158.
- [15] D. Dubois, S. Gottwald, P. Hajek, J. Kacprzyk, and H. Prade. Terminological difficulties in fuzzy set theory – the case of intuitionistic fuzzy sets. *Fuzzy Sets Syst.* 156 (2005) 485–491.
- [16] P. Grzegorzewski and E. Mrowka. Some notes on (Atanassov) Intuitionistic fuzzy sets. *Fuzzy Sets Syst.* 156 (2005) 492–495.
- [17] K.T. Atanassov. Answer to D. Dubois, S. Gottwald, P. Hajek, J. Kacprzyk and H. Prade’s paper ‘terminological difficulties in fuzzy set theory – the case of intuitionistic fuzzy sets.’ *Fuzzy Sets Syst.* 156 (2005) 496–499.
- [18] G. Cattaneo and D. Ciucci. Basic intuitionistic principles in fuzzy set theories and its extensions (a terminological debate on atanassov IFS). *Fuzzy Sets Syst.* 157 (2006) 3198–3219.
- [19] M.Y. Gentilhomme. Les ensembles flous en linguistique. *Cahiers de linguistique théorétique et appliquée, Bucarest* 47 (1968) 47–65.
- [20] G.C. Moisil. Les ensembles flous et la logique à trois valeurs (texte inédit). In: [21], 1972, chapter 15, pp. 99–103.
- [21] G.C. Moisil. *Essais sur les logiques non Chrysippiennes*. Edition de l’Académie de la République Socialiste de Roumanie, Bucharest, 1972.
- [22] G. Cattaneo and G. Nisticò. Brouwer-Zadeh posets and three valued Łukasiewicz posets. *Fuzzy Sets Syst.* 33 (1989) 165–190.
- [23] D. Coker. A note on intuitionistic sets and intuitionistic points. *Turk. J. Math.* 20 (1996) 343–351.
- [24] W. Ziarko. Variable precision rough sets model. *J. Comput. Syst. Sci.* 43(1) (1993) 39–59.
- [25] W. Pedrycz. Granular computing with shadowed sets. In: *RSFDGrC 2005*, Vol. 3641 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, 2005, pp. 23–32.
- [26] G. Cattaneo, M.L. Dalla Chiara, and R. Giuntini. Some algebraic structures for many-valued logics. *Tatra Mt. Math. Publ.* 15 (1998) 173–196. Special Issue: Quantum Structures II, Dedicated to Gudrun Kalmbach.
- [27] G. Cattaneo, R. Giuntini, and R. Pilla. BZMV^{dM} and Stonian MV algebras (applications to fuzzy sets and rough approximations). *Fuzzy Sets Syst.* 108 (1999) 201–222.
- [28] C.C. Chang. Algebraic analysis of many valued logics. *Trans. Am. Math. Soc.* 88 (1958) 467–490.
- [29] E. Turunen. *Mathematics behind Fuzzy Logic*. Physica-Verlag, Heidelberg, 1999.
- [30] A. Monteiro. Sur la définition des algèbres de Łukasiewicz trivalentes. *Bull. Math. Soc. Sci. Math. Phys. R. P. Roumaine* 7(1–2) 1963.

- [31] R. Grigolia. Algebraic analysis of Łukasiewicz Tarski's n -valued logical systems. In: R. Wójcicki and G. Malinowski (eds), *Selected Papers on Łukasiewicz Sentential Calculi*. Polish Academy of Sciences, Ossolineum, Wrocław, 1997, pp. 81–92.
- [32] G. Cattaneo, D. Ciucci, R. Giuntini, and M. Konig. Algebraic structures related to many valued logical systems. Part II: Equivalence among some widespread structures. *Fundam. Inf.* 63(4) (2004) 357–373.
- [33] G. Cattaneo, D. Ciucci, R. Giuntini, and M. Konig. Algebraic structures related to many valued logical systems. Part I: Heyting Wajsberg algebras. *Fundam. Inf.* 63(4) (2004) 331–355.
- [34] M. Baaz. Infinite-valued Gödel logics with 0-1 projections and relativizations. In: P. Hájek (ed.), *GÖDEL96—Logical Foundations of Mathematics, Computer Science and Physics*, Vol. 6 of *Lecture Notes in Logic*. Springer-Verlag, Berlin, 1996, pp. 23–33.
- [35] P. Hájek. *Metamathematics of Fuzzy Logic*. Kluwer, Dordrecht, 1998.
- [36] L.P. Belluce. Generalized fuzzy connectives on MV-algebras. *J. Math. Anal. Appl.* 206(1) (1997) 485–499.
- [37] G. Cattaneo and G. Marino. Brouwer-Zadeh posets and fuzzy set theory. In: A. Di Nola and A. Ventre (eds), *Proceedings of the 1st Napoli Meeting on Mathematics of Fuzzy Systems*, Napoli, June 1984, pp. 34–58.
- [38] G. Birkhoff. *Lattice Theory*, Vol. XXV of *American Mathematical Society Colloquium Publication*, 3rd ed. American Mathematical Society, Providence, RI, 1967.
- [39] H. Rasiowa and R. Sikorski. *The Mathematics of Metamathematics*, Vol. 41 of *Monografie Matematyczne*, 3rd ed. Polish Scientific Publishers, Warszawa, 1970.
- [40] G. Cattaneo and D. Ciucci. Some methodological remarks about categorical equivalence in the abstract approach to roughness. Part I. In: *RSKT 2006*, Vol. 4062 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, 2006, pp. 277–283.
- [41] G. Cattaneo. Abstract approximation spaces for rough theories. In: L. Polkowski and A. Skowron (eds) *Rough Sets in Knowledge Discovery 1*. Physica-Verlag, Heidelberg New York, 1998, pp. 59–98.
- [42] G. Cattaneo and D. Ciucci. Shadowed sets and related algebraic structures. *Fundam. Inf.* 55 (2003) 255–284.
- [43] R. McNaughton. A theorem about infinite-valued sentential logic. *J. Symbol. Log.* 16 (1951) 1–13.
- [44] P. Cintula. Advances in the \mathcal{LP} and $\mathcal{LP}_{\frac{1}{2}}$ logics. *Arch. Math. Log.* 42 (2003) 449–468.
- [45] F. Montagna and G. Panti. Adding structure to mv-algebras. *J. Pure Appl. Algebr.* 164 (2001) 356–387.

28

Fuzzy Representations of Spatial Relations for Spatial Reasoning

Isabelle Bloch

28.1 Introduction

Spatial reasoning can be defined as the domain of spatial knowledge representation, in particular spatial relations between spatial entities, and of reasoning on these entities and relations. This field has been largely developed in artificial intelligence, in particular using qualitative representations based on logical formalisms. In image interpretation and computer vision, it is much less developed and is mainly based on quantitative representations. A typical example in this domain concerns model-based structure recognition in images. The model constitutes a description of the scene where objects have to be recognized. This description can be of iconic type, as for instance a digital map or a digital anatomical atlas, or of symbolic type, as linguistic descriptions of the main structures. The model can be attached to a specific scene, the typical example being a digital map used for recognizing structures in an aerial or satellite image of a specific region. It can also be more generic, as an anatomical atlas, which is a schematic representation that can be used for recognizing structures in a medical image of any person. In both types of descriptions (iconic and symbolic), objects are usually described through some characteristics like shape, size, and appearance in the images. But this is generally not enough to discriminate all objects in the scene, in particular if they are embedded in a complex environment. For instance in a magnetic resonance image of the brain, several internal structures appear as smooth shapes with similar grey levels, making their individual recognition difficult. Similar examples can be found in other application domains. In such cases, spatial relations play a crucial role, and it is important to include them in the model in order to guide the recognition [1]. The importance of spatial relations has been similarly recognized in many different works. Many authors have stressed the importance of topological relations, but metric relations, such as distances and directional relative position, are also important. More generally, philosophical considerations lead to interpret geometry as a theory of relations [2], enhancing the importance of spatial relations in spatial reasoning.

Moreover, imprecision has to be taken into account in spatial reasoning. Its causes can be found at several levels, from the observed phenomenon to the semantics of some relations (such as ‘left of,’ ‘quite far,’ etc.), the available knowledge and the type of question to be answered. Different levels of granularity [3, 4] are also involved in this domain.

In summary, the main ingredients in problems related to spatial reasoning include knowledge representation (including spatial relations), imprecision representation and management, fusion of heterogeneous

information, and decision making. Fuzzy set theory is of great interest to provide a consistent mathematical framework for all these aspects. It allows representing imprecision of objects, relations, knowledge, and aims, and it provides a flexible framework for information fusion as well as powerful tools for reasoning and decision making.

This paper is organized as follows. Based on different views of space in different domains, the granularity aspects involved in spatial reasoning are highlighted in Section 28.2. Advantages and limits of reasoning at the most detailed level on the one hand and at the most abstract one on the other hand are summarized in Sections 28.3 and 28.4, respectively. A focus on intermediate level using fuzzy representations is provided in Section 28.5, along with a discussion of their advantages. Fuzzy representations can be seen as an answer to the semantic gap problem, as described in Section 28.6, which is a very important feature of such representations. Then, in Section 28.7, we summarize our work on definitions of spatial relations based on mathematical morphology, which constitutes a unifying formal framework dealing with different levels of granularity. Illustrative examples in recognition of image structures are given in Section 28.8.

28.2 Some Views on Space and Granularity in Spatial Reasoning

The issue of perception and representation of space and spatial relations and the issue of spatial reasoning have been addressed by researchers in many communities. This can be partly explained by the fact that spatial knowledge is foundational to common-sense knowledge. We summarize here some views on space that are inspiring computer scientists [5]. Although many philosophers have contributed to thinking about space and spatial concepts, we restrict the presentation to a few aspects in linguistics, human perception, and cognition, still far from exhaustivity.

28.2.1 Linguistics

Natural languages usually offer a rich variety of lexical terms for describing spatial location of entities. These terms are not only numerous; they also concern all lexical categories, such as nouns, verbs, adjectives, adverbs, and prepositions [6].

The domain of linguistics is a source of inspiration of many works on qualitative spatial information representation and qualitative spatial reasoning [7]. Modeling qualitative spatial relations strongly relies on the way these relations are expressed verbally. Several properties are exhibited, such as the asymmetry of some expressions, the non-bijective relation between language and spatial concepts (in particular for prepositions [6, 8, 9]), the interaction between distances and orientation, etc. [8, 10].

Another important characteristic of linguistic expressions is the imprecision attached to ternary (or more) spatial relations (for instance being ‘among’ people), but also to binary ones. Usually the context allows dealing with such expressions and the linguistic statements are generally clear and informative enough to prevent misunderstanding. A remarkable feature is that representation and communication are then achieved without using numbers [6].

Conversely, apparently precise statements (for instance, containing crisp numbers) should not always be understood as really precise, but rather as order of magnitudes. Let us consider for instance the sentence ‘Paris and Toulouse are at a distance of 700 km.’ The number 700 should not be considered as an exact value. It gives an idea of the distance, and its interpretation is subject to some considerations such as the areas of Paris and of Toulouse that are really concerned, the way to travel from one city to the other, etc.

Too precise statements can even become inefficient if they make the message too complex. This appears typically in the problem of route description for helping navigation and path finding. The example of giving directions in Venice is particularly eloquent [11].

Moreover, the way to describe spatial situations, as well as the vision and the representation of space, is not fixed and is likely to be modified depending on perceptual data and discourse situation [6]. In linguistic statements about space and distance, the geometrical terms of the language that are involved in these statements are usually not sufficient to get a clear meaning. The statement context is also of prime importance, as well as functional properties of the considered physical entities. This appears for instance in the use of prepositions, where for example the shape of an object influences the interpretation

of a preposition preceding the name of this object. In [6], three levels are therefore distinguished for analyzing and representing the meaning of spatial expressions:

- geometrical level, concerning the objective space;
- functional level, accounting for the properties of the entities described in the text and for the non geometrical relations;
- pragmatic level, including the underlying principles for a good communication.

Languages exhibit strong differences in their spatial terms. This concerns the way to partition the space, the terms describing motion events, and the preferred lexical categories. For instance French, as well as other Romance languages, shows a typological preference for the lexicalization of the path in the main verb. On the contrary in Germanic and Slavic languages, the path is rather encoded in satellites associated to the verb (particle or prefix) [12].

Another subdivision considers a linguistic expression as composed of a theme, a frame of reference, and a medium. The medium can typically be related to distance, quality, etc. Three levels are then distinguished [13]:

- thematic segmentation, involving topology and qualitative thresholds (for instance, *close to*, etc.), with a possible multiscale aspect;
- pseudo-analog representation, involving a metric;
- knowledge.

The multiscale aspect allows us to deal with different levels of granularity. This overcomes some of the limits of approaches which have a fixed granularity and cannot properly manage both large-scale and close-range information and of approaches which deal with infinitesimals but are faced with Zeno's paradox. For instance 'close to' has clearly a different meaning in case of close-range (for instance, objects in a room) or large-scale (such as distances between cities) information.

28.2.2 Human Perception

Let us consider the perception of distances, as a typical example. It is influenced by a number of factors, leading to different measures [7]:

- Purely spatial measures, in a geometric sense, give rise to 'metric distances,' and are related to intrinsic properties of the objects; it should be noted that these characteristics do not involve only purely geometrical distances, but also topological, size, shape properties of the objects.
- Temporal measures lead to distances expressed as travel time and can be considered of extrinsic type, as opposed to the previous class; this advocates for treating space and time together (which will not be done in this chapter).
- Economic measures, in terms of costs to be invested, are also of extrinsic type.
- Perceptual measures lead to distances of deictic type; they are related to an external point of view, which can be concrete or just a mental representation, which can be influenced by environmental features, by subjective considerations, leading to distances that are not necessarily symmetrical; the discourse situation also plays a role at this level, as mentioned above for the linguistic aspects.

As mentioned in [14, 15], the perception of distance between objects also depends on the presence or absence of other objects in the environment. If there are no other objects, the perception and human reasoning are mainly of geometrical type and distances are absolute. On the contrary when there are other objects, the perception of distance becomes relative. The size of the area and the frame of reference also play a crucial role in the perception of distances [7], in particular by defining the scale and the upper bound of the perceived distances. Perception is therefore not scale independent [16], while language is to a large extent scale independent [10]. Finally, attractiveness of the objects strongly affects the perception of proximity [15].

Most of these remarks apply to other types of spatial relations as well.

While images are common physical supports used for transmitting messages to the brain, the way the nervous system acts is complex and still not completely explained [17]. Perception is achieved through two complementary processes in the human visual system, a passive one and an active one. Both processes are highly conditioned by the context (grey level scale, geometry, perspective, etc.). The level of detail of the image representation plays also an important role. One of the explanations of the way objects are interpreted from visual percepts can be found in the Gestalt theory [17, 18], in which objects are supposed to be perceived as a whole, instead of a concatenation of their different parts, i.e., at a quite advanced level of granularity.

28.2.3 *Cognition*

Spatial reasoning has often to deal with both quantitative measures and qualitative statements. The advantage of quantitative measures lies in their absolute meaning. On the contrary, qualitative information is dependent on the context. However qualitative information is easily handled by humans and often more meaningful and eloquent, and therefore preferred [7, 15]. This raises the question of links between quantitative data and qualitative information, which is largely addressed in the fuzzy set community.

Let us illustrate the dependence on the context of qualitative information with the example of spatial distances. The meaning of ‘A is far from B’ depends on the relative size of A and B, on other scale factors, on the type of deduction and consequence we expect from this statement (for instance how can I go from A to B), etc. The translation of this statement in a quantitative value can lead to completely different results depending on this context. For instance, saying that my preferred bookstore is far from my house can be understood as a distance about a few hundred meters to a few kilometers, while saying that my cousin lives far can be understood as a distance about a few hundred to a few thousands kilometers. Such difficulties are related to the linguistic aspects mentioned before, as well as to the subjectivity of perception (in particular concerning the attractiveness of the objects).

The cognitive understanding of a spatial environment, in particular in large-scale spaces, is issued from two types of processes [7, 19, 20]:

- route knowledge acquisition, which consists in learning from sensorimotor experience (i.e., actual navigation) and implies an order information between visited landmarks,
- survey knowledge acquisition, from symbolic sources such as maps, leading to a global view (‘from above’) including global features and relations, which is independent of the order of landmarks.

During their development and growth, children first acquire route knowledge and have a local perception and representation of space. They acquire survey knowledge later, when they become able to perceive space from a more global perspective. The ability to reason about metrical properties of space comes at a very late stage of development [21–24]. The mapping between spatial words and basic spatial concepts does not appear to be universal and languages differ in their partitioning of the space. Children are able to distinguish between the different spatial categories of their own language at the age of 18–24 months. Such differences between languages can also be observed in the representation of motion events.

These two processes can be observed in neuroimaging [25]. For instance, a right hippocampal activation can be observed for both mental navigation and mental map tasks. A parahippocampal gyrus activation is additionally observed only for mental navigation, when route information and object landmarks have to be incorporated. Moreover, a mental simulation of a subject before reproducing a path from memory affects both maplike and routelike representations of the environment, and it allows the subject to better reproduce the path [26]. This is mostly observed for simple shapes, suggesting that the internal representation of space depends on geometric properties of the environment.

Experiments in case of sensory conflicts between visual and non-visual information have been performed in [27] and show that either visual or non-visual information can be used according to the task and to the sensory context. There are therefore at least two cognitive strategies of memory storage and retrieval for mental simulation of the same path.

As for the internal representation of space in the brain, a distinction is usually made between egocentric and allocentric representations [7, 28]. Although the notion of ‘map in the head’ has recognized

limitations as a cognitive theory, it is still quite popular and corresponds to the allocentric representations. It is important to note that the psychological space does not need to mirror the physical space.

As shown in [29], the egocentric route strategy needs the memory of the associated movements with landmarks and episodes (kinesthetic memory). Solving the Pythagoras' theorem from memory is possible using vestibular information, but requires converting an egocentric representation into an allocentric representation.

The mental representation is also combined with other factors in cognitive processes about space. For instance, questions such as 'where am I' can find different answers corresponding to [30]:

- autobiographical memory,
- semantic memory,
- stress and emotion,
- egocentric spatial representation.

Cognitive studies report that distance and direction are quite dissociated. On the contrary, as mentioned for the perception, from a cognitive point of view, time and space cannot be easily separated.

The importance of the frame of reference, highlighted in all domains, also has a cognitive flavor: cognitive studies have shown that multiple frames of reference are usually used and appear as necessary for understanding and navigating in a spatial environment [7, 31].

Changes of view point are also strongly involved in social interactions and are required in order to understand and memorize where others are glancing [29].

These cognitive concepts have been intensively used in several works in the modeling and conception of geographic information systems (GIS), where spatial information is the core [32, 33]. Let us just mention two examples. In [34, 35], a fuzzy cognitive map framework is introduced for GIS, inspired by the cognitive aspects of space and spatial relations. It aims at integrating quantitative and qualitative data, taking into account the fuzziness of relations, and at providing a decision support producing cognitive descriptions similar to the ones a human expert could derive and use. Another example is the geocognostics framework proposed in [22], which aims at integrating in a common framework both formal geometric representations of spatial information and formal representations of cognitive processes. The idea is to express views and trajectories in cognitive terms and then to reinterpret them geometrically.

Another field where cognitive aspects about space inspire the development of frameworks and systems is the domain of mobile robotics. The work by Kuipers is fundamental in this respect [31, 36, 37]. His spatial semantic hierarchy is a model of knowledge of large-scale space including both qualitative and quantitative representations and is strongly inspired by the properties of the human cognitive map. It aims at providing methods for robot exploration and map building. The hierarchy consists of sensory, control, causal, topological, and metrical levels.

Recently, a new approach was proposed in [38], called conceptual spaces. These spaces can be considered as a representation of cognitive systems, intermediate between the high-level symbolic representations and the subconceptual connectionist representations [39, 40]. They emphasize orders and measures, and a key notion is distances between concepts, leading to geometrical representations, but using quality dimensions. They offer a nice and natural way to model categories, to express similarities. Distances are therefore put to the fore in such spaces. Gärdenfors shows that 'a conceptual mode based on geometrical and topological representations deserves at least as much attention in cognitive science as the symbolic and the associationistic approaches,' and his book is therefore about the 'geometry of thoughts' [40].

28.2.4 Granularity

As seen in the previous paragraphs, granularity is involved more or less explicitly at many different levels and is an important aspect of space and reasoning on spatial concepts (see e.g., [7, 41]).

Summarizing, granularity is involved:

- in the objects or spatial entities and their descriptions (crisp well-defined, rough, fuzzy, abstract or symbolic), in particular, works in linguistics show that spatial considerations are not scale independent.

Object information is granular in the sense of Zadeh [3]; i.e., ‘data points within a granule have to be dealt with as a whole rather than individually’;

- in the types and expressions of spatial relations and queries; granularity is also closely related to the context, the notion of reference, which provide spatial relations and spatial reasoning with a strongly relative nature [9, 42];
- in the type of expected or potential result (truth value, number, fuzzy number, interval, distribution, etc.).

These three aspects are discussed in the following sections.

It is interesting to note that image interpretation is among the typical examples mentioned in [4] for highlighting the importance of the notion of information granule and the major role played in this domain by granularity. This naturally extends to spatial reasoning, for which images are often a useful support. Three main points underlined in [4] are quoted below:

- ‘information granules are the key components of knowledge representation and processing’;
- ‘the level of granularity becomes crucial to the problem description and solving’;
- ‘there is no universal level of granularity of information.’

As will be seen in the next sections, these three main features are also crucial in spatial reasoning, and may find a proper representation framework in fuzzy sets theory.

28.3 Most Detailed Level

In this section, we consider the most detailed level (highest degree of granularity).

At this level, the objects and spatial entities are precisely described, as sets of points (in \mathbb{Z}^n), or continuous shapes in \mathbb{R}^n . In particular, it is possible to state, for any point of space, whether or not it belongs to a spatial entity. This makes reasoning at point level possible, based on classical Euclidean geometry, and is the main approach in image processing and computer vision [43], or in computational geometry [44]. Typically in images, such entities are directly linked to visual percepts in images (pixels or voxels, groups of pixels, features). An important advantage of these representations with respect to the granularity issue relies on the numerous work on multiresolution and multiscale approaches developed in image processing and computer vision [45]. Here the granularity concerns the definition of image points or regions and naturally leads to multiscale reasoning.

The main difficulty when using such representations is to define semantically relevant objects. In terms of granularity, this problem amounts to establishing a correspondence between fine descriptions, as extracted from images, and a set of symbols, representing, at a high representation level, the semantics of the objects. This refers to the well-known semantic gap problem, defined as ‘the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation’ [46]. It is close to the problem of symbol grounding or anchoring addressed in artificial intelligence [47] and in robotics [48]. This problem also occurs when reasoning at different levels of granularity, and when objects, in a scale-space representation for instance, are not necessarily directly linked to hierarchical semantic concepts. This question is further addressed in Section 28.6.

Let us now consider spatial relations and spatial reasoning. Several usual spatial relations are well defined and can be computed precisely based on mathematical formulas. This is the case for instance for topological relations (such as intersection and adjacency) and for distances. When objects are precisely defined, these relations provide numbers or binary truth values.

For topological relations, answering questions such as ‘are two given spatial entities adjacent, are they intersecting?’ relies on simple computation on the objects (since they are perfectly known), and leads to all-or-nothing (binary) answers. An issue with these representations is that the result is highly sensitive to the definition of the objects. For instance, the adjacency relation can switch from true to false by changing

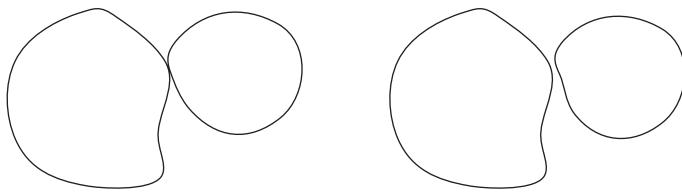


Figure 28.1 Sensitivity of crisp adjacency: small modifications in the shapes may completely change the adjacency relation and thus prevent a correct reasoning based on this relationship

only one point in the objects. Figure 28.1 illustrates this problem. Making this relation more robust calls for different definitions, for instance, in a fuzzy set framework [49], as discussed in Section 28.5.

For distance relations, several definitions exist, with different properties. The Hausdorff distance is a true distance, satisfying all properties of a distance function. Other definitions do not lead to true distances, such as the minimum (closest point) distance, or the average distance, but may be interesting in practice. All definitions, when applied on well defined objects, provide numerical results, i.e., numbers in \mathbb{R}^+ . The range of existing definitions raises the problem of the choice of the appropriate definition for a specific problem. A discussion on the properties and their usefulness in spatial reasoning in images can be found in [50]. Another issue concerns their behavior, leading to different robustness properties. For instance the Hausdorff and minimum distances are highly sensitive to the definition of the objects. On the contrary the average distance has a smoothing effect that makes it more robust.

At the most detailed level of granularity, questions such as ‘what is the distance between two given spatial entities, is the distance between two entities equal to 10 cm?’ can be answered precisely, leading to binary results. Although a high precision can be expected, this type of reasoning is strongly limited when it has to deal with precise statements with an actually imprecise meaning (such as the example of ‘700 km’ in Section 28.2), or when it has to incorporate other types of knowledge, expressed in a rougher way. Common-sense knowledge and reasoning often deal with expressions such as ‘the road is close to the railway,’ which requires abandoning precise numerical values.

Another limitation is that some relations deviate from the well-defined hypothesis. This is the case for relative direction relations (such as ‘left of’), or more complex relations such as ‘between,’ ‘along’ [51, 52]. Modeling these relations and reasoning on them, even for precisely defined objects, requires moving to another level of granularity. This is a typical example illustrating the interest and usefulness of intermediate and semi-qualitative representations, in particular using fuzzy sets, as already suggested in [53].

Summarizing, at the most detailed level, it is possible to reason on precisely defined objects and well defined spatial relations, in order to address precise queries and provide precise answers. However, spatial reasoning at this level has to face numerous difficulties, related to semantics, sensitivity, lack of robustness, and the de facto exclusion of several types of relations, knowledge, questions, reasoning processes, which cannot be dealt with in this restricted context.

28.4 Most Abstract Level

At the most abstract level, objects or spatial entities are considered as abstract regions, without references to points [41, 54]. This corresponds to the notion of granule in the sense of Zadeh [3, 55], since a spatial entity is then considered as a whole, and its points are not considered individually. They are typically represented as elements of a language, propositional terms, logical formulas in the qualitative spatial reasoning community, or as symbolic concepts in knowledge-based systems, GISs, and ontologies. Usually an exhaustive set of basic relations is defined, often expressed as operators, modalities, etc., from which more complex ones can be built, in particular using composition rules. Questions addressed at this level concern satisfiability, path consistency assessment, inference, prediction, diagnosis, interpretation,

which may be incorporated in complex reasoning schemes [56]. Several logics of space have been developed, which will not be detailed here (see e.g., [57–61]). A modal logic based on mathematical morphology, developed in [62], will be mentioned in Section 28.7 for its interesting links with other spatial relations models.

According to [63], ‘the challenge of qualitative spatial reasoning is to provide calculi which allow machine to represent and reason with spatial entities of higher dimension, without resorting to the traditional quantitative techniques prevalent in, for example, the computer graphics and the computer vision community.’ The nature of qualitative calculus is discussed in [64] and a general framework is proposed, which highlights the algebraic properties of all qualitative spatial calculi formalisms and puts the notion of weak representation to the fore.

As mentioned in Section 28.2, a large source of inspiration for knowledge representation and reasoning is found in the literature on linguistics and cognitive science (see e.g., [65]). Interestingly enough, spatial language is rather nonmetric (or metric information is digitized in a very rough way) but intensively uses directions, mainly three coordinate axes [10]. This advocates reasoning processes based on linguistic descriptions of a scene or a situation where distance plays almost no role. A remarkable feature of linguistic expressions is that representation and communication are then achieved without using numbers. The main relations dealt with in qualitative spatial reasoning are indeed of topological or directional nature. A review of the main approaches can be found in [41]. Let us just mention the most popular ones.

Part-whole relations and topological relations have been developed in the qualitative setting of mereology and mereotopology [66, 67], in particular within the famous framework of region connection calculus (RCC) [68]. A lattice of spatial relations is derived from parthood and connection predicates. Another approach, known as 9-intersections [69], partitions the space into three regions for each object (its boundary, its interior, and its complement), which constitutes the basis for computing relations.

As far as directional relations are concerned, qualitative representations are less developed than topological relations. Cardinal directions (i.e., north, south, east, west) are used in [70]. Other approaches are inspired by the temporal interval representations [71], and one of the most used representations (in particular in GIS) is two-dimensional strings [72] which use relations between the projections of the considered objects on two orthogonal axes and interval-based representations on each axis. Finally, let us mention the approach in [73] which represents the relative position of a point with respect to two other points as a 5×3 matrix based on a subdivision of the space into six sectors related to the two reference points.

Another class of approaches relies on ontologies. General aspects of an ontology of space are discussed in [41] and ontological questions about space are raised in [74] (primitive spatial entities, nature of the embedding space, types of computations allowed, modeling of multidimensional space).

As mentioned in [75], several ontological frameworks for describing space and spatial relations have been developed recently. In spatial cognition and linguistics, the OntoSpace¹ project aims at developing a cognitively based common-sense ontology for space. Some interesting works on spatial ontologies can also be found in GIS [76, 77], in object recognition in images or videos [78, 79], in robotics [80], or in medicine concerning the formalization of anatomical knowledge [81–84]. All these ontologies concentrate on the representation of spatial concepts according to the application domains. They do not provide an explicit and operational mathematical formalism for all the types of spatial concepts and spatial relations. For instance, in medicine, these ontologies are often restricted to concepts from the mereology theory [83]. These concepts are fundamental for spatial relations ontologies [85], and these ontologies are useful for qualitative and symbolic reasoning on topological relations, but there is still a gap to fill before using them for image interpretation, where more relations are required.

The main advantages of these various classes of approaches can be summarized as

- the compact representations they provide, while keeping a good level of expressiveness; (Actually the main emphasis in the qualitative spatial reasoning literature is on these representation issues.)

¹ <http://www.ontospace.uni-bremen.de/twiki/bin/view/Main/WebHome>.

- usually an exhaustive set of basic relations is defined, from which more complex ones can be built (in particular using composition tables);
- they benefit from the logical apparatus, enhancing the reasoning power, and from the important work on these aspects (including issues such as complexity analysis, path consistency, definition of maximal tractable subsets, etc.).

But their use for spatial reasoning in images is also limited. The main causes are

- There is no direct link with image information, and it is usually difficult to relate the formalized concepts to quantitative information, as extracted from the images. A few attempts have been done to use RCC for image region interpretation (e.g. [86]), assuming perfect low-level processing, which is still far from real applications based on image processing. (Applications are usually merely a motivation for the theoretical developments.)
- Along the same line, an important issue concerns the transition between the expressed knowledge (usually in a textual form) and a model that can be manipulated.
- Most of these approaches do not deal with hierarchical reasoning, although it is an important component of human spatial reasoning; some works toward this aim are worth being mentioned here [87, 88], since they explicitly consider granularity and multiresolution representations.
- Imprecision and uncertainty, which are important features of spatial information when reasoning at different levels of granularity, are usually not modeled. Let us however mention the interesting extension of RCC to deal with imprecise objects in the egg-yolk approach [89].
- A lot of work concerns topological relations, some approaches also deal with directional relations, but distances are seldom included. Some work on qualitative distances can be found, e.g., in [7], based on a given number of distinctions (such as close, far, etc.), this number depending on the required level of granularity.

28.5 Intermediate Level: Fuzzy Representations

Fuzzy representations constitute a good intermediate level between the most detailed one and the most abstract one. They allow covering a large range of granularity levels and bridge the gap between numeric and symbolic concepts. These features are illustrated in the following paragraphs, for the different aspects where granularity is involved, as mentioned in Section 28.2. Some of these considerations apply for interval-based or rough representations as well, but are not further described here.

28.5.1 Objects

Considering spatial entities as fuzzy objects is very interesting since it allows for representing explicitly spatial imprecision. A spatial fuzzy set is a fuzzy set defined on the image space, denoted by \mathcal{S} . Its membership function μ (defined from \mathcal{S} into $[0, 1]$) represents the imprecision on the spatial definition of the object (its position, size, shape, boundaries, etc.). For each point x of \mathcal{S} (pixel or voxel in digital two- or three-dimensional images), $\mu(x)$ represents the degree to which x belongs to the fuzzy object. Objects defined as classical crisp sets are but particular cases, for which μ takes only values 0 and 1. The imprecision may have very different origins: it can be already present in the observed phenomenon itself (gradual transition between two objects for instance), it can be related to the type of observation (resolution of the imaging device, imperfection of the reconstruction process, etc.), or to some processing applied to the observation (such as segmentation, registration). Such representations also allow coping with multiresolution and multiscale aspects. For instance the rougher granularity level of a low-resolution or a large-scale representation entails more imprecision, which is then directly represented using fuzzy sets. An advantage of fuzzy representations to face the problems related to digitization is that a gain in robustness is obtained, in particular in the computation of object properties [1, 90, 91].

28.5.2 Spatial Relations

As underlined in Section 28.3, several relations are intrinsically vague and cannot be properly modeled in a crisp way. Fuzzy representations overcome this limitation, and a larger variety of relations can be modeled in this framework, such as directional position, rough distance relations, etc. Another feature of fuzzy representations of spatial relations is that they provide a continuum from quantitative information to symbolic one, which allows reasoning at different levels of granularity in one common framework. Let us consider distances relations, as an illustrative example. We may want to reason with precise values, or at a very rough level (using linguistic values such as small and large for instance). Expressing distances as linguistic variables, with semantics of each value defined by fuzzy sets, is an elegant way to address these questions: different levels of granularity can be represented, depending on the number of linguistic values involved in the representation. Moreover, this approach separates the language domain (here, the choice of linguistic values of interest) and the quantitative scale (here \mathbb{R}^+), on which the fuzzy sets are defined. This separation is a fundamental aspect, in particular for reasoning purposes.

Typically, in spatial reasoning, questions and reasoning may concern:

1. the relations that are satisfied or not between two given objects (or satisfied to some degree);
2. the region of the space \mathcal{S} where a relation to one reference object is satisfied (to some degree).

Fuzzy sets are appropriate for answering both types of questions. The first type of question can be addressed by computing a relation between two given objects (whatever their representation and their level of granularity), this relation being precisely defined or not. This leads to different types of answers, as described next. Fuzzy sets are also useful for defining spatial representations of spatial relations in order to answer the second type of question. Some examples are given in Section 28.7.3.

Here again fuzziness allows more robustness in the computation of relations, depending on the granularity of the representation. Let us take the example of adjacency (see Figure 28.1). In digital spaces, this relation is highly sensitive, since in the binary case, the result can depend on one point only. Also the segmentation can induce errors that can completely change the relations. For instance two objects that are expected to be adjacent (for instance, if they are described as such in a model of the observed scene) can appear as not adjacent depending on the digitization and granularity level or if some even tiny errors occur during the segmentation. In the fuzzy case, the problem is much less crucial. Indeed, there is no more strict membership, the fuzziness allows dealing with some gradual transition between objects or between object and background, and relations then become a matter of degree, and are more robust to the above-mentioned problems.

28.5.3 Type of Result

While in the crisp case a relation between two objects is usually represented by a number (either 0/1 for an all-or-nothing relation, or a numerical value for a distance for instance), in the fuzzy case, several representations are possible. They can typically be intervals, for instance representing necessity and possibility degrees, fuzzy numbers, distributions.

Let us consider the example of distance relations (similar considerations hold for other types of spatial relations). While in the crisp case, a distance between two objects is always represented by a number in \mathbb{R}^+ , in the fuzzy case, several representations are possible:

- The most used representation of a distance between two fuzzy sets is as a number d , taking values in \mathbb{R}^+ , or more specifically in $[0, 1]$ for some of them, such as distances defined as dissimilarity measures.
- However, since we consider fuzzy sets, i.e., objects that are imprecisely defined, through membership functions μ and ν , we may expect that the distance between them is imprecise too [92, 93]. Then the distance is better represented as a fuzzy set, and more precisely as a fuzzy number.
- In [93], Rosenfeld defines two concepts to define distances as fuzzy numbers. One is distance density, denoted by $\delta(\mu, \nu)$, and the other distance distribution, denoted by $\Delta(\mu, \nu)$, both being fuzzy sets on

\mathbb{R}^+ . They are linked together by the following relation:

$$\Delta(\mu, \nu)(n) = \int_0^n \delta(\mu, \nu)(n') dn'. \quad (1)$$

While the distance distribution value $\Delta(\mu, \nu)(n)$ represents the degree to which the distance between μ and ν is less than n , the distance density value $\delta(\mu, \nu)(n)$ represents the degree to which the distance is equal to n .

- Histograms of distances inspired from angle histograms, carrying a complete information about distance relations but at the price of a heavier representation, have been introduced in [5].
- The concept of distance can be represented as a linguistic variable. This assumes a granulation of the set of possible distance values into symbolic classes such as ‘close to,’ ‘far from,’ etc., each of these classes having a semantics defined as a fuzzy set on \mathbb{R}^+ . This approach has been drawn, e.g., in [94], where the relation ‘far from’ is defined as a decreasing function of the average distance between both sets.

These different representations correspond to different granularity levels and illustrate the large range of granularities that can be accounted for.

In all these examples, the semantics of the distance is assumed to involve two given objects, thus dealing with the first type of questions mentioned above. As for the second type of question, expressing distance constraints to a reference object, a spatial representation as a fuzzy region of S is preferred [5, 95]. Such constraints are often expressed as imprecise statements or in linguistic terms, which reinforces the usefulness of fuzzy modeling. The membership function value at each point of this region represents the degree to which the relation is satisfied at this point.

Let us now consider relative directions. Here granularity associated with fuzzy representations can be exploited in a very relevant way: using only one dominant direction, or only cardinal directions in a crisp way is very limiting; on the contrary, representing a few directions using linguistic variables, as done, e.g., in [96], exploits the features of both granularity to adapt the number of directions and linguistic values to the needs of the application, and fuzzy sets, to provide flexibility in the semantics of each direction.

28.5.4 Reasoning

The separation between the symbolic (linguistic) level and the numerical level, as highlighted above, has several advantages for reasoning purposes:

- It allows general reasoning at symbolic level (using for instance formal logics such as in the approaches described in Section 28.4, rule-based systems, ontological reasoning tools such as Racer, Fact++ or Pellet, etc.)
- It allows more specific reasoning as well, using the fuzzy representations of objects and spatial relations and the associated tools of fuzzy logics.
- It also establishes the necessary links between the symbolic level and the quantitative level: the fuzzy set approach such bridges the gap between numerical and symbolic concepts at both relation level and reasoning level.

These different levels of reasoning are very useful for dealing with concrete problems. For instance we may have to reason based on general rules and knowledge, but also based on specific or particular facts or observations. We may want a fine grain in the type of information to be handled or the type of expected result, but without being obliged to do so. Precise information may be not reliable enough, and we may then want to go to a somewhat more general level, or rougher level of granularity, in order to still be able to draw some partial conclusions. All these situations find a suitable reasoning framework in the fuzzy approaches.

Another powerful aspect of fuzzy representations concerns fusion and decision, which are very important for spatial reasoning (e.g., fusion of several spatial relations to assess the position of an object, very useful for image interpretation or structural recognition). Fusion and decision are extensively developed in the fuzzy sets community and benefit from a large variety of fusion operators, able to cope with a wide spectrum of situations [97–99].

28.5.5 Overview of Fuzzy Representations of Spatial Relations

Several teams have proposed fuzzy representations of spatial relations. We just briefly mention them here. See, e.g., [100] for a synthesis of the existing fuzzy definitions of spatial relations. Definitions based on mathematical morphology are further detailed in Section 28.7.

Fuzzy inclusion has been defined by several authors [94, 101–103], based on basic axioms a fuzzy inclusion should satisfy and on set theoretical considerations. Other approaches exploit some links with fuzzy entropy [103, 104], or derive inclusion from fuzzy implication [103, 105, 106].

Several topological relations are derived from basic concepts such as local neighborhood. These notions have been extended to fuzzy sets, leading to notions of fuzzy neighborhood [49, 107], fuzzy connectedness [108, 109], from which topological relations between fuzzy sets are derived. In particular, fuzzy adjacency has been defined from a notion of visibility [110], or using the notion of contours, frontiers, and neighborhood [49, 107].

A lot of work in the fuzzy sets community concerns similarities and distances. Classifications have been proposed, such as in [111], but not with the point of view of spatial reasoning. A review can be found in [5], where the spatial aspects are discussed. Two classes of fuzzy distances can be distinguished. In the first class, the distance between two fuzzy sets is obtained by comparing membership functions, and does not really take distances in the spatial domain into account. The second class of methods tries to include the spatial distance d_S (Euclidean distance in S for instance) in the distance between fuzzy sets, hence being more interesting for spatial reasoning purposes. Among these distances, some of them are based on a geometrical approach [92, 93], or on weighting procedures, using membership values as weights in classical distance formulations [93]. Some authors define a fuzzy distance as a fuzzy set on \mathbb{R}^+ [93, 112], in particular a lot of work can be found on fuzzy Hausdorff distance [92, 111, 113–116]. A morphological approach has been proposed in [50, 117] (see Section 28.7 for more details on this approach).

As a typical example of relations that are not well defined, even if the objects are crisp, directional relations have been defined in a fuzzy set framework using several approaches: a linguistic definition of directions is proposed in [118, 119], which provides the semantics of these directions in terms of fuzzy sets, to which information extracted from the objects can then be compared. Several methods rely on this principle: centroid method [94, 118], compatibility approach [96, 119], aggregation [94, 118], histogram of forces [120]. Other methods rely on the projection of the objects on a line defined according to the direction of interest [121]. A morphological approach was also proposed in [51, 122, 123] (see Section 28.7). Angle-based approaches can be extended to define more complex relations, such as ‘surround’ [96]. Slightly different approaches for visual surroundness and topological surroundness have been proposed in [110]. A complete discussion and comparison can be found in [124].

A few more complex relations can also be found in the literature, such as ‘surround,’ already mentioned, ‘between’ [52], ‘along’ [125]. The problem becomes even more difficult for such relations since their semantics highly depend on the context in which the objects are embedded, on the objects themselves, on their shape, and on the type of question to be addressed.

A question that is often raised by these fuzzy definitions is the one of computation complexity. Although fast algorithms can sometimes be designed, as proposed for instance in [51] for relative direction, using different granularities, expressed for instance as simple subsampling procedures, can also be very helpful. Let us illustrate this idea on a simple example. Figure 28.2 illustrates a slice of a three-dimensional segmentation of the lungs in a CT image. The region ‘between the lungs’ has been computed using fuzzy dilations based on the histograms of angles, as proposed in [52]. Assessing the degree to which the heart is between the lungs leads to the value 0.973. By subsampling the images by a factor 3 in all directions

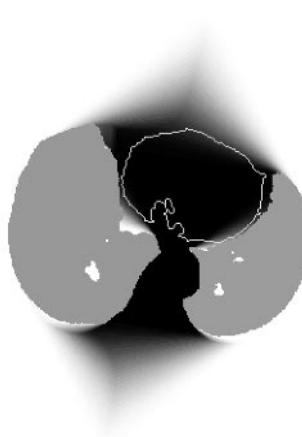


Figure 28.2 Fuzzy region ‘between’ the lungs, superimposed on an axial slice of the segmented lungs. The contours of the heart are superimposed too

and recomputing the relation, a degree equal to 0.968 is obtained, i.e., a very similar result, but in a much shorter time. This shows the robustness of the proposed definition with respect to the granularity level, and hence the interest of exploiting different granularity levels in order to achieve better computation times. Figure 28.3 shows the histogram of angles computed for two different granularity levels. Only slight modifications can be observed, which have no consequence on the final results.

28.6 Fuzzy Representations as an Answer to the Semantic Gap Problem

The semantic gap problem and communication issues have already been mentioned at several places in the previous sections. Let us summarize here the advantages of fuzzy representations from this point of view:

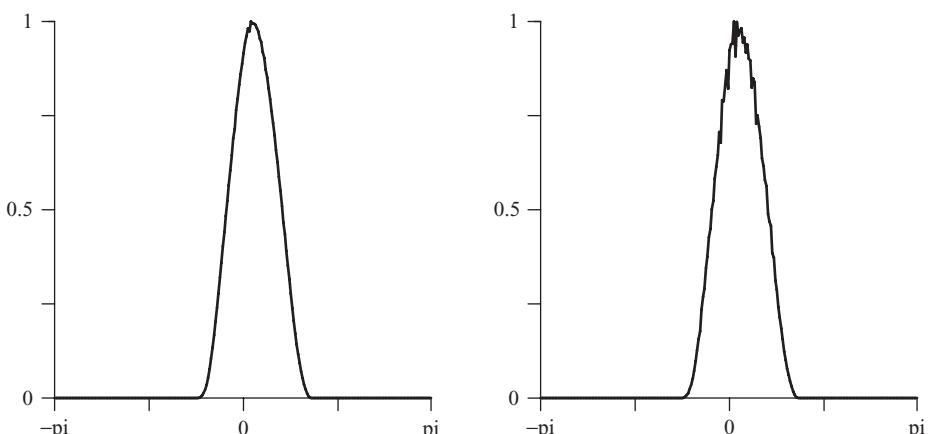


Figure 28.3 Histogram of angles between the two lungs of Figure 28.2, with the original resolution, and with a rougher level of granularity (subsampling by a factor 3 in all directions)

- They allow representing the imprecision which is inherent to the definition of a concept (see Section 28.2); for instance, the concept ‘close to’ is intrinsically vague and imprecise, and its semantics depends on the context in which objects are embedded, on the scale of the objects and of their environment, on the reference system and the observer.
- They allow managing imprecision related to the expert knowledge in the concerned domain (for instance, they can cope with apparently precise statements that are actually not, as the example of ‘700 km’ distance in Section 28.2).
- They constitute an adequate framework for knowledge representation and reasoning, reducing the semantic gap between symbolic concepts and numerical information.

All these aspects make fuzzy representations a powerful communication tool between different systems of information granules [4]. In particular, the multiscale aspect and the different levels of granularity necessary to cope with both large-scale and close-range information (see Section 28.2) are naturally taken into account using fuzzy representations.

For instance they allow answering linguistic queries on numerical objects on which numbers can be evaluated. These queries may involve linguistic terms, which need fuzzy semantics to make them adapted to a specific domain or context. These fuzzy semantics provide a natural link with numerical objects. In a recent work, we have shown that fuzzy representations are indeed appropriate to provide semantics to symbolic and ontological approaches [126]. This leads to operational tools in concrete applications, for instance, for image interpretation. As mentioned in Section 28.2, perception is not scale independent, while language is (to some extent). This gap is bridged by these fuzzy representations, since they establish a clear link between concepts expressed in natural language and percepts characterizing objects (for instance, extracted from images).

Having these links between percepts and concepts is also a convenient way to have a common framework to model both route knowledge acquisition (based on percepts) and survey knowledge acquisition (based on models and concepts).

Let us further detail the example of image processing. Fuzzy set theory finds in spatial information processing a growing application domain. This may be explained not only by its ability to model the inherent imprecision of such information (such as in image processing, vision, mobile robotics ...) together with expert knowledge, but also by the large and powerful toolbox it offers for dealing with spatial information under imprecision. This is in particular highlighted when spatial structures or objects are directly represented by fuzzy sets. If even less information is available, we may have to reason about space in a purely qualitative way, and the symbolic setting is then more appropriate. In artificial intelligence, mainly symbolic representations are developed and several works addressed the question of qualitative spatial reasoning (see Section 28.4).

Limitations of purely qualitative spatial reasoning have already been stressed in [56], as well as the interest of adding semiquantitative extension to qualitative value (as done in the fuzzy set theory for linguistic variables [55, 98]) for deriving useful and practical conclusions (as for recognition). Purely quantitative representations are limited in the case of imprecise statements, and of knowledge expressed in linguistic terms, as mentioned in Section 28.3), hence the interest of fuzzy sets. As another advantage of fuzzy representations, both quantitative and qualitative knowledge can be integrated, using semiquantitative (or semiqualitative) interpretation of fuzzy sets. These representations can also cope with different levels of granularity of the information, from a purely symbolic level, to a very precise quantitative one. As already mentioned in [53], this allows us to provide a computational representation and interpretation of imprecise spatial constraints, expressed in a linguistic way, possibly including quantitative knowledge. Therefore the fuzzy set framework appears as a central one in this context.

Let us illustrate the link between semantic level and image on a simple example, displayed in Figure 28.4. Let us assume that a building has been found in an aerial image and that some knowledge base states that some interesting object may lay at an approximate distance (roughly between 10 and 20 m) from this building. This symbolic knowledge can be transposed as a fuzzy interval on the real line, expressing the semantics of this approximate interval. Two structuring elements can then be built to express this knowledge, now in the spatial domain (see also Section 28.7.3). Dilating the building image by these two fuzzy structuring elements provides regions of space representing respectively the

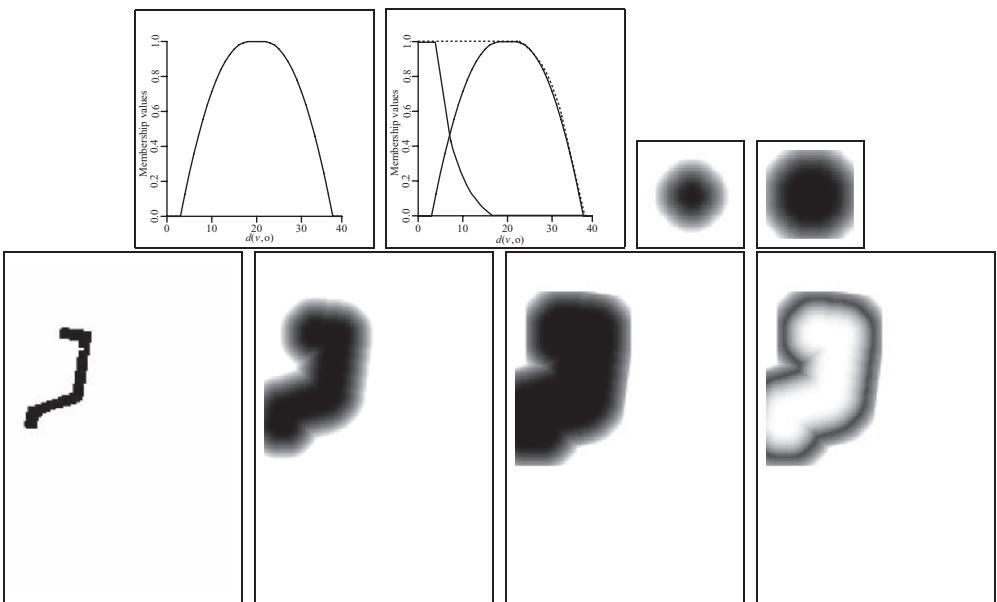


Figure 28.4 First line: fuzzy interval providing the semantics of the symbolic knowledge ‘about 10–20 m,’ translation of this knowledge into two structuring elements in the spatial domain. Second line: reference object (a building extracted from an aerial image), two fuzzy dilations, set difference representing the region of interest matching the knowledge

region with is too close to the object to be acceptable given the symbolic information (‘less than about 10 m’), and the region with includes all potential regions (‘less than about 20 m’). The set difference between both regions represents, in the spatial domain, the translation of the symbolic knowledge as a fuzzy region of interest matching this knowledge, i.e., in which the other object can be searched for. This example shows how fuzzy representations establish the correspondence between symbolic information and spatial information, hence contributing to fill the semantic gap.

28.7 Mathematical Morphology as a Unifying Formal Framework

One of the powerful features of mathematical morphology lies in its strong algebraic structure, that finds equivalents in set theoretical terms, fuzzy sets theory and logics. Moreover this theory is able to deal with local information, based on the concept of structuring element, but also with more global and structural information since several spatial relations can be expressed in terms of morphological operations (mainly dilations). The aim of this section (to a large part reproduced from [127]) is to show that the framework of mathematical morphology allows representing in a unified way spatial relations in various settings (and at different levels of granularity): a purely quantitative one if objects are precisely defined, a semiquantitative one if objects are imprecise and represented as spatial fuzzy sets, and a qualitative one, for reasoning in a logical framework about space. The proposed framework, briefly presented in Section 28.7.1, allows us to address three questions. We first consider the problem of defining and computing spatial relations between two objects, in both the crisp and fuzzy cases (Section 28.7.2), answering the first type of question raised in Section 28.5. Then in Section 28.7.3 we propose a way to represent spatial knowledge in the spatial domain, answering the second type of question. Finally in Section 28.7.4 we show that spatial relations can be expressed in the framework of normal modal logics, using morphological operations applied on logical formulas. This can be useful for symbolic (purely qualitative) spatial reasoning.

28.7.1 Basic Morphological Operations, Fuzzy and Logical Extensions

Classical Morphology

Let us first recall the definitions of dilation and erosion of a set X by a structuring element B in a space \mathcal{S} (e.g., \mathbb{R}^n , or \mathbb{Z}^n for discrete spaces like images), denoted respectively by $D_B(X)$ and $E_B(X)$ [128]:

$$D_B(X) = \{x \in \mathcal{S} \mid B_x \cap X \neq \emptyset\}, \quad (2)$$

$$E_B(X) = \{x \in \mathcal{S} \mid B_x \subseteq X\}, \quad (3)$$

where B_x denotes the translation of B at point x . In these equations, B defines a neighborhood that is considered at each point. It can also be seen as a relation between points. From these two fundamental operations, a lot of others can be built [128].

Fuzzy Mathematical Morphology

Several definitions of mathematical morphology on fuzzy sets with fuzzy structuring elements have been proposed in the literature (see e.g., [101, 129, 130]). Here we use the approach using t -norms and t -conorms as fuzzy intersection and fuzzy union. However, what follows applies as well if other definitions are used. Dilation and erosion of a fuzzy set μ by a fuzzy structuring element v , both defined in a space \mathcal{S} , are respectively defined as:

$$D_v(\mu)(x) = \sup_{y \in \mathcal{S}} t[v(y - x), \mu(y)], \quad (4)$$

$$E_v(\mu)(x) = \inf_{y \in \mathcal{S}} T[c(v(y - x)), \mu(y)], \quad (5)$$

where t is a t -norm, c a fuzzy complementation, and T is the t -conorm associated to t with respect to c . These definitions guarantee that most properties of morphological operators are preserved [101, 131].

Morphologics

Now, we express morphological operations in a symbolic framework, using logical formulas. Let us consider a language generated by a finite set of propositional symbols and the usual connectives. Kripke's semantics is used. The set of all worlds is denoted by Ω . The set of worlds where a formula φ is satisfied is $Mod(\varphi) = \{\omega \in \Omega \mid \omega \models \varphi\}$.

The underlying idea for constructing morphological operations on logical formulas is to consider set interpretations of formulas and worlds. Since in classical propositional logics, the set of formulas is isomorphic to 2^Ω , up to the logical equivalence, we can identify φ with $Mod(\varphi)$, and then apply set-theoretic morphological operations. We recall that $Mod(\varphi \vee \psi) = Mod(\varphi) \cup Mod(\psi)$, $Mod(\varphi \wedge \psi) = Mod(\varphi) \cap Mod(\psi)$, and $Mod(\varphi) \subseteq Mod(\psi)$ iff $\varphi \models \psi$.

Using these equivalences, dilation and erosion of a formula φ are defined as [132]:

$$Mod(D_B(\varphi)) = \{\omega \in \Omega \mid B(\omega) \cap Mod(\varphi) \neq \emptyset\}, \quad (6)$$

$$Mod(E_B(\varphi)) = \{\omega \in \Omega \mid B(\omega) \models \varphi\}, \quad (7)$$

where $B(\omega) \models \varphi$ means $\forall \omega' \in B(\omega), \omega' \models \varphi$.

The structuring element B represents a relation between worlds and defines a ‘neighborhood’ of worlds. It can be for instance defined as a ball of a distance between worlds [133]. The condition for dilation expresses that the set of worlds in relation to ω should be consistent with φ ; i.e., $\exists \omega' \in B(\omega), \omega' \models \varphi$. The condition for erosion is stronger and expresses that φ should be satisfied in all worlds in relation to ω .

Now we consider the framework of normal modal logics [134] and use an accessibility relation as relation between worlds. We define an accessibility relation from any structuring element B (or the converse) as: $R(\omega, \omega')$ iff $\omega' \in B(\omega)$. Let us now consider the two modal operators \square and \diamond defined from

the accessibility relation as [134]:

$$\mathcal{M}, \omega \models \Box\varphi \text{ iff } \forall \omega' \in \Omega, R(\omega, \omega') \Rightarrow \mathcal{M}, \omega' \models \varphi, \quad (8)$$

$$\mathcal{M}, \omega \models \Diamond\varphi \text{ iff } \exists \omega' \in \Omega, R(\omega, \omega') \text{ and } \mathcal{M}, \omega' \models \varphi, \quad (9)$$

where \mathcal{M} denotes a standard model related to R . Equation (8) can be rewritten as

$$\omega \models \Box\varphi \Leftrightarrow B(\omega) \models \varphi, \quad (10)$$

which exactly corresponds to the definition of erosion of a formula, and equation (9) can be rewritten as

$$\omega \models \Diamond\varphi \Leftrightarrow B(\omega) \cap Mod(\varphi) \neq \emptyset, \quad (11)$$

which exactly corresponds to a dilation. This shows that we can define modal operators derived from an accessibility relation as erosion and dilation with a structuring element:

$$\Box\varphi \equiv E_B(\varphi), \quad (12)$$

$$\Diamond\varphi \equiv D_B(\varphi). \quad (13)$$

The modal logic constructed from erosion and dilation has a number of theorems and rules of inference, detailed in [62, 135], which increase its reasoning power. All these definitions and properties extend to the fuzzy case, if we consider fuzzy formulas, for which $Mod(\varphi)$ is a fuzzy set of Ω . A fuzzy structuring element can be interpreted as a fuzzy relation between worlds. This extension is useful for expressing intrinsically vague spatial relations such as directional relative position.

28.7.2 Computing Spatial Relations from Mathematical Morphology: Quantitative and Semiquantitative Setting

Set Relations

Computing set relations, like inclusion, intersection, etc., if the objects are precisely defined does not call for specific developments. If the objects are imprecise, stating if they intersect or not, or if one is included in the other, becomes a matter of degree. A degree of inclusion can be defined as an infimum of a t -conorm (as for erosion). A degree of intersection μ_{int} can be defined using a supremum of a t -norm (as for fuzzy dilation) or using the fuzzy volume of the t -norm in order to take more spatial information into account. The degree of non-intersection is then simply defined by $\mu_{-int} = 1 - \mu_{int}$. The interpretations in terms of erosion and dilation allow including set relations in the same mathematical morphology framework as the other relations.

Adjacency

For any two subsets X and Y in the digital space \mathbb{Z}^n , the adjacency of X and Y can be expressed, in terms of morphological dilation, as $X \cap Y = \emptyset$ and $D_B(X) \cap Y \neq \emptyset, D_B(Y) \cap X \neq \emptyset$, where B denotes the elementary structuring element associated to the chosen digital connectivity. This structuring element is usually symmetrical, which means that the two conditions $D_B(X) \cap Y \neq \emptyset$ and $D_B(Y) \cap X \neq \emptyset$ are equivalent, so only one needs to be checked.

Adjacency between fuzzy sets can be defined by translating this expression into fuzzy terms, by using fuzzy dilation. The binary concept becomes then a degree of adjacency between fuzzy sets μ and ν :

$$\mu_{adj}(\mu, \nu) = t[\mu_{-int}(\mu, \nu), \mu_{int}[D_B(\mu), \nu]]. \quad (14)$$

This definition represents a conjunctive combination (using a t -norm t) of a degree of non-intersection μ_{-int} between μ and ν and a degree of intersection μ_{int} between one fuzzy set and the dilation of the other. This definition is symmetrical, reduces to the binary definition if μ, ν and B are binary, and is invariant with respect to geometrical transformations.

Distances

Mathematical morphology allows defining distances between fuzzy sets that combine spatial information and membership comparison. In the binary case, there exist strong links between mathematical morphology (in particular dilation) and distances (from a point to a set, and several distances between two sets), and this can also be exploited in the fuzzy case. The advantage is that distances are then expressed in set theoretical terms and are therefore easier to extend with nice properties than usual analytical expressions. Here we present the case of Hausdorff distance. The binary equation defining the Hausdorff distance $d_H(X, Y) = \max[\sup_{x \in X} d(x, Y), \sup_{y \in Y} d(y, X)]$ can be expressed in morphological terms as $d_H(X, Y) = \inf\{n, X \subseteq D^n(Y) \text{ and } Y \subseteq D^n(X)\}$.

A distance distribution, expressing the degree to which the distance between μ and μ' is less than n is obtained by translating this equation into fuzzy terms:

$$\Delta_H(\mu, \mu')(n) = t \left[\inf_{x \in S} T[D_v^n(\mu)(x), c(\mu'(x))], \inf_{x \in S} T[D_v^n(\mu')(x), c(\mu(x))] \right], \quad (15)$$

where c is a complementation, t a t -norm, and T a t -conorm.

A distance density, expressing the degree to which the distance is equal to n , can be derived implicitly from this distance distribution. A direct definition of a distance density can be obtained from $d_H(X, Y) = 0 \Leftrightarrow X = Y$, and for $n > 0$ $d_H(X, Y) = n \Leftrightarrow X \subseteq D^n(Y)$ and $Y \subseteq D^n(X)$ and $(X \not\subseteq D^{n-1}(Y) \text{ or } Y \not\subseteq D^{n-1}(X))$. Translating these equations leads to a definition of the Hausdorff distance between two fuzzy sets μ and μ' as a fuzzy number:

$$\delta_H(\mu, \mu')(0) = t \left[\inf_{x \in S} T[\mu(x), c(\mu'(x))], \inf_{x \in S} T[\mu'(x), c(\mu(x))] \right], \quad (16)$$

$$\begin{aligned} \delta_H(\mu, \mu')(n) = t \left[\inf_{x \in S} T[D_v^n(\mu)(x), c(\mu'(x))], \inf_{x \in S} T[D_v^n(\mu')(x), c(\mu(x))] \right], \\ T \left(\sup_{x \in S} t[\mu(x), c(D_v^{n-1}(\mu')(x))], \sup_{x \in S} t[\mu'(x), c(D_v^{n-1}(\mu(x)))] \right). \end{aligned} \quad (17)$$

The obtained distance is positive (the support of this fuzzy number is included in \mathbb{R}^+). It is symmetrical with respect to μ and μ' . The separability property (i.e., $d(\mu, \nu) = 0 \Leftrightarrow \mu = \nu$) is not always satisfied. However, we have $\delta_H(\mu, \mu')(0) = 1$ implies $\mu = \mu'$ for T being the bounded sum ($T(a, b) = \min(1, a + b)$), while it implies μ and μ' crisp and equal for $T = \max$. The triangular inequality is not satisfied in general.

Directional Relative Position from Conditional Fuzzy Dilation

As mentioned in Section 28.5, because of the inherent vagueness of directional relations, they may find a better understanding in the framework of fuzzy sets, as fuzzy relations, even for crisp objects. The approach summarized here relies on a fuzzy dilation that provides a map (or fuzzy landscape) where the membership value of each point represents the satisfaction degree of the relation to the reference object. This approach has interesting features: it works directly in the image space, without reducing the objects to points or histograms, and it takes the object shape into account.

We consider a (possibly fuzzy) object R in the space S , and denote by $\mu_\alpha(R)$ the fuzzy subset of S such that points of areas which satisfy to a high degree the relation ‘to be in the direction \vec{u}_α with respect to object R ’ have high membership values, where \vec{u}_α is a vector making an angle α with respect to a reference axis. We express $\mu_\alpha(R)$ as the fuzzy dilation of μ_R by v , where v is a fuzzy structuring element depending on α : $\mu_\alpha(R) = D_v(\mu_R)$, where μ_R is the membership function of the reference object R . This definition applies both to crisp and fuzzy objects and behaves well even in case of objects with highly concave shape. In polar coordinates (but this extends to three-dimensional as well), v is defined by²: $v(\rho, \theta) = f(\theta - \alpha)$ and $v(0, \theta) = 1$, where $\theta - \alpha$ is defined modulo π and f is a decreasing function; e.g., $f(\beta) = \max[0, \cos \beta]^2$ for $\beta \in [0, \pi]$.

² This definition of v is discontinuous at the origin. A continuous function could be obtained by modeling the fact that the direction of a point or of an object closed to the origin is imprecise.

Once we have defined $\mu_\alpha(R)$, we can use it to define the degree to which a given object A is in direction \vec{u}_α with respect to R . Let us denote by μ_A the membership function of the object A . The evaluation of relative position of A with respect to R is given by a function of $\mu_\alpha(R)(x)$ and $\mu_A(x)$ for all x in S . The histogram of $\mu_\alpha(R)$ conditionally to μ_A is such a function. A summary of the contained information could be more useful in practice, and an appropriate tool for this is the fuzzy pattern matching approach [136]: the matching between two possibility distributions is summarized by two numbers, a necessity degree N (a pessimistic evaluation) and a possibility degree Π (an optimistic evaluation), as often used in the fuzzy set community. The possibility corresponds to a degree of intersection between the fuzzy sets A and $\mu_\alpha(R)$, while the necessity corresponds to a degree of inclusion of A in $\mu_\alpha(R)$. These operations can also be interpreted in terms of fuzzy mathematical morphology, since Π corresponds to a dilation, while N corresponds to an erosion.

Some More Complex Relations

This idea of directional dilation can also be exploited for defining more complex relations. For instance, it allows defining the region ‘between’ two (possibly fuzzy) objects, as a fuzzy region, and assessing the degree to which a third object is between them [52]. Measures on the fuzzy ‘between’ region can also be defined to assess to which degrees the two objects are ‘along’ each other [125].

28.7.3 Spatial Representations of Spatial Relations

Now we address a second type of problem, and given a reference object, we define a spatial fuzzy set that represents the region of space where some relation to this reference object is satisfied (to some degree). The advantage of these representations is that they map all types of spatial knowledge in the same space, which allows for their fusion and for spatial reasoning. (This occurs typically in model-based pattern recognition, where heterogeneous knowledge has to be gathered to guide the recognition.) This constitutes a new way to represent spatial knowledge in the spatial domain [137].

For each piece of knowledge, we consider its ‘natural expression,’ i.e., the usual form in which it is given or available, and translate it into a spatial fuzzy set in S having different semantics depending on the type of information (on objects, spatial imprecision, relations to other objects, etc.).

The numerical representation of membership values assumes that we can assign numbers that represent degrees of satisfaction of a relation for instance. These numbers can be derived from prior knowledge or learned from examples, but usually there remain some quite arbitrary choices. However, we have to keep in mind that mostly the ranking is important, not the individual numerical values.

Set Relations

Set relations specify if areas where other objects can be localized are forbidden or possible. The corresponding region of interest has a binary membership function (1 in authorized portions of the space, 0 elsewhere). This extends to the fuzzy case as: $\mu_{\text{set}}(x) = t[\mu_{O^{\text{in}}}(x), 1 - \mu_{O^{\text{out}}}(x)]$, where t is a t -norm, which expresses a conjunction between inclusion constraint in the objects O^{in} and exclusion constraint from the objects O^{out} .

Other Topological Relations

Other topological relations (adjacency, etc.) can be treated in a similar way and involve morphological operators. For instance, an object that is a non tangential proper part of μ has to be searched in $E_v(\mu)$.

Distances

Again, morphological expressions of distances, as detailed in Section 28.7.2, directly lead to spatial representations of knowledge about distances. Let us assume that we want to determine B , subject to satisfy some distance relation with an object A . According to the algebraic expressions of distances, the dilation of A is an adequate tool for this. For example, if knowledge expresses that $d(A, B) \geq n$, then B should be looked for in $D^{n-1}(A)^C$. Or, if knowledge expresses that B should lay between a distance n_1 and a distance n_2 of A ; i.e., the minimum distance should be greater than n_1 and the maximum distance should be less than n_2 , then the possible domain for B is reduced to $D^{n_2}(A) \setminus D^{n_1-1}(A)$.

In cases where imprecision has to be taken into account, fuzzy dilations are used, with the corresponding equivalences with fuzzy distances. The extension to approximate distances calls for fuzzy structuring elements. We define them through their membership function ν on \mathcal{S} , with a spherical symmetry, where ν only depends on the distance to the center of the structuring element and corresponds to the knowledge expression, as a fuzzy interval for instance [138] (see the example in Figure 28.4).

Relative Directional Position

The definition of directional position between two sets described in Section 28.7.2 relies directly on a spatial representation of the degree of satisfaction of the relation to the reference object. Therefore the first step of the proposed approach directly provides the desired representation as the fuzzy set $\mu_\alpha(A)$ in \mathcal{S} .

28.7.4 Symbolic Representations of Spatial Relations

In this section, we use the logical framework presented in Section 28.7.1. For spatial reasoning, interpretations can represent spatial entities, like regions of space. Formulas then represent combinations of such entities, and define regions, objects, etc., which may be not connected. For instance, if a formula φ is a symbolic representation of a region X of the space, it can be interpreted for instance as ‘the object we are looking at is in X .’ In an epistemic interpretation, it could represent the belief of an agent that the object is in X . The interest of such representations is also to deal with any kind of spatial entities, without referring to points. If φ represents some knowledge or belief about a region X of the space, then $\square\varphi$ represents a restriction of X . If we are looking at an object in X , then $\square\varphi$ is a necessary region for this object. Similarly, $\diamond\varphi$ represents an extension of X , and a possible region for the object.

Topological Relations

Let us first consider topological relations, and two formulas φ and ψ representing two regions X and Y of the space. Note that all what follows holds in both crisp and fuzzy cases. Simple topological relations such as inclusion, exclusion, intersection do not call for more operators than the standard ones of propositional logic. But other relations such that X is a tangential part of Y can benefit from the morphological modal operators. Such a relation can be expressed as $\varphi \rightarrow \psi$ and $\diamond\varphi \wedge \neg\psi$ consistent. Indeed, if X is a tangential part of Y , it is included in Y but its dilation is not. If we also want X to be a proper part, we have to add the condition $\neg\varphi \wedge \psi$ consistent.

Let us now consider adjacency (or external connection). Saying that X is adjacent to Y means that they do not intersect and as soon as one region is dilated, it intersects the other. In symbolic terms, this relation can be expressed as $\varphi \wedge \psi$ inconsistent and $\diamond\varphi \wedge \psi$ consistent and $\varphi \wedge \diamond\psi$ consistent.

It is interesting to link these types of representations with the ones developed in the community of mereotopology, where such relations are defined respectively from parthood and connection predicates [66, 68]. Interestingly enough, erosion is defined from inclusion (i.e., a parthood relation) and dilation from intersection (i.e., a connection relation). Some axioms of these domains can be expressed in terms of dilation. For instance from a parthood postulate $P(X, Y)$ between two spatial entities X and Y and from dilation, tangential proper part can be defined as $TPP(X, Y) = P(X, Y) \wedge \neg P(Y, X) \wedge \neg P(D(X), Y)$.

Distances

Again we use expressions of minimum and Hausdorff distances in terms of morphological dilations. The translation into a logical formalism is straightforward. Expressions like $d_{\min}(X, Y) \leq n$ translate into $\diamond^n\varphi \wedge \psi$ consistent and $\diamond^n\psi \wedge \varphi$ consistent. Similarly for Hausdorff distance, we translate $d_H(X, Y) = n$ by $(\forall m < n, \psi \wedge \neg \diamond^m\varphi$ consistent or $\varphi \wedge \neg \diamond^m\psi$ consistent) and $(\psi \rightarrow \diamond^n\varphi$ and $\varphi \rightarrow \diamond^n\psi)$. The first condition corresponds to $d_H(X, Y) \geq n$ and the second one to $d_H(X, Y) \leq n$.

Let us consider an example of possible use of these representations for spatial reasoning. If we are looking at an object represented by ψ in an area which is at a distance in $[n_1, n_2]$ of a region represented by φ , this corresponds to a minimum distance greater than n_1 and to a Hausdorff distance less than n_2 .

Then we have to check the following relation: $\psi \rightarrow \neg\Diamond^{n_1}\varphi \wedge \Diamond^{n_2}\varphi$. This expresses in a symbolic way an imprecise knowledge about distances represented as an interval. If we consider a fuzzy interval, this extends directly using fuzzy dilation.

These expressions show how we can convert distance information, which is usually defined in an analytical way, into algebraic expressions through mathematical morphology, and then into logical ones through morphological expressions of modal operators.

Directional Relative Position

Here we rely again on the approach where the reference object is dilated with a particular structuring element defined according to the direction of interest. Let us denote by D^d the dilation corresponding to a directional information in the direction d , and by \Diamond^d the associated modal operator. Expressing that an object represented by ψ has to be in direction d with respect to a region represented by φ amounts to check the following relation: $\psi \rightarrow \Diamond^d\varphi$. In the fuzzy case, this relation can hold to some degree.

28.8 Examples in Recognition of Image Structures

Let us briefly illustrate how the concepts presented here can be used for recognizing objects in images. We consider the case of brain images, acquired with magnetic resonance imaging (MRI). In this domain, we have generic knowledge, often expressed in linguistic terms, such as: the caudate nucleus is ‘to the right’ and ‘close’ to the lateral ventricles. Fuzzy representations provide an efficient way to represent these relations, as well as interindividual variability, hence to solve the semantic gap problem by providing representations of concepts and symbols in an adequate way to reason about image features.

In previous work [95, 139, 140], two methods have been proposed for recognizing brain structures, a global one and a sequential one.

Sequential Approach

In a sequential approach [95, 140], the structures are recognized successively. To detect a structure, its spatial relations with the previously recognized structures are used to reduce the search space to image areas that satisfy these relations.

For instance, the search space in the image domain for the right caudate nucleus corresponds to the area to the right and close to the right lateral ventricle, derived from the conjunctive fusion of the results of the two morphological operations, still performed in the spatial domain (Figure 28.5).

The next step consists in segmenting the caudate nucleus. The fuzzy region of interest derived from the previous steps is used to constrain the search space and to drive the evolution of a deformable model. An

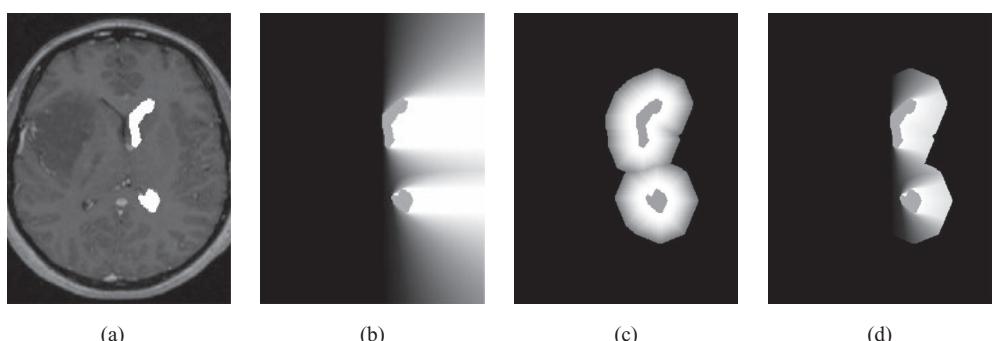


Figure 28.5 (a) The right ventricle is superimposed on one slide of the original image (MRI here). The search space of the object ‘caudate nucleus’ corresponds to the conjunctive fusion of the spatial relations ‘to the right of the right ventricle’ (b) and ‘close to the right ventricle’ (c). The fusion result is shown in (d)

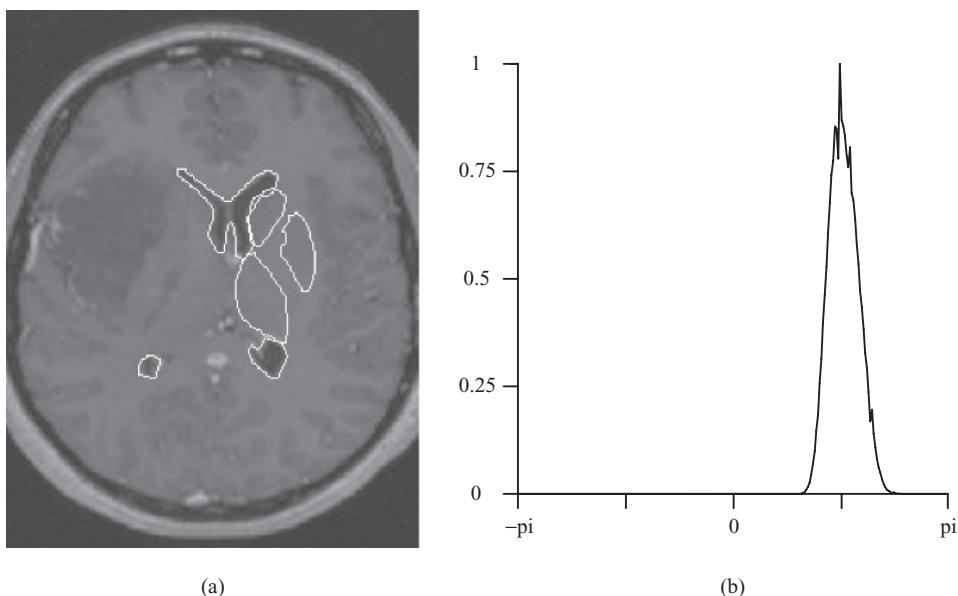


Figure 28.6 (a) Segmentation of some structures. (b) Histogram of angles between the two rightmost structures. The comparison between this histogram and the semantics of the relation ‘below’ makes it possible to compute to which degree this relation between the two structures is satisfied (0.9 here). Hence the structures should be below each other in the ontology. Similar computations of other relations lead to the recognition of the segmented structures: caudate nucleus, putamen and thalamus

initial surface is deformed toward the solution under a set of forces, including forces derived from spatial relations [140, 141]. Fusion aspects, also adequately performed with fuzzy operators, are involved when several types of knowledge are expressed for the same object. Spatial representations of each knowledge type have to be combined in order to define a search space which satisfy the fusion of the constraints (in the previous example, a constraint on distance and a constraint on direction). Fusion between different forces is also involved in the evolution process of the deformable model.

More generally, this type of approach and the use of spatial representations of spatial relations are appropriate for problems of scene navigation where the knowledge about the scene is incrementally refined when more and more objects are recognized: starting with simple objects, the scene structure is learned progressively and exploited in order to detect and recognize objects that would have been difficult to recognize directly.

Global Approach

While in the sequential approach, segmentation and recognition are performed simultaneously, in a global approach [139], several objects are first extracted from the image using a segmentation method, and then recognized. The recognition can be achieved by assessing if the spatial relations between two objects x and y are those existing in the knowledge base or in the domain ontology. This leads to a labeling of each individual region (thalamus, putamen, and caudate nucleus in Figure 28.6).

28.9 Conclusion

Spatial reasoning has to deal with different levels of granularity, concerning the spatial entities, the spatial relations, and the type of result. Fuzzy sets provide an adequate framework dealing with all these aspects.

Mathematical morphology provides a unified and consistent framework to express different types of spatial relations and to answer different questions about them, with good properties. Due to the strong algebraic structure of this framework, it applies to objects represented as sets, as fuzzy sets, and as logical formulas as well. This establishes links between theories that were so far disconnected.

Applications of this work concern model-based pattern recognition, spatial knowledge representation issues, and spatial reasoning. For instance, the spatial arrangement of objects in images provides important information for recognition and interpretation tasks, in particular when the objects are embedded in a complex environment like in medical images, as shown in the last example. Fuzzy representations of spatial relations can be incorporated efficiently in recognition methods, while bridging the gap between generic knowledge expressed in a symbolic form and image features.

References

- [1] I. Bloch. Fuzzy spatial relationships from mathematical morphology for model-based pattern recognition and spatial reasoning. In: *Discrete Geometry for Computer Imagery DGCI 2003*, Vol. LNCS 2886, Naples, Italy, November 2003, pp. 16–33.
- [2] H. Reichenbach. *The Philosophy of Space and Time*. Dover, New York, 1958.
- [3] L.A. Zadeh. Fuzzy sets and information granularity. In: M. Gupta, R. Ragade, and R. Yager (eds), *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, 1979, pp. 3–18.
- [4] W. Pedrycz. From granular computing to computational intelligence and human-centric systems. *IEEE Computational Intelligence Society*. IEEE, May 2005.
- [5] I. Bloch. On fuzzy spatial distances. In: P. Hawkes (ed.), *Advances in Imaging and Electron Physics*, Vol. 128. Elsevier, Amsterdam, 2003, pp. 51–122.
- [6] M. Aurnague, L. Vieu, and A. Borillo. Représentation formelle des concepts spatiaux dans la langue. In: M. Denis (ed.), *Language et Cognition Spatiale*, Masson, Paris, 1997, pp. 69–102.
- [7] E. Clementini, P. Di Felice, and D. Hernandez. Qualitative representation of positional information. *Artif. Intell.* 95 (1997) 317–356.
- [8] A. Herskovits. *Language and Spatial Cognition. A Interdisciplinary Study of the Prepositions in English*. Cambridge University Press, Cambridge, MA, 1986.
- [9] C. Vandeloise. *L'espace en français: sémantique des prépositions spatiales*. Seuil, travaux en linguistique, Paris, 1986.
- [10] L. Talmy. How language structures space. In: H.L. Pick and L.P. Acredolo (eds), *Spatial Orientation: Theory, Research and Application*. Plenum Press, New York, 1983.
- [11] M. Denis, F. Pazzaglia, C. Cornoldi, and L. Bertolo. Spatial discourse and navigation: An analysis of route directions in the city of Venice. *Appl. Cognit. Psychol.* 13 (1999) 145–174.
- [12] L. Talmy. *Toward a Cognitive Semantics*. MIT Press, Cambridge, MA, 2000.
- [13] J.-L. Dossal. *Aux origines du langage*. Hermès, Paris, 2000.
- [14] M. Gahegan. Proximity operators for qualitative spatial reasoning. In: A.U. Frank and W. Kuhn (eds), *Spatial Information Theory: A Theoretical Basis for GIS*, Vol. 988 of *LNCS*. Springer, Heidelberg, 1995.
- [15] H.W. Guesgen and J. Albrecht. Imprecise reasoning in geographic information systems. *Fuzzy Sets Syst.* 113 (2000) 121–131.
- [16] D.R. Montello. Scale and multiple psychologies of space. In: *International Conference COSIT'93*, Vol. 716 of *LNCS*, Elba Island, Italy, 1993, pp. 312–321.
- [17] I.E. Gordon. *Theories of Visual Perception*. John Wiley, New York, 1998.
- [18] M. Wertheimer. Gestalt theorie. *Soc. Res.* 11 (1944) 78–99.
- [19] R. Briggs. Urban cognitive distance. In: R.M. Downs and D. Stea (eds), *Image and Environment: Cognitive Mapping and Spatial Behavior*. Aldine, Chicago, 1973, pp. 361–388.
- [20] R.A. Hart and G.T. Moore. The development of spatial cognition: A review. In: R.M. Downs and D. Stea (eds), *Image and Environment: Cognitive Mapping and Spatial Behavior*. Aldine, Chicago, 1973.
- [21] M. Blades. The development of the abilities required to understand spatial representations. In: D.M. Mark and A.U. Frank (eds), *Cognitive and Linguistic Aspects of Geographic Space*, NATO ASI. Kluwer Academic Publishers, Dordrecht, 1991, pp. 81–116.
- [22] G. Edwards. Geocognostics: A new framework for spatial information theory. In: *Spatial Information Theory: A Theoretical Basis for GIS*, Vol. 1329 of *LNCS*. Springer, 1997, pp. 455–471.
- [23] J. Piaget and B. Inhelder. *The Child's Conception of Space*. Norton, New York, 1967.
- [24] A.W. Siegel and S.H. White. The development of spatial representations of large-scale environments. In: H.W. Reese (ed.), *Advances in Child Development and Behavior*, Vol. 10. Academic Press, New York, 1975.

- [25] E. Mellet, S. Bricogne, N. Tzourio-Mazoyer, O. Ghaëm, L. Petit, L. Zago, O. Etard, A. Berthoz, B. Mazoyer, and M. Denis. Neural correlates of topographic mental exploration: The impact of route versus survey perspective learning. *NeuroImage* 12(5) (2000) 588–600.
- [26] S. Vieilledent, S.M. Kosslyn, A. Berthoz, and M.D. Giraud. Does mental simulation of following a path improve navigation performance without vision? *Cognit. Brain Res.* 16 (2) (2003) 238–249.
- [27] S. Lambrey, I. Viaud-Delmon, and A. Berthoz. Influence of a sensorimotor conflict on the memorization of a path traveled in virtual reality. *Cognit. Brain Res.*, 14 (2002) 177–186.
- [28] L. Nadel. The psychobiology of spatial behavior: the hippocampal formation and spatial mapping. In: E. Alleva, H.-P. Lipp, L. Nadel, A. Fasolo, and L. Ricceri (eds), *Behavioral Brain Research in Naturalistic and Semi-Naturalistic Settings: Possibilities and Perspectives*. Kluwer Press, Dordrecht, 1995.
- [29] A. Berthoz. Stratégies cognitives et mémoire spatiale. In: *Colloque Cognitique*, Paris, France, 2002.
- [30] L. Nadel. Multiple perspectives in spatial cognition. In: *Colloque Cognitique*, Paris, France, 2002.
- [31] B. Kuipers. The spatial semantic hierarchy. *Artif. Intell.* 119 (2000) 191–233.
- [32] D.M. Mark and M.J. Egenhofer. Modeling spatial relations between lines and regions: Combining formal mathematical models and human subjects testing. *Cartography Geogr. Inf. Syst.* 21(4) (1994) 195–212.
- [33] D.J. Peuquet. Representations of geographical space: Toward a conceptual synthesis. *Ann. Assoc. Am. Geographers* 78(3) (1988) 375–394.
- [34] Z.-Q. Liu and R. Satur. Contextual fuzzy cognitive map for decision support in geographic information systems. *IEEE Trans. Fuzzy Syst.*, 7(5) (1999) 495–505.
- [35] R. Satur and Z.-Q. Liu. A contextual fuzzy cognitive map framework for geographic information systems. *IEEE Trans. Fuzzy Syst.* 7(5) (1999) 481–494.
- [36] B. Kuipers. Modeling spatial knowledge. *Cognit. Sci.* 2 (1978) 129–153.
- [37] B.J. Kuipers and T.S. Levitt. Navigation and mapping in large-scale Space. *AI Mag.* 9(2) (1988) 25–43.
- [38] P. Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. MIT Press, Cambridge, MA, 2000.
- [39] J. Aisbett and G. Gibbon. A general formulation of conceptual spaces as a meso level representation. *Artif. Intell.* 133 (2001) 189–232.
- [40] P. Gärdenfors and M.-A. Williams. Reasoning about Categories in Conceptual Spaces. In: *IJCAI'01*, Seattle, USA, 2001, pp. 385–392.
- [41] L. Vieu. Spatial representation and reasoning in artificial intelligence. In: O. Stock (ed.), *Spatial and Temporal Reasoning*. Kluwer, Dordrecht, 1997, pp. 5–41.
- [42] C. Vandelouise. *La dimension en français, de l'espace à la matière*. Hermès, Lavoisier, Paris, 2004.
- [43] A.C. Bovik and J.D. Gibson. *Handbook of Image and Video Processing*. Academic Press, Inc. Orlando, FL, 2000.
- [44] P. Preparata and M. I. Shamos. *Computational Geometry*. Springer-Verlag, New York, 1985.
- [45] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic, Boston, 1994.
- [46] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.* 22(12) (2000) 1349–1380.
- [47] S. Harnad. The symbol grounding problem. *Physica* 42 (1990) 335–346.
- [48] S. Coradeschi and A. Saffiotti. Anchoring symbols to vision data by fuzzy logic. In: A. Hunter and S. Parsons (eds), *ECSQARU'99*, Vol. 1638 of *LNCS*, London, July 1999. Springer, pp. 104–115.
- [49] I. Bloch, H. Maître, and M. Anvari. Fuzzy adjacency between image objects. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 5(6) (1997) 615–653.
- [50] I. Bloch. On fuzzy distances and their use in image processing under imprecision. *Pattern Recognit.* 32(11) (1999) 1873–1895.
- [51] I. Bloch. Fuzzy relative position between objects in image processing: A morphological approach. *IEEE Trans. Pattern Anal. Mach. Intell.* 21(7) (1999) 657–664.
- [52] I. Bloch, O. Colliot, and R. Cesar. On the ternary spatial relation between. *IEEE Trans. Syst. Man Cybern. SMC-B* 36(2) (2006) 312–327.
- [53] J. Freeman. The modelling of spatial relations. *Comput. Graph. Image Process.* 4 (2) (1975) 156–171.
- [54] A.G. Cohn. Qualitative spatial representations. In: *IJCAI99 Workshop on Adaptive Spatial Representations of Dynamic Environments*, Stockholm, Sweden, July 31–August 6, 1999, pp. 33–52.
- [55] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Inf. Sci.* 8 (1975) 199–249.
- [56] S. Dutta. Approximate spatial reasoning: Integrating qualitative and quantitative constraints. *Int. J. Approx. Reason.* 5 (1991) 307–331.
- [57] M. Aiello. *Spatial Reasoning, Theory and Practice*. PhD Thesis. University of Amsterdam, February 2002.
- [58] M. Aiello, I. Pratt-Hartmann, and J. van Benthem (eds). *Handbook of Spatial Logic*. Springer, Dordrecht, The Netherland, 2006.

- [59] M. Aiello and J. van Benthem. A modal walk through space. *J. Appl. Non Class. Log.* 12(3–4) (2002) 319–364.
- [60] P. Balbiani. The modal multilogic of geometry. *J. Appl. Non-Class. Log.* 8 (1998) 259–281.
- [61] P. Balbiani. *Représentation Logique et Traitement Algorithmique de L'espace*. Habilitation à diriger des recherches. Laboratoire d'Informatique de Paris Nord, Université Paris 13, 1999.
- [62] I. Bloch. Modal logics based on mathematical morphology for spatial reasoning. *J. Appl. Non-Classical Log.* 12(3–4) (2002) 399–424.
- [63] A.G. Cohn. Qualitative spatial representation and reasoning techniques. In: G. Brewka, C. Habel, and B. Nebel (eds), *KI-97, LNAI*. Springer Verlag, London, 1997, pp. 1–30.
- [64] G. Ligozat and J. Renz. What is qualitative calculus? A general framework. In: *PRICAI'04, LNCS 3157*, Auckland, New Zealand, 2004, pp. 53–64.
- [65] B. Landau and R. Jackendoff. ‘What’ and ‘Where’ in spatial language and spatial cognition. *Behav. Brain Sci.* 16 (1993) 217–265.
- [66] N. Asher and L. Vieu. Toward a geometry of common sense: A semantics and a complete axiomatization of mereotopology. In: *IJCAI'95*, San Mateo, CA, 1995, pp. 846–852.
- [67] A. Varzi. Parts, wholes, and part-whole relations: The prospects of mereotopology. *Data Knowl. Eng.* 20(3) (1996) 259–286.
- [68] D. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connection. In: B. Nebel, C. Rich, and W. Swartout (eds), *Principles of Knowledge Representation and Reasoning KR'92*, San Mateo, CA, 1992, Kaufmann, pp. 165–176.
- [69] C. Freksa, M. Knauff, B. Krieg-Bruckner, B. Nebel, and T. Barkowsky. *Spatial Cognition IV – Reasoning, Action, Interaction*. Springer Series in Perception Engineering. Springer Verlag, Heidelberg, 2004.
- [70] G. Ligozat. Reasoning about cardinal directions. *J. Vis. Lang. Comput.* 9 (1998) 23–44.
- [71] J. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM* 26(11) (1983) 832–843.
- [72] S.K. Chang, Q.Y. Shi, and C.W. Yan. Iconic indexing by 2D strings. *IEEE Trans. Pattern Anal. Mach. Intell.* 9(3) (1987) 413–428.
- [73] C. Freksa and K. Zimmermann. On the utilization of spatial structures for cognitively plausible and efficient reasoning. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Chicago, October 1992, pp. 261–266.
- [74] A. Cohn, B. Bennett, J. Gooday, and N.M. Gotts. Representing and reasoning with qualitative spatial relations about Regions. In: O. Stock (ed.), *Spatial and Temporal Reasoning*. Kluwer, Dordrecht, 1997, pp. 97–134.
- [75] J. Bateman and S. Farrar. Towards a generic foundation for spatial ontology. In: *International Conference on Formal Ontology in Information Systems (FOIS-2004)*, Trento, Italy, 2004, pp. 237–248.
- [76] R. Casati, B. Smith, and A.C. Varzi. Ontological tools for geographic representation. In: N. Guarino (ed.), *Formal Ontology in Information Systems*. IOS Press, Amsterdam, 1998, pp. 77–85.
- [77] E. Klien and M. Lutz. The role of spatial relations in automating the semantic annotation of geodata. In: A.G. Cohn and D.M. Marks (eds), *Conference on Spatial Information Theory (COSIT 2005)*, Vol. LNCS 3693, New York, September 14–18, 2005.
- [78] S. Dasiospoulou, V. Mezaris, I. Kompatsiaris, V.K. Papastathis, and MG Strintzis. Knowledge-assisted semantic video object detection. *IEEE Trans. Circuits Syst. Video Technol.* 15(10) (2005) 1210–1224.
- [79] D. Han, B.J. You, Y.S.E. Kim, and I.L.H. Suh. A generic shape matching with anchoring of knowledge primitives of object ontology. In: M. Kamel and A. Campilho (eds), *ICIPAR 2005*, Vol. LNCS 3646, Toronto, Canada, September 28–30, 2005, pp. 473–480.
- [80] P.F. Dominey, J.D. Boucher, and T. Inui. Building an adaptive spoken language interface for perceptually grounded human-robot interaction. In: *4th IEEE/RAS International Conference on Humanoid Robots*, Vol. 1, Los Angeles, CA, 2004, pp. 168–183.
- [81] O. Dameron. Symbolic model of spatial relations in the human brain. In: *Mapping the Human Body: Spatial Reasoning at the Interface between Human Anatomy and Geographic Information Science*. University of Buffalo, USA, April 2005.
- [82] O. Dameron, B. Gibaud, and X. Morandi. Numeric and symbolic knowledge representation of cerebral cortex anatomy: Methods and preliminary results. *Surg. Radiol. Anat.* 26(3) (2004) 191–197.
- [83] M. Donnelly, T. Bittner, and C. Rosse. A formal theory for spatial representation and reasoning in biomedical ontologies. *Artif. Intell. Med.* 36(1) (2006) 1–27.
- [84] S. Schulz, U. Hahn, and M. Romacker. Modeling anatomical spatial relations with description logics. In: *Annual Symposium of the American Medical Informatics Association. Converging Information, Technology, and Health Care (AMIA 2000)*, Los Angeles, CA, 2000, pp. 779–783.
- [85] J.G. Stell. Part and complement: Fundamental concepts in spatial relations. *Ann. Math. Artif. Intell.* 41(1) (2004) 1–17.

- [86] F. Le Ber and L. Mangelink. A formal representation of landscape spatial patterns to analyze satellite images. *AI Appl.* 12(1–3) (1998) 51–59.
- [87] J.G. Stell. The representation of discrete multi-resolution spatial knowledge. In: A.G. Cohn, F. Giunchiglia, and B. Selman (eds), *7th International Conference on Principles of Knowledge Representation and Reasoning KR 2000*, Breckenridge, CO, 2000. Morgan Kaufmann, San Francisco, CA, pp. 38–49.
- [88] J.G. Stell. Spatio-temporal granularity. In: *ECAI 2000, Workshop on Spatio-Temporal Reasoning*, Berlin, Germany, 2000, pp. 55–61.
- [89] S. Hazarika and A.G. Cohn. A taxonomy for spatial vagueness: An alternative egg-yolk interpretation. In: *Spatial Vagueness, Uncertainty and Granularity Symposium*, Ogunquit, Maine, 2001.
- [90] N. Sladoje and J. Lindblad. Representation and reconstruction of fuzzy disks by moments. *Fuzzy Sets Syst.* 158(5) (2007) 517–534.
- [91] N. Sladoje, I. Nyström, and P.K. Saha. Perimeter and area estimations of digitized objects with fuzzy borders. In: *DGCI 2003 LNCS 2886*, Napoli, Italy, 2003, pp. 368–377.
- [92] D. Dubois and H. Prade. On distance between fuzzy points and their use for plausible reasoning. In: *International Conference Systems, Man, and Cybernetics*, 1983, pp. 300–303.
- [93] A. Rosenfeld. Distances between Fuzzy Sets. *Pattern Recognit. Lett.* 3 (1985) 229–233.
- [94] R. Krishnapuram, J.M. Keller, and Y. Ma. Quantitative analysis of properties and spatial relations of fuzzy image regions. *IEEE Trans. Fuzzy Syst.* 1(3) (1993) 222–233.
- [95] I. Bloch, T. Géraud, and H. Maître. Representation and fusion of heterogeneous fuzzy information in the 3D space for model-based structural recognition – Application to 3D brain imaging. *Artif. Intell.* 148 (2003) 141–175.
- [96] K. Miyajima and A. Ralescu. Spatial organization in 2D images. In: *Third IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'94*, Orlando, FL, June 1994, pp. 100–105.
- [97] I. Bloch. Information combination operators for data fusion: A comparative review with classification. *IEEE Trans. Syst. Man Cybern.* 26(1) (1996) 52–67.
- [98] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, 1980.
- [99] D. Dubois, H. Prade, and R. Yager. Merging fuzzy information. In: J.C. Bezdek, D. Dubois, and H. Prade, (eds), *Handbook of Fuzzy Sets Series, Approximate Reasoning and Information Systems*. Kluwer, Dordrecht 1999, Chapter 6.
- [100] I. Bloch. Fuzzy spatial relationships for image processing and interpretation: A review. *Image Vis. Comput.* 23(2) (2005) 89–110.
- [101] I. Bloch and H. Maître. Fuzzy mathematical morphologies: A comparative study. *Pattern Recognit.* 28(9) (1995) 1341–1387.
- [102] D. Sinha and E.R. Dougherty. Fuzzification of set inclusion: Theory and applications. *Fuzzy Sets Syst.* 55 (1993) 15–42.
- [103] V.R. Young. Fuzzy subsethood. *Fuzzy Sets Syst.* 77 (1996) 371–384.
- [104] B. Kosko. Fuzziness vs. probability. *Int. J. Gen. Syst.* 17 (1990) 211–240.
- [105] W. Bandler and L. Kohout. Fuzzy power sets and fuzzy implication operators. *Fuzzy Sets Syst.* 4 (1980) 13–30.
- [106] R. Willmott. Two fuzzier implication operators in the theory of fuzzy power sets. *Fuzzy Sets Syst.* 4 (1980) 31–36.
- [107] C. Demko and E.H. Zahzah. Image understanding using fuzzy isomorphism of fuzzy structures. In: *IEEE International Conference on Fuzzy Systems*, Yokohama, Japan, March 1995, pp. 1665–1672.
- [108] A. Rosenfeld. The Fuzzy Geometry of Image Subsets. *Pattern Recognit. Lett.* 2 (1984) 311–317.
- [109] J.K. Udupa and S. Samarasekera. Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. *Graph. Models Image Process.* 58(3) (1996) 246–261.
- [110] A. Rosenfeld and R. Klette. Degree of adjacency or surroundness. *Pattern Recognit.* 18(2) (1985) 169–177.
- [111] R. Zwick, E. Carlstein, and D.V. Budescu. Measures of similarity among fuzzy concepts: A comparative analysis. *Int. J. Approx. Reason.* 1 (1987) 221–242.
- [112] M. Masson and T. Denœux. Multidimensional scaling of fuzzy dissimilarity data. *Fuzzy Sets Syst.* 128 (2002) 339–352.
- [113] L. Boxer. On Hausdorff-like metrics for fuzzy sets. *Pattern Recognit. Lett.* 18 (1997) 115–118.
- [114] B.B. Chauduri and A. Rosenfeld. On a metric distance between fuzzy sets. *Pattern Recognit. Lett.* 17 (1996) 1157–1160.
- [115] L.C. de Barros, R.C. Bassanezi, and P.A. Tonelli. On the continuity of the Zadeh's extension. In: *Seventh IFSA World Congress*, Vol. II, Prague, June 1997, pp. 3–8.
- [116] M.L. Puri and D.A. Ralescu. Differentials of fuzzy functions. *J. Math. Anal. Appl.* 91 (1983) 552–558.
- [117] I. Bloch. Distances in fuzzy sets for image processing derived from fuzzy mathematical morphology

- (invited conference). In: *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Granada, Spain, July 1996, pp. 1307–1312.
- [118] J. M. Keller and X. Wang. Comparison of spatial relation definitions in computer vision. In: *ISUMA-NAFIPS'95*, College Park, MD, September 1995, pp. 679–684.
- [119] K. Miyajima and A. Ralescu. Spatial organization in 2D segmented images: Representation and recognition of primitive spatial relations. *Fuzzy Sets Syst.* 65 (1994) 225–236.
- [120] P. Matsakis and L. Wendling. A new way to represent the relative position between areal objects. *IEEE Trans. Pattern Anal. Mach. Intell.* 21(7) (1999) 634–642.
- [121] L.T. Koczy. On the description of relative position of fuzzy patterns. *Pattern Recognit. Lett.* 8 (1988) 21–28.
- [122] I. Bloch. Fuzzy relative position between objects in images: A morphological approach. In: *IEEE International Conference on Image Processing ICIP'96*, Vol. II, Lausanne, September 1996, pp. 987–990.
- [123] I. Bloch. Fuzzy relative position between objects in image processing: New definition and properties based on a morphological approach. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 7(2) (1999) 99–133.
- [124] I. Bloch and A. Ralescu. Directional relative position between objects in image processing: A comparison between fuzzy approaches. *Pattern Recognit.* 36 (2003) 1563–1582.
- [125] C.M. Takemura, R.M. Cesar, Jr. and I. Bloch. Fuzzy modeling and evaluation of the spatial relation ‘Along.’ In: *10th Iberoamerican Congress on Pattern Recognition, CIARP*, Vol. LNCS 3773, La Havana, Cuba, November 2005, pp. 837–848.
- [126] C. Hudelot, J. Atif, and I. Bloch. An ontology of spatial relations using fuzzy concrete domains. In: *AISB symposium on Spatial Reasoning and Communication*, Newcastle, UK, April 2007.
- [127] I. Bloch. Mathematical morphology and spatial relationships: Quantitative, semi-quantitative and symbolic settings. In: L. Szandera and P. Matsakis (eds), *Applying Soft Computing in Defining Spatial Relationships*. Physica Verlag, Springer, 2002, pp. 63–98.
- [128] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.
- [129] B. de Baets. Fuzzy morphology: A logical approach. In: B. Ayyub and M. Gupta (eds), *Uncertainty in Engineering and Sciences: Fuzzy Logic, Statistics and Neural Network Approach*. Kluwer Academic, Dordrecht, 1997, pp. 53–67.
- [130] D. Sinha and E. Dougherty. Fuzzy mathematical morphology. *J. Vis. Commun. Image Represent.* 3(3) (1992) 286–302.
- [131] M. Nachtegael and E.E. Kerre. Classical and fuzzy approaches towards mathematical morphology. In E.E. Kerre and M. Nachtegael (eds), *Fuzzy Techniques in Image Processing*, Studies in Fuzziness and Soft Computing. Physica-Verlag, Springer, Heidelberg, 2000, Chapter 1, pp. 3–57.
- [132] I. Bloch and J. Lang. Towards mathematical morpho-logics. In: *8th International Conference on Information Processing and Management of Uncertainty in Knowledge based Systems IPMU 2000*, Vol. III, Madrid, Spain, 2000, pp. 1405–1412.
- [133] C. Lafage and J. Lang. Logical representation of preferences for group decision making. In: A.G. Cohn, F. Giunchiglia, and B. Selman (eds), *7th International Conference on Principles of Knowledge Representation and Reasoning KR 2000*, Breckenridge, CO, 2000. Morgan Kaufmann, San Francisco, CA, pp. 457–468.
- [134] B. Chellas. *Modal Logic, An Introduction*. Cambridge University Press, Cambridge, 1980.
- [135] I. Bloch. Using mathematical morphology operators as modal operators for spatial reasoning. In: *ECAI 2000, Workshop on Spatio-Temporal Reasoning*, Berlin, Germany, 2000, pp. 73–79.
- [136] D. Dubois, H. Prade, and C. Testemale. Weighted fuzzy pattern matching. *Fuzzy Sets Syst.* 28 (1988) 313–331.
- [137] I. Bloch. Spatial representation of spatial relationships knowledge. In: A.G. Cohn, F. Giunchiglia, and B. Selman (eds), *7th International Conference on Principles of Knowledge Representation and Reasoning KR 2000*, Breckenridge, CO, 2000. Morgan Kaufmann, San Francisco, CA, pp. 247–258.
- [138] T. Géraud, I. Bloch, and H. Maître. Atlas-guided recognition of cerebral structures in MRI using fusion of fuzzy structural information. In: *CIMAF'99 Symposium on Artificial Intelligence*, La Havana, Cuba, 1999, pp. 99–106.
- [139] E. Bengoetxea, P. Larranaga, I. Bloch, A. Perchant, and C. Boeres. Inexact graph matching by means of estimation of distribution algorithms. *Pattern Recognit.* 35 (2002) 2867–2880.
- [140] O. Colliot, O. Camara, and I. Bloch. Integration of fuzzy spatial relations in deformable models – Application to brain MRI segmentation. *Pattern Recognit.* 39 (2006) 1401–1414.
- [141] J. Atif, O. Nempong, O. Colliot, E. Angelini, and I. Bloch. Level set deformable models constrained by fuzzy spatial relations. In: *Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU*, Paris, France, 2006, pp. 1534–1541.

29

Rough–Neural Methodologies in Granular Computing

Sushmita Mitra and Mohua Banerjee

29.1 Introduction

The theory of *rough sets* [1] has turned out to be very useful in managing uncertainty that arises from *granularity* in the domain of discourse. Effectiveness of the theory has been investigated in the areas of artificial intelligence and cognitive sciences, especially for representation of and reasoning with vague and/or imprecise knowledge, data classification and analysis, machine learning, and knowledge discovery [2]. The main use of rough sets in pattern recognition has been toward dimensionality reduction, classification, and clustering.

Hybridization, exploiting the characteristics of rough sets, includes the *rough-fuzzy* [3, 4], *rough-neuro* [5, 6], and *rough-neuro-fuzzy* [7–10] approaches. The primary role of rough sets here is in managing uncertainty and extracting domain knowledge. Other recent investigations concern the modular evolutionary *rough-neuro-fuzzy* integration [10, 11] for classification and rule mining. Here the use of evolutionary computing helps in generating an optimal neuro-fuzzy architecture, which is initially encoded using rough sets for extracting crude domain knowledge from data.

Granular computing [12] is useful in finding meaningful patterns in data by expressing and processing chunks of information (granules). These are regarded as essential entities in all cognitive pursuits geared toward establishing meaningful patterns in data. Soft granules can be defined in terms of membership functions. Increased granularity reduces attribute distinctiveness, resulting in loss of useful information, while finer grains lead to partitioning difficulty. The concept of granular computing allows one to concentrate all computational effort on some specific and problem-oriented subsets of a complete database. It also helps split an overall computing effort into several subtasks, leading to a *modularization* effect. This enables efficient mining of large data sets.

The granularity in rough set theory is due to an *indiscernibility* relation between objects in the domain, which may be induced by a given set of *attributes* ascribed to the objects. In general, in the presence of this granularity, it is not possible to describe/recognize a concept. The basic idea of the theory is to approximate a *rough* (imprecise) concept in the domain by the *exact* concepts of *lower* and *upper approximations*, determined by the indiscernibility relation. The indiscernibility relation and these approximations are used to define notions of *discernibility matrices*, *discernibility functions*, *reducts*, and *dependency factors*, all of which play a fundamental role in the reduction of knowledge. Rough set notions are thus formulated

assuming the presence of an intrinsic granularity in the domain, and provide important methodologies in granular computing.

In the next section, we make a survey of work done using these (and hybrid) methodologies to design neural networks. Section 29.3 presents some of our own work in the area.

29.2 Rough Sets and Neural Structures

Many have looked into the implementation of decision rules extracted from operation data using rough set formalism, especially in problems of machine learning from examples, and control theory [2]. In the context of neural networks, one of the first attempts of such implementation was made by Yasdi [6]. The intention was to use rough sets as a tool for structuring the neural networks. The methodology consists of generating rules from training examples by rough set learning and mapping them into a single layer of connection weights of a four-layered neural network. Attributes appearing as rule antecedents (consequents) become the input (output) nodes, while the dependency factors become the weight of the adjoining links in the hidden layer. The input and output layers involve non-adjustable binary weights. *Max*, *min*, and OR operators are modeled at the hidden nodes, based on the syntax of the rules. The backpropagation algorithm is slightly modified. However, the network was not tested on any real-life problem and no comparative study was provided to bring out the effectiveness of this hybrid approach. Some other attempts on application of rough sets in neurocomputing use rough sets for knowledge discovery at the level of data acquisition, (viz., in preprocessing of the feature vectors), and not for structuring the network, e.g., [5, 13, 14].

The concept of a *rough neuron* was introduced by Lingras [15] – it is taken to consist of two parts, the *lower bound* and *upper bound* neurons. A network is framed employing the backpropagation learning algorithm, but the novelty lies in the definition of connections between rough neurons. The work was applied on urban traffic data. The *approximation neuron* [16, 17] was designed based on set approximations and rough membership functions. The computation considers an input depending on equivalence classes of measurements obtained from known cases. Based on the notion of approximation neurons, a *decider neuron* was also proposed in [16, 17]. The input consists, in one part, of a set of measurements by an approximation neuron network obtained corresponding to an object that is to be classified. The other part of the input is an information granule represented by a set of decision rules.

Generalizations of neural network models have been discussed in [18]. An *information granule system* is defined, and general granule construction schemes are formulated. The key feature is an interface represented by approximation spaces to enable communication between sending and receiving agents. These schemes for complex object construction are called *rough neural networks*. Using the concept of information granule systems, a feedforward neural-like network has been proposed in [19]. Another generalization is presented in [20], where a ‘calculus of approximate parts,’ viz., *rough mereology* is used to propose the notion of a *rough mereological perceptron*.

Apart from these proposals, there has been some work in designing other hybrid neural networks involving rough sets. The *rough-fuzzy multilayer perceptron* (MLP) was introduced in [7]. The intention was to use fuzzy sets in handling linguistic input information and ambiguity in output decision, while rough sets would help in extracting domain knowledge for determining the network parameters. The *evolutionary rough-fuzzy MLP* presented in the next section, in fact, uses this work. Sarkar and Yegnanarayana [9] have used a fuzzy–rough set theoretic approach to determine the importance of different subsets of incomplete information sources, which are used by several small feedforward subnetworks. The individual solutions are then combined to obtain the final classification result. In [21], a *normalizing neural network* is presented. It is a probabilistic network that is an extension of the classical Bayesian approach to classification problems. Inputs are defined on finite decision distributions. The idea is to take into account possible inconsistencies due to multiple sources of decision distributions. This is a generalization of the case when an input neuron is associated with a single decision rule (generated by reducts), and there is one output neuron for each decision value.

29.3 Evolutionary Rough–Fuzzy MLP

A recent trend in neural network design for large-scale problems is to split the original task into simpler subtasks and use a subnetwork module for each of the subtasks. It has been shown that by combining the output of several subnetworks in an ensemble, one can improve the generalization ability over that of a single large network. This type of *divide-and-conquer* strategy makes it possible to effectively mine large volumes of data while discovering information [22]. In this section we describe a modular approach in the evolutionary rough–fuzzy–neural framework [10, 11, 23], with application to classification and rule generation.

Rough set theory is used to encode the weights of the neural network with initial knowledge as well as to determine the network size. Fuzzy set theory is utilized for discretization of the feature space. Information granules in this discretized feature space are then used for extracting dependency rules in terms of fuzzy membership values. This helps in preserving all the class representative points in the dependency rules by adaptively applying a threshold that automatically takes care of the shape of membership functions. Since the computation is performed on the granules rather than on the patterns themselves, the time required is smaller. Hence the system is suitable for handling large data sets. An analogous approach to designing a rough self-organizing map (RSOM) has also been developed [24].

An l -class classification problem is split into l two-class subproblems. Crude subnetwork modules are initially encoded, for each two-class sub-problem, from the dependency rules. These subnetworks are then combined and the final network is evolved using a genetic algorithm (GA) with *restricted* mutation operator which utilizes the knowledge of the modular structure already generated for faster convergence. The GA tunes the fuzzification parameters, and network weight and structure simultaneously, by optimizing a single fitness function. This methodology helps in imposing a structure on the weights, which results in a network more suitable for rule generation. Performance of the algorithm is compared with related techniques.

29.3.1 Rough–Fuzzy MLP

Any input feature value is described in terms of some combination of overlapping membership values in the linguistic property sets *low* (L), *medium* (M), and *high* (H). An n -dimensional pattern $\mathbf{X}_i = [a_1, a_2, \dots, a_n]$ is represented as a $3n$ -dimensional vector

$$\mathbf{X}_i = [\mu_{\text{low}(a_1)}(\mathbf{X}_i), \mu_{\text{medium}(a_1)}(\mathbf{X}_i), \mu_{\text{high}(a_1)}(\mathbf{X}_i), \dots, \mu_{\text{high}(a_n)}(\mathbf{X}_i)], \quad (1)$$

where the μ values indicate the input membership functions of the corresponding linguistic functions *low*, *medium*, and *high* along each feature axis.

An l -class problem domain corresponds to l nodes in the output layer of the network. The output membership of the i th pattern in class k , lying in the range $[0, 1]$, is defined as

$$\mu_{ik}(\mathbf{X}_i) = \frac{1}{1 + \left(\frac{z_{ik}}{f_d}\right)^{f_e}}, \quad (2)$$

where z_{ik} is the weighted distance of the training pattern \mathbf{X}_i from class C_k , and the positive constants f_d and f_e are the denominational and exponential fuzzy generators controlling the amount of fuzziness in the class membership set. This constitutes the fuzzy MLP [25].

29.3.1.1 Rule Generation

Let $\mathcal{S} = < U, A >$ be a decision table, with C and $D = \{d_1, \dots, d_l\}$ its sets of condition and decision attributes, respectively. Divide the decision table $\mathcal{S} = < U, A >$ into l tables $\mathcal{S}_i = < U_i, A_i >$, $i = 1, \dots, l$, corresponding to the l decision attributes d_1, \dots, d_l , where

$$U = U_1 \cup \dots \cup U_l \quad \text{and} \quad A_i = C \cup \{d_i\}.$$

Let $\{x_{i_1}, \dots, x_{i_p}\}$ be the set of those objects of U_i that occur in \mathcal{S}_i , $i = 1, \dots, l$.

Now for each d_i -reduct $B = \{b_1, \dots, b_k\}$, say, a discernibility matrix (denoted $\mathbf{M}_{d_i}(B)$) from the d_i -discernibility matrix is defined as follows [7]:

$$c_{ij} = \{a \in B : a(x_i) \neq a(x_j)\}, \quad (3)$$

for $i, j = 1, \dots, n$.

For each object $x_j \in x_{i_1}, \dots, x_{i_p}$, the discernibility function $f_{d_i}^{x_j}$ is defined as

$$f_{d_i}^{x_j} = \bigwedge \left\{ \bigvee (c_{ij}) : 1 \leq i, j \leq n, j < i, c_{ij} \neq \emptyset \right\}, \quad (4)$$

where $\bigvee (c_{ij})$ is the disjunction of all members of c_{ij} . Then $f_{d_i}^{x_j}$ is brought to its conjunctive normal form (c.n.f.). One thus obtains a dependency rule r_i , namely, $P_i \leftarrow d_i$, where P_i is the disjunctive normal form (d.n.f.) of $f_{d_i}^{x_j}$, $j \in i_1, \dots, i_p$.

The dependency factor df_i for r_i is given by

$$df_i = \frac{\text{card}(POS_i(d_i))}{\text{card}(U_i)}, \quad (5)$$

where $POS_i(d_i) = \bigcup_{X \in I_{d_i}} l_i(X)$, and $l_i(X)$ is the lower approximation of X with respect to I_i . In this case, $df_i = 1$ [7].

29.3.1.2 Knowledge Encoding

Consider the case of feature j for class C_k in the l -class problem domain. The inputs for the i th representative sample \mathbf{X}_i are mapped to the corresponding three-dimensional feature space of $\mu_{low(a_j)}(\mathbf{X}_i)$, $\mu_{medium(a_j)}(\mathbf{X}_i)$, and $\mu_{high(a_j)}(\mathbf{X}_i)$ by equation (1). Let these be represented by L_j , M_j , and H_j , respectively. As the method considers multiple objects in a class, a separate $n_k \times 3n$ -dimensional attribute-value decision table is generated for each class C_k (where n_k indicates the number of objects in C_k).

The absolute distance between each pair of objects is computed along each attribute L_j , M_j , H_j for all j . We modify equation (3) to directly handle a real-valued attribute table consisting of fuzzy membership values. We define [7]

$$c_{ij} = \{a \in B : |a(x_i) - a(x_j)| > Th\} \quad (6)$$

for $i, j = 1, \dots, n_k$, where Th is an adaptive threshold. Note that the adaptivity of this threshold is in-built, depending on the inherent shape of the membership function.

The hidden layer nodes model the first-level (innermost) operator in the antecedent part of a rule, which can be either a conjunct or a disjunct. The output layer nodes model the outer-level operands, which can again be either a conjunct or a disjunct. For each inner-level operator, corresponding to one output class (one dependency rule), one hidden node is dedicated. Only those input attributes that appear in this conjunct or disjunct are connected to the appropriate hidden node, which in turn is connected to the corresponding output node. Each outer-level operator is modeled at the output layer by joining the corresponding hidden nodes. Note that a single attribute (involving no inner-level operators) is directly connected to the appropriate output node via a hidden node, to maintain uniformity in rule mapping.

Let the dependency factor for a particular dependency rule for class C_k be $df = \alpha = 1$ by equation (5). The weight w_{ki}^1 between a hidden node i and output node k is set at $\frac{\alpha}{fac} + \varepsilon$, where fac refers to the number of outer-level operands in the antecedent of the rule and ε is a small random number taken to destroy any symmetry among the weights. Note that $fac \geq 1$ and each hidden node is connected to only one output node. Let the initial weight so clamped at a hidden node be denoted as β . The weight $w_{ia_j}^0$ between an attribute a_j (where a corresponds to *low* (L), *medium* (M), or *high* (H)) and hidden node i is set to $\frac{\beta}{fac} + \varepsilon$, such that $facd$ is the number of attributes connected by the corresponding inner-level operator. Again, $facd \geq 1$. Thus for an l -class problem domain there are at least l hidden nodes. All other

possible connections in the resulting network are set as small random numbers. It is to be mentioned that the number of hidden nodes is automatically determined from the number of dependency rules, while their connectivity follows from the syntax of these rules.

29.3.2 Modular Approach

Embedding modularity (i.e., to perform local and encapsulated computation) into neural networks leads to many advantages as compared to the use of a single network. For instance, constraining the network connectivity increases its learning capacity and permits its application to large-scale problems with relevance to data mining [22]. It is easier to encode *a priori* knowledge in modular neural networks. In addition, the number of network parameters can be reduced by using modularity. This feature speeds computation and can improve the generalization capability of the system.

It involves two phases. First an l -class classification problem is split into l two-class problems. Rough set theoretic concepts are used to encode domain knowledge into each of the l subnetworks, using equations (4)–(6). The number of hidden nodes and connectivity of the knowledge-based subnetworks is automatically determined. A two-class problem leads to the generation of one or more crude subnetworks, each encoding a particular decision rule. Let each of these constitute a pool. So we obtain $m \geq l$ pools of knowledge-based modules. Each pool k is perturbed to generate a total of n_k subnetworks, such that $n_1 = \dots = n_k = \dots = n_m$. These pools constitute the initial population of subnetworks, which are then evolved independently using genetic algorithms.

At the end of training, the modules (or subnetworks) corresponding to each two-class problem are concatenated to form an initial network for the second phase. The intermodule links are initialized to small random values as depicted in Figure 29.1. A set of such concatenated networks forms the initial population of the GA. Note that the individual modules cooperate, rather than compete, with each other while evolving toward the final solution. The mutation probability for the intermodule links is now set to a high value, while that of intramodule links is set to a relatively lower value. This sort of *restricted* mutation helps preserve some of the localized rule structures, already extracted and evolved, as potential solutions. The initial population for the GA of the entire network is formed from all possible combinations of these individual network modules and random perturbations about them. This ensures that for complex multimodal pattern distributions all the different representative points remain in the population. The algorithm [10, 11] then searches through the reduced space of possible network topologies. The steps are summarized below, followed by an example:

1. **for** each class, generate rough set dependency rules.
2. Map each of the dependency rules to a separate subnetwork module (fuzzy MLP).
3. Partially evolve each of the subnetworks using conventional GA.
4. Concatenate the subnetwork modules to obtain the complete network. For concatenation the intramodule links are left unchanged, while the intermodule links are initialized to low random values. Note that each of the subnetworks solves a two-class classification problem, while the concatenated network solves the actual l -class problem. Every possible combination of subnetwork modules is generated to form a pool of networks.
5. The pool of networks is evolved using a *modified* GA with an adaptive or variable mutation operator. The mutation probability is set to a low value for the intramodule links and to a high value for the intermodule links.

29.3.2.1 Example

Consider a problem of classifying a two-dimensional data into two-classes. The input fuzzifier maps the features into a six-dimensional feature space. Let a sample set of rules obtained from rough set theory be

$$C_1 \leftarrow (L_1 \wedge M_2) \vee (H_2 \wedge M_1), C_2 \leftarrow M_2 \vee H_1, C_2 \leftarrow L_2 \vee L_1,$$

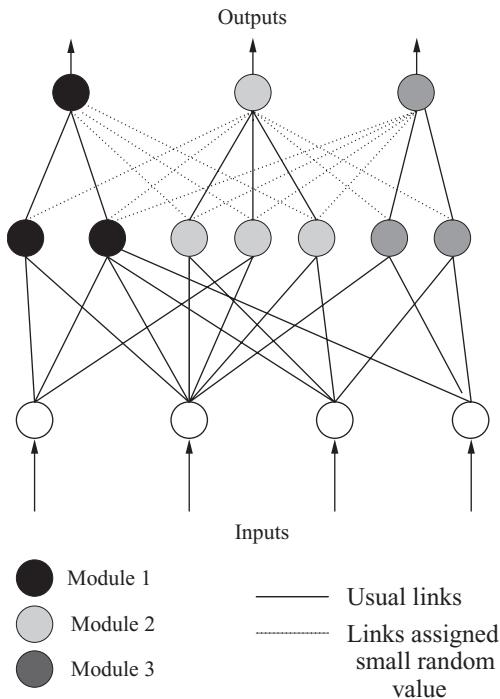


Figure 29.1 Intra- and intermodule links

where L_j , M_j , H_j correspond to $\mu_{low(a_j)}$, $\mu_{medium(a_j)}$, $\mu_{high(a_j)}$, respectively. For the first phase of the GA, three different pools are formed, using one crude subnetwork for class 1 and two crude subnetworks for class 2, respectively. Three partially trained subnetworks result from each of these pools. They are then concatenated to form $(1 \times 2) = 2$ networks. The population for the final phase of the GA is formed with these networks and perturbations about them. The steps followed in obtaining the final network is illustrated in Figure 29.2.

29.3.2.2 Characteristics

Use of this scheme for generating modular knowledge-based networks has several advantages:

- Sufficient reduction in training time is obtained, as the above approach parallelizes the GA to an extent. Because the search string of the GA for subnetworks is smaller, more than linear decrease in searching time is obtained. Also very small number of training cycles are required in the refinement phase, as the network is already very close to the solution.
- The use of rough sets for knowledge encoding provides an established mathematical framework for network decomposition. The search space is reduced, leading to shorter training time. The initial network topology is also automatically determined and provides good *building blocks* for the GA.
- The algorithm indirectly constrains the solution in such a manner that a structure is imposed on the connection weights. This is helpful for subsequent rule extraction from the weights, as the resultant network has sparse but strong interconnection among the nodes.

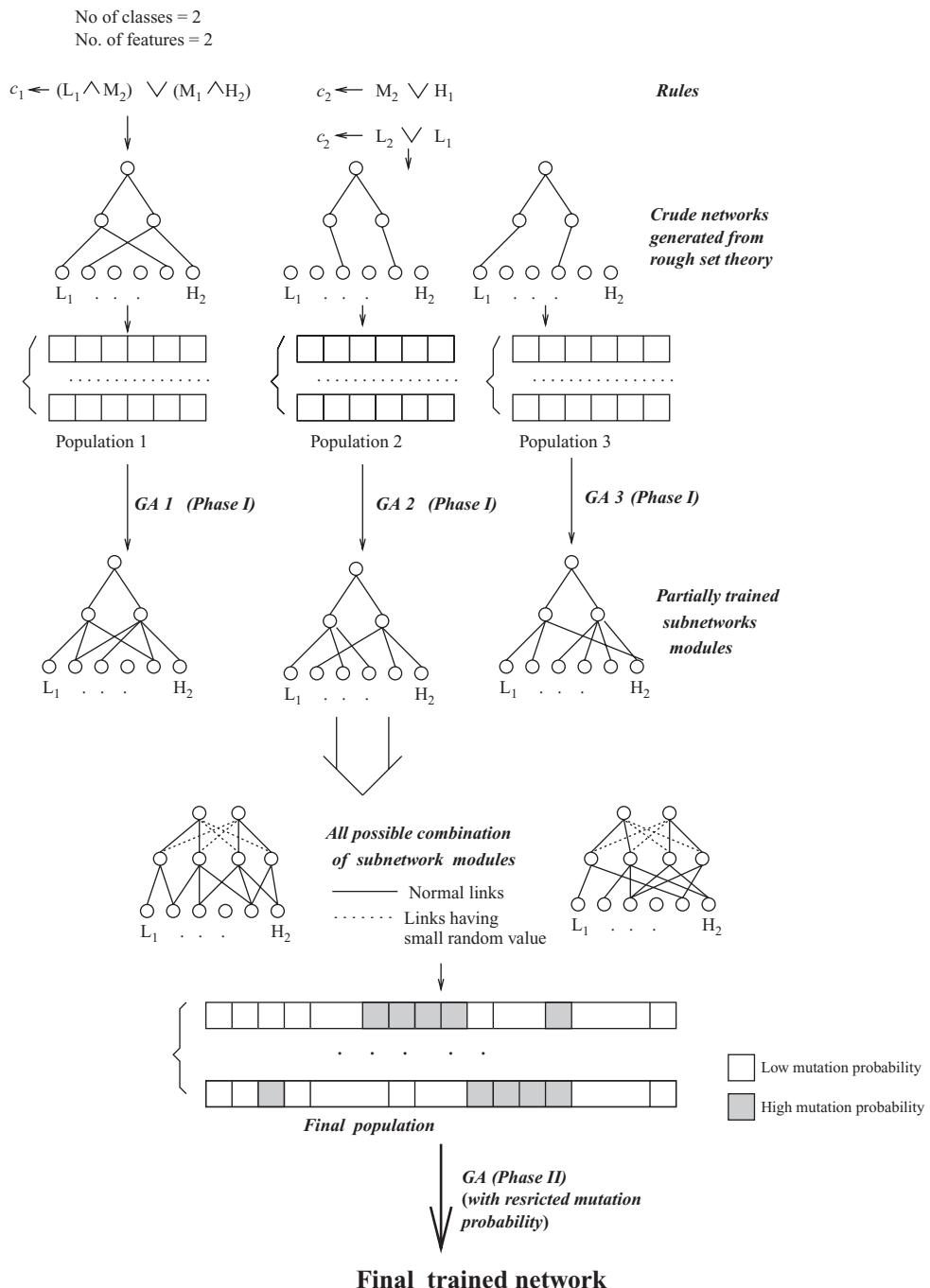


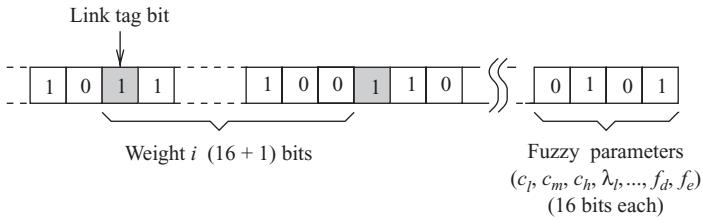
Figure 29.2 Steps for designing a sample modular rough–fuzzy MLP

29.3.3 Evolutionary Design

Here we describe the use of GAs for evolving the weight values as well as the structure of the modular subnetworks. The input and output fuzzification parameters are also tuned. The initial population consists of all possible networks generated from rough set theoretic rules. Typically, GAs involve three basic procedures, namely, (i) encoding of the problem parameters in the form of binary strings, (ii) application of genetic operators like crossover and mutation, and (iii) selection of individuals based on some objective function to create a new population. Each of these aspects is discussed below with relevance to this algorithm [10].

29.3.3.1 Chromosomal Representation

The problem variables consist of the weight values and the input/output fuzzification parameters. Each of the weights is encoded into a binary word of 16-bit length, where [000...0] decodes to -128 and [111...1] decodes to 128 . An additional bit is assigned to each weight to indicate the presence or absence of the corresponding link. If this bit is 0, then the remaining bits are unrepresented in the phenotype. The total number of bits in the string is therefore *dynamic*. Thus a total of 17 bits are assigned for each weight. The fuzzification parameters tuned are the centers (c) and radius (λ) for each of the linguistic attributes *low*, *medium*, and *high* of each feature, and output fuzzifier parameters f_d and f_e . These are also coded as 16-bit strings in the range $[0, 2]$. The chromosome is obtained by concatenating all the above strings. Sample values of the string length are around 2000 bits for reasonably sized networks.



Initial population is generated by coding the networks obtained by rough-set-based knowledge encoding and by random perturbations about them. A population size of 64 was considered.

29.3.3.2 Genetic Operators

Here we provide details on the implementation aspects of the different genetic operators, namely, crossover, mutation, selection, and fitness function used.

Crossover: It is obvious that due to the large string length, single-point crossover would have little effectiveness. Multiple-point crossover is adopted, with the distance between two crossover points being a random variable between 8 and 24 bits. The crossover probability is fixed at 0.7.

Mutation: Because the search string is very large, the influence of mutation is more on the search. Each of the bits in the string is chosen to have some mutation probability ($pmut$), but with a spatiotemporal variation. The mutation probabilities vary along the encoded string, with the bits corresponding to intermodule links being assigned a higher probability as compared to intramodule links. This is done to ensure least alterations in the structure of the individual modules already evolved by incorporating the domain knowledge extracted through rough set theory.

Choice of fitness function: In GAs the fitness function is the final arbiter for string creation, and the nature of the solution obtained depends on the objective function. An objective function of the form described below is chosen:

$$F = \alpha_1 f_1 + \alpha_2 f_2, \quad (7)$$

where

$$f_1 = \frac{\text{No. of correctly classified samples in training set}}{\text{Total no. of samples in training set}},$$

$$f_2 = 1 - \frac{\text{No. of links present}}{\text{Total no. of links possible}}.$$

Here α_1 and α_2 determine the relative weightage of each of the factors. α_1 is taken to be 0.9 and α_2 is taken as 0.1, to give more importance to the classification score as compared to the network size in terms of number of links. Note that we optimize the network connectivity, weights, and input/output fuzzification parameters simultaneously.

Selection: This is done by the *roulette wheel* method. The probabilities are calculated on the basis of ranking of the individuals in terms of the objective function. *Elitism* is incorporated in the selection process by comparing the fitness of the best individual of a new generation to that of the current generation. If the latter has a higher value, then the corresponding individual replaces a randomly selected individual in the new population.

29.3.4 Rule Extraction

A rule extraction algorithm [11], based on the modular hybrid model, is presented here. The performance of the rules is evaluated quantitatively. The steps of the algorithm are provided below [7, 8]:

1. Compute the following quantities:
 $P\text{Mean}$ = mean of all positive weights, $P\text{Thres}_1$ = mean of all positive weights less than $P\text{Mean}$, $P\text{Thres}_2$ = mean of all weights greater than $P\text{Mean}$. Similarly, calculate $N\text{Mean}$, $N\text{Thres}_1$, and $N\text{Thres}_2$ for negative weights.
2. **for** each hidden and output unit
 - (a) **for** all weights greater than $P\text{Thres}_2$, search for positive rules; and **for** all weights less than $N\text{Thres}_2$, search for negative rules, only, by *Subset* method.
 - (b) Search for combinations of positive weights above $P\text{Thres}_1$ and negative weights greater than $N\text{Thres}_2$ that exceed the bias. Similarly, search for negative weights less than $N\text{Thres}_1$ and positive weights below $P\text{Thres}_2$.
3. Associate with each rule j a confidence factor cf_j [11].

$$cf_j = \inf_{j: \text{all nodes in the path}} \frac{\left(\sum_i w_{ji} - \theta_j \right)}{\sum_i w_{ji}}, \quad (8)$$

where w_{ji} is the i th incoming link weight to node j and θ_j is its threshold.

Since the learning algorithm imposes a structure on the network, resulting in a sparse network having few strong links, the $P\text{Thres}$ and $N\text{Thres}$ values are well separated. Hence the above rule extraction algorithm generates most of the embedded rules over a small number of computational steps.

An important consideration is the order of application of rules in a rulebase. Since most of the real-life patterns are noisy and overlapping, rulebases obtained are often not totally consistent. Hence multiple rules may fire for a single example. Several existing approaches apply the rules sequentially [26], often leading to degraded performance. The confidence factors, associated with the extracted rules, help in circumventing this problem.

29.3.5 Results

This genetic rough neuro fuzzy algorithm has been implemented on both real-life (speech, medical) and artificially generated data [10, 23]. In this section we provide sample results on the *Cervical Cancer* data, consisting of a set of 221 patient cases obtained from the database of the Chittaranjan National Cancer

Institute (CNCI), Kolkata. Cross validation of results is made with oncologists. There are four classes corresponding to the Stages I–IV of the cancer (classes C_1 to C_4 , respectively), each containing 19, 41, 139, and 19 patient cases, respectively. The features represent the presence or absence of the symptoms and the signs observed on physical examination. The 21 Boolean input features refer to *Vulva: healthy* ($Vu(h)$), *Vulva: lesioned* ($Vu(l)$), *Vagina: healthy* ($Va(h)$), *Vagina: spread to upper part* ($Va(u)$), *Vagina: spread to middle part* ($Va(m)$), *Vagina: spread to lower part* ($Va(l)$), *Cervix: healthy* ($Cx(h)$), *Cervix: eroded* ($Cx(e)$), *Cervix: small ulcer* ($Cx(su)$), *Cervix: ulcerative growth* ($Cx(u)$), *Cervix: proliferative growth* ($Cx(p)$), *Cervix: ulcero-proliferative growth* ($Cx(l)$), *Paracervix: free* ($PCx(f)$), *Paracervix: infiltrated* ($PCx(i)$), *Urinary bladder base: soft* ($BB(s)$), *Urinary bladder base: hard* ($BB(h)$), *Rectrovaginal septum: free* ($RVS(f)$), *Rectrovaginal septum: infiltrated* ($RVS(i)$), *Parametrium: free* ($Para(f)$), *Parametrium: spread but not upto* ($Para(nu)$), and *Parametrium: spread upto* ($para(u)$), respectively. Rough set theory is applied on the data to extract some knowledge, in the form of dependency rules, which is initially encoded among the connection weights of the subnetworks. The methodology described here is termed Model S. Its performance is compared with that of Model O, an ordinary MLP trained using backpropagation.

29.3.5.1 Classification

Recognition scores obtained by Models S and O are presented in Table 29.1. In all cases, 50% of the samples are used as training set and the remaining samples are used as test set. The dependency rules, as generated via rough set theory and used for encoding crude domain knowledge, are shown in Table 29.2.

These extracted rules are encoded to generate the knowledge-based MLP (Model S). It is observed from Table 29.1 that the performance of Model S is superior to that of Model O.

29.3.5.2 Rule Extraction

We use the algorithm explained in Section 29.3.4 to extract refined rules from the trained network. A sample set of rules extracted from the network is presented in Table 29.3.

Here, we provide the expertise obtained from oncologists. In Stage I, the cancer has spread from the lining of the cervix into the deeper stretches of the connective tissue of the cervix. But it is still confined within the cervix. Stage II signifies the spread of cancer beyond the cervix to nearby areas like parametrial tissue, which are still inside the pelvic area. In Stage III, the cancer has spread to the lower part of the vagina or the pelvic wall. It may be blocking the uterus (tubes that carry urine from the kidneys to the bladder). Stage IV is the most advanced stage of cervical cancer. Now the cancer has spread to other parts of the body, such as rectum, bladder, or lungs. It may be mentioned that the rules generated by this algorithm are validated by the experts' opinion [23].

Table 29.1 Comparative performance of Models S and O, on *Cervical Cancer* data

Stage	Model O		Model S	
	Train	Test	Train	Test
C_1 (%)	65.0	64.7	65.0	64.7
C_2 (%)	69.1	67.7	69.1	68.1
C_3 (%)	93.7	93.0	94.1	90.0
C_4 (%)	42.1	40.1	44.2	41.9
Total (%)	81.0	79.2	81.0	79.5
No. of links	175		118	
Sweeps	90		50	

Table 29.2 Rough set dependency rules for *Cervical Cancer* data

$C_1 \leftarrow Cx(su) \vee Para(f)$
$C_1 \leftarrow Cx(p) \vee Para(f)$
$C_1 \leftarrow Cx(su) \vee Para(nu)$
$C_2 \leftarrow Va(h) \vee Cx(u)$
$C_2 \leftarrow Va(h) \vee Cx(l)$
$C_2 \leftarrow Va(u) \vee Cx(u)$
$C_2 \leftarrow Para(nu)$
$C_2 \leftarrow Pcx(f)$
$C_3 \leftarrow Para(nu)$
$C_3 \leftarrow Para(u)$
$C_3 \leftarrow Va(u)$
$C_3 \leftarrow (Va(u) \wedge Cx(u)) \vee Cx(l) \vee Va(m)$
$C_3 \leftarrow (Va(h) \wedge Cx(u)) \vee (Va(u) \wedge Cx(u)) \vee Cx(l)$
$C_3 \leftarrow (Va(u) \wedge Cx(p)) \vee Va(m) \vee Cx(l)$
$C_4 \leftarrow (Va(l) \wedge Cx(u)) \vee (Cx(u) \wedge Va(u)) \vee (Va(l) \wedge Para(u))$
$C_4 \leftarrow (Va(l) \wedge Cx(p)) \vee Va(m).$

Although the classification performance (Table 29.1) does not demonstrate significant difference, yet the process of knowledge encoding and structured training lead to the imposition of a structure on the weights values. This results in a sparse network having stronger links, whereas the ordinary MLP generates a dense network with moderate and weak links. Hence, the knowledge-based network is better suited for the extraction of crisp and more interpretable rules. This is an advantage in the medical domain, where explanation of the results obtained is required to be available for examination by clinical practitioners.

The performance of the popular C4.5 machine learning system [27] on the data set was also studied as a benchmark. The program gave classification scores of 81.5% on training data and 80.2% on test data. Sample rules generated by C4.5 are

$$\begin{aligned} C_1 &\leftarrow Va(h) \wedge PCx(f) \wedge Para(f) \\ C_2 &\leftarrow Para(f) \\ C_2 &\leftarrow BB(s) \\ C_3 &\leftarrow BB(s) \wedge Para(u). \end{aligned}$$

We observe that the rulebase obtained using the rough-set-based method of knowledge encoding is more complete than that generated using C4.5.

Table 29.3 Rules extracted from trained network for *Cervical Cancer* data

$C_1 \leftarrow (Va(h) \wedge Para(f)) \vee (Cx(h) \wedge Cx(u) \wedge BB(s))$
$C_2 \leftarrow (PCx(f) \wedge PCx(i)) \vee Para(f) \vee Para(nu)$
$C_3 \leftarrow Va(h) \wedge Cx(u) \wedge Cx(l) \wedge Para(u)$
$C_4 \leftarrow Va(m) \vee (Cx(u) \wedge Cx(p)) \vee (Para(nu) \wedge Para(u))$

29.4 Conclusion

Much work has been done on rough–neuro hybridization in the granular computing framework. It ranges from the design of rough neurons, through rough neural networks, right upto rough–neuro–fuzzy integration. The chapter looks back at existing literature in the area, and finally presents some of our own investigations. It is observed that a divide-and-conquer strategy involving modular subnetworks is effective for mining large data sets. Such a methodology, incorporating the four soft computing tools (namely, ANNs, fuzzy sets, GAs, and rough sets), has been used for designing a knowledge-based network for pattern classification and rule generation in the granular computing framework.

The algorithm involves synthesis of several fuzzy MLP modules, each encoding the rough set rules for a particular class. These knowledge-based modules are refined using a GA. The genetic operators are implemented in such a way that they help preserve the modular structure already evolved. It is found that this scheme results in superior performance in terms of classification score, training time, and network sparseness (thereby enabling easier extraction of rules). The extracted rules are compared with some of the related rule extraction techniques on the basis of some quantitative performance indices. It is observed that these rules are less in number, yet accurate, and have high certainty factor and low confusion with less computation time. A decision support system for cervical cancer management is also developed.

References

- [1] Z. Pawlak. *Rough Sets, Theoretical Aspects of Reasoning about Data*. Kluwer Academic, Dordrecht, 1991.
- [2] R. Slowiński (ed.). *Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer Academic, Dordrecht, 1992.
- [3] M. Banerjee and S.K. Pal. *Roughness of a fuzzy set*. *Inf. Sci. Inf. Comput. Sci.* 93 (1996) 235–246.
- [4] S.K. Pal and A. Skowron (eds). *Rough Fuzzy Hybridization: A New Trend in Decision Making*. Springer-Verlag, Singapore, 1999.
- [5] A. Czyzowski and A. Kaczmarek. Speech recognition systems based on rough sets and neural networks. In: *Proceedings of Third Workshop on Rough Sets and Soft Computing (RSSC'94)*, San José, USA, November 10–12, 1994, pp. 97–100.
- [6] R. Yasdi. Combining rough sets learning and neural learning method to deal with uncertain and imprecise information. *Neurocomputing* 7 (1995) 61–84.
- [7] M. Banerjee, S. Mitra, and S.K. Pal. Rough fuzzy MLP: Knowledge encoding and classification. *IEEE Trans. Neural Netw.* 9 (1998) 1203–1216.
- [8] S. Mitra, M. Banerjee, and S.K. Pal. Rough knowledge-based network, fuzziness and classification. *Neural Comput. Appl.* 7 (1998) 17–25.
- [9] M. Sarkar and B. Yegnanarayana. Fuzzy-rough sets and fuzzy integrals in modular neural networks. In: S.K. Pal and A. Skowron (eds), *Rough-Fuzzy Hybridization: New Trends in Decision Making*. Springer-Verlag, Singapore, 1998.
- [10] S. Mitra, P. Mitra, and S.K. Pal. Evolutionary modular design of rough knowledge-based network using fuzzy attributes. *Neurocomputing* 36 (2001) 45–66.
- [11] S.K. Pal, S. Mitra, and P. Mitra. Rough Fuzzy MLP: Modular evolution, rule generation and evaluation. *IEEE Trans. Knowl. Data Eng.* 15 (2003) 14–25.
- [12] L.A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 19 (1997) 111–127.
- [13] W. Su, Y. Su, H. Zhao, and X. Zhang. Integration of rough set and neural network for application of generator fault diagnosis. In: *Proceedings of RSCTC 2004, Uppsala*, Vol. LNAI 3066, 2004, pp. 549–553.
- [14] R. Swiniarski, F. Hunt, D. Chalvet, and D. Pearson. Prediction system based on neural networks and rough sets in a highly automated production process In: *Proceedings of 12th System Science Conference*, Wrocław, Poland, September 12–15, 1995.
- [15] P. Lingras. Rough neural networks. In: *Proceedings of Sixth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 1996)*, Granada, Spain, 1996, pp. 1445–1450.
- [16] J.F. Peters, A. Skowron, L. Han, and S. Ramanna. Towards rough neural computing based on rough membership functions: Theory and applications. In: *Proceedings of RSCTC 2000*, Banff, Canada, October 16–19, 2000, Vol. LNAI 2005, pp. 604–611.

- [17] J.F. Peters, A. Skowron, Z. Suraj, L. Han, and S. Ramanna. Design of rough neurons: Rough set foundation and petri net model. In: *Proceedings of ISMIS 2000*, Charlotte NC, October 11–14, 2000, Vol. LNAI 1932, pp. 283–291.
- [18] A. Skowron and J. Stepaniuk. Information granules and rough-neural computing. In: S.K. Pal, L. Polkowski, and A. Skowron (eds), *Rough-Neural Computing: Techniques for Computing with Words*. Springer-Verlag, Berlin, 2004, pp. 43–84.
- [19] D. Slezak, M. Szczuka, and J. Wroblewski. Feedforward concept networks. In: B. Dunin-Keplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds), *Monitoring, Security, and Rescue Techniques in Multiagent Systems*. Springer-Verlag, Heidelberg, 2005, pp. 281–292.
- [20] L. Polkowski. A rough-neural computation model based on rough mereology. In: S.K. Pal, L. Polkowski, and A. Skowron (eds), *Rough-Neural Computing: Techniques for Computing with Words*. Springer-Verlag, Berlin, 2004, pp. 85–108.
- [21] D. Slezak, J. Wroblewski, and M. Szczuka. Constructing extensions of Bayesian classifiers with use of normalizing neural networks. In: *Proceedings of ISMIS 2003*, Maebashi, Japan, October 28–31, 2003, Vol. LNAI 2871, pp. 408–416.
- [22] S. Mitra and T. Acharya. *Data mining: Multimedia, Soft Computing, and Bioinformatics*. John Wiley, New York, 2003.
- [23] P. Mitra, S. Mitra, and S.K. Pal. Staging of cervical cancer with soft computing. *IEEE Trans. Biomed. Eng.* 47 (2000) 934–940.
- [24] S.K. Pal, B. Dasgupta, and P. Mitra. Rough self-organizing map. *Appl. Intell.* 21 (2004) 289–299.
- [25] S.K. Pal and S. Mitra. *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing*. John Wiley, New York, 1999.
- [26] I.A. Taha and J. Ghosh. Symbolic interpretation of artificial neural networks. *IEEE Trans. Knowl. Data Eng.* 11 (1999) 448–463.
- [27] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

30

Approximation and Perception in Ethology-Based Reinforcement Learning

James F. Peters

30.1 Introduction

An approximation space . . . serves as a formal counterpart of perception ability or observation.

—Ewa Orlowska, March 1982.

The problem considered in this chapter is how to guide run-and-twiddle (RT) forms of reinforcement learning based on perceptual granules that reflect perceived, acceptable behavior patterns. The solution to this problem is made possible by considering behavior patterns of swarms in the context of perceptual granules extracted from approximation spaces. Considerable work on approximation and approximation spaces and their applications has been carried in recent years [1–26]. Zdzislaw Pawlak introduced approximation spaces during the early 1980s as part of his research on classifying objects by means of attributes [7–9], which has recently led to a study of nearness of objects in the context of approximation spaces and near sets [6, 13–15, 27, 28]. Approximation plays a fundamental role in rough set theory also introduced by Pawlak and in the design of approximate actor–critic (AC) methods presented in this chapter.

This chapter presents a continuation of numerous studies of reinforcement learning using rough-set-based granular computing methods (see, e.g., [12, 16, 20, 29–34]). Knowledge representations systems introduced by Zdzislaw Pawlak starting in the early 1970s provide a ground for deriving pattern-based rewards within approximation spaces as well as perceptual granulation. Both conventional and approximation-space-based forms of a Selfridge–Watkins [35–37] RT adaptive control mechanism incorporated in the AC reinforcement learning method are investigated.

It is an age-old adage that experience is a good teacher, and one learns from experience. This is at the heart of reinforcement learning where estimates of the value of an action are based on past experience. One might ask, for example, how to guide action choices by an actor that is influenced by a critic governed by the evaluation of past actions. Specifically, one might ask how to measure the value of an action relative to what has been learned from experience (i.e., from previous patterns of behavior) and how to learn good

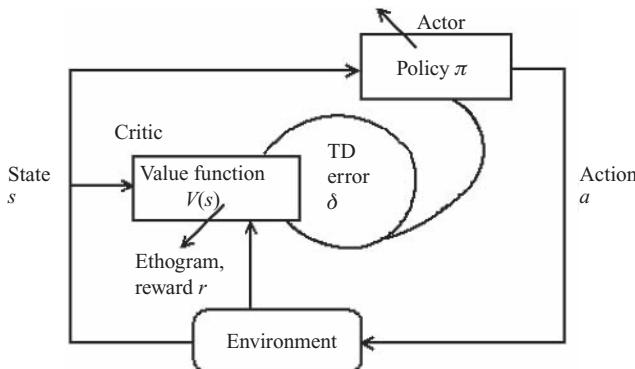


Figure 30.1 Actor–critic architecture with ethogram

policies for choosing rewarding actions. The solution to this problem stems from a rough set approach to reinforcement learning by cooperating agents.

In reinforcement learning, the choice of an action is based on estimates of the value of a state and/or the value of an action in the current state. A swarm learns the best action to take in each state by maximizing a reward signal obtained from the environment. Two different forms of AC method are investigated in this chapter, namely, a conventional AC method and a form of AC method that includes an adaptive learning strategy called RT played out in the context of remembered behavior patterns that accumulate in what are known as ethograms. An actor is governed by a policy function π used to select promising actions and a critic is represented by an estimated value function $V(s)$ used to formulate what is known as (temporal difference) TD error δ that provides a basis for ‘criticizing’ (modifying) the actions made by an actor. The critic can be influenced by a record of past behavior patterns stored in what is known as an ethogram. An approach to formulating TD error is given in Section 30.2. The basic AC architecture is shown in Figure 30.1, which is a slightly modified version of the architecture given in [38].

A rough-set-based *ethogram* [34] is a table of stored behavior patterns (i.e., vectors of measurements associated with behavior features) borrowed from ethology [39, 40]. Quantitative comparisons of past behavior patterns with a template representing ‘normal’ or desirable behavior are carried out within the framework of an approximation space. Approximation spaces were introduced by Zdzisław Pawłak during the early 1980s [7], elaborated in [9, 41], and generalized in [23, 25]. The motivation for considering approximation spaces as an aid to reinforcement learning stems from the fact that it becomes possible to derive pattern-based evaluation of actions chosen during an episode (see, e.g., [33]).

AC methods have been studied extensively (see, e.g., [12, 20, 29, 37, 38, 42–47]). The conventional AC method evaluates whether things have gotten better or worse than expected as a result of an action selection in the previous state. A TD error term δ is computed by the critic to evaluate an action previously selected. An estimated action preference in the current state is then determined by an actor using δ . Swarm actions are generated by a policy that is influenced by action preferences. In the study of swarm behavior of multiagent systems such as systems of cooperating bots, it is helpful to consider ethological methods (see, e.g., [40]), where each proximate cause (stimulus) usually has more than one possible response. Swarm actions with positive TD error tend to be favored. A second form of AC method is defined in the context of an approximation space (see, e.g., [16, 18–20, 23, 25, 48, 49]), and which is an extension of recent work with reinforcement comparison and the AC method (see, e.g., [12, 16, 20, 29, 30, 33, 34]). The form of the proposed AC method utilizes what is known as a reference reward, which is pattern based and action specific. Each action has its own reference reward which is computed within an approximation space that makes it possible to measure the closeness of action-based blocks of equivalent behaviors to a standard. The contribution of this chapter is a framework for RT-based reinforcement learning defined in the context of approximation spaces and perceptual granules.

This chapter is organized as follows. An brief introduction to semimartingales, TD error, and twiddling rules is given in Section 30.2. For the sake of completeness, rough set theory is briefly introduced in

Section 30.3. The basic idea of an approximation space is presented in Section 30.4. Conventional, rough coverage and an RT approximation space-based forms of the AC method as well as experimental results are given in Section 30.5. A comparison of the AC methods is given in Section 30.6.

30.2 Basis for Twiddling: Semimartingale

This section briefly introduces an approach to determining when an organism should consider twiddling (i.e., pausing to consider how to modify its behavior in a satisfactory way) based on what is known as a semimartingale. Informally, a martingale is a discrete-time stochastic process (i.e., sequence of random variables) in which the conditional expectation of the next observation (given all past observations) equals the value of the last observation [50, 51]. The notion of a stochastic process and what is known as semimartingales are important in RT adaptive learning introduced in this chapter.

Definition 1: Stochastic Process. *A stochastic process is any family of random variables $\{X_t, t \in T\}$ [50]. In practice, X_t is an observation at time t . A random variable (r.v.) X_t is a real-valued function $X : \Omega \rightarrow \mathbb{R}$ defined on (Ω, \mathcal{F}) , where Ω, \mathcal{F} is sample space and family of events, respectively [51, 52].*

It can be shown that during each episode of RT AC learning, what is known as a semimartingale is constructed and each semimartingale is finite (see, e.g., [12]). Semimartingales were introduced by Doob during the early 1950s [50] and elaborated by many others (see, e.g., [51, 52]).

Definition 2: Semimartingale. *A semimartingale is a stochastic process $\{X_t, t \in T\}$ such that*

$$E[X_t] \leq E[X_{t+1}],$$

where $E[|X_t|] < \infty$.

The form of semimartingale¹ we have in mind is $\{R_t, t \in T\}$, $E[R_t] \leq E[R_{t+1}]$, where R_t is the return on a sequence of actions at time t during an episode. For example, each time through the loop in Algorithm 3, a term is added to a semimartingale. The for-loop in Algorithm 3 guarantees each semimartingale constructed during an episode ends in a finite time.

Let $r_i \in \mathbb{R}$ denote a numerical reward resulting from action a_i in state s_i . A behavior is defined by a finite state sequence that occurs during an episode that ends at time $t \leftarrow T_m$ in Algorithm 3 and is represented by (1).

$$s_0, a_0, r_1, s_1, a_1, r_2, \dots, s_i, a_i, r_{i+1}, \dots, s_{t-1}, a_{t-1}, r_t, s_t. \quad (1)$$

The return R_t (i.e., cumulative future discounted rewards) on a sequence of actions is defined by (2).

$$R_t = r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^{t-1} r_t = \sum_{m=1}^t \gamma^{t-m} r_m, \quad (2)$$

where $\gamma \in [0, 1]$ is called a discount rate, and r_t is the reward signal from the environment that results from an action performed at time $t - 1$. A basic assumption is that r_t a random variable defined on a set of events Ω , where each $\omega \in \Omega$ is an observable signal from the environment representing a reward for an action performed. For example, Ω can be a finite set of observable Euclidean distances between a moving organism and some goal such as a cache containing food or moving camera in relation to the center of mass of a moving target. As a consequence, R_t is also a random variable, since a function of a random variable is also random [53]. The basic idea during reinforcement learning is to choose actions

¹ The expectation $E[X_{t_n}]$ is normally written as a conditional expectation $E[X_{t_n} | X_{t_1}, \dots, X_{t_{n-1}}]$, $t_1 < \dots < t_{n-1} < t_n$ [50]. For simplicity, here and in the sequel, we write $E[X_{t_n}]$ instead.

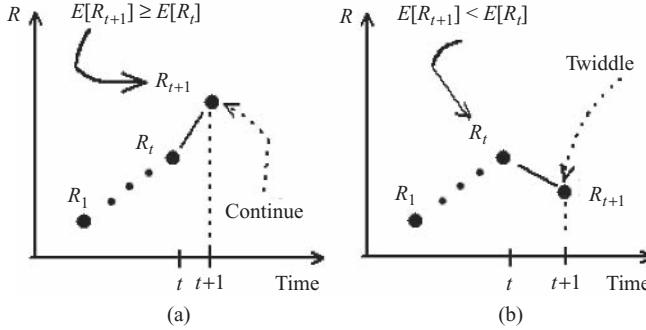


Figure 30.2 Sample Martingales: (a) non-decreasing returns and (b) occasion to twiddle

during an episode that ends in a terminal state at time T so that the expected discounted return $E^\pi(R_t)$ following policy π improves. To do this, it is necessary to estimate the expected value of R_t .

A basic assumption in the study of biologically inspired, artificial ecosystems is that swarms live in a non-stationary environment and a perfect model of the ecosystem environment is not available. Let $Pr(X = x)$ denote the probability that X equals x . It is assumed that the return R (cumulative future discounted rewards) for a sequence of actions is a discrete random variable, and the probability $Pr(R = r)$ is not known. An assumption made here is that the episodic behavior of a swarm yields a stochastic process $\{R_t, t \in T\}$ that is a semimartingale, where

$$E [R_{t+1}] \geq E [R_t], \quad (3)$$

which provides a basis for a stopping time for an episode² and triggers the need to consider a means of improving the returns on chosen actions (i.e., ‘twiddling’) as shown in Figure 30.2b. This leads to Rule 1.

Rule 1: Semimartingale Twiddle Rule. *An organism twiddles whenever the expected return in the next state is not part of a semimartingale; i.e., condition (3) fails to occur. Otherwise, an organism continues its behavior.*

The term *organism*, in general, is understood in Whitehead’s sense as something that emerges from (belongs to) the world [63]. The scenario that provides a basis for Rule 1 is analogous to the situation described by Doob [50], where a gambler finds it is more profitable to gamble on a sequence of returns on bets that converge to some desirable value rather than the color of a card. One can imaginatively conjecture that the situations described by Rule 1 are also analogous to a male silk moth riding the wind while following the scent of perfume emitted by some distant female silk moth [35]. The male moth continues its flight as long as it finds that further flight in the current direction can be ‘expected’ to lead to roughly the same or stronger perfume scent. Carrying this analogy further, it can be assumed that the male silk moth would pause (‘twiddle’) and then change its direction in search of a stronger perfume scent whenever it finds that flight in the current direction yields diminishing returns (condition (3) fails to occur).

In this work, it assumed that the value of a state $V(s)$ is defined by (4).

$$V(s) = E [R_t]. \quad (4)$$

Let $V(s')$ denote the value of the next state s' at time step $t + 1$. In this work, $V(s)$ and $V(s')$ are used in what is known as TD error defined in (5):

$$\delta = r + \gamma V(s') - V(s), \quad (5)$$

² An episode arrives at a terminal state at the time when the expected return for the next state has diminished to the point where (3) fails to occur.

where $\gamma \in [0, 1]$ is called a discount factor. The influence of a semimartingale in (3) is absorbed in the critic, which provides a basis for modifying and, possibly, improving the behavior of an organism. This influence is reflected in Rule 2.

Rule 2: δ Twiddle Rule. *An organism is forced to modify its behavior (twiddle) whenever $\delta_t < 0$.*

30.3 Basic Concepts: Rough Sets

For the sake of completeness and because there are subtle differences between the proposed approach and what is traditionally considered in rough set theory, this section briefly presents some fundamental concepts in rough set theory that provide a foundation for a granular computing approach to reinforcement learning by collections of cooperating agents. The rough set approach introduced by Zdzisław Pawlak [8, 9] provides a ground for concluding to what degree a set of equivalent behaviors are *covered* by a set of behaviors representing a standard. The term ‘coverage’ is used relative to the extent that a given set is contained in a standard set. An overview of rough set theory and its applications is given in [11, 54–61]. For computational reasons, a syntactic representation of knowledge is provided by rough sets in the form of data tables. A data (information) table *IS* is represented by a pair (U, \mathcal{F}) , where U is a non-empty, finite set of elements and \mathcal{F} is a non-empty, finite set of probe functions representing selected features such as contour, color, shape, texture, symmetry, reward, action, state, discount, and learning rate. Let $\mathcal{O}, \mathcal{O}'$ be an observed object and known object, respectively. A *probe function* is a function which is invariant with respect to the function values $f(\mathcal{O}), f(\mathcal{O}')$ in (6).

$$\mathcal{O} \approx \mathcal{O}' \Leftrightarrow f(\mathcal{O}) = f(\mathcal{O}'), \quad (6)$$

which is an incipient model for object recognition. Let \mathcal{O} be a non-empty set of objects \mathcal{O} . In effect, a probe is a mapping $f : \mathcal{O} \rightarrow M$, where M is a set representing a range of measurements associated with a feature.³ In the language of traditional rough set theory, $\mathcal{O}, \mathcal{O}'$ have matching measurements relative to some feature of interest; i.e., the observed measurements are indiscernible from each other. The introduction of probe functions as opposed to object attributes defined by partial functions opens the door to a study of object recognition normally carried out in science and pattern recognition (see, e.g., [64–66]). A probe makes it possible to determine if two objects are associated with the same pattern without necessarily specifying which pattern (classification). A detailed explanation about probe functions vs. attributes in the classification of objects is given in [27].

For present purposes, to each feature there is only one probe function associated and its value set is taken to be a finite set (usually of real numbers). Thus one can identify the set of features with the set of associated probe functions, and hence we use f rather than $f_{\mathcal{F}}$ and call $\mathcal{V}_f = \mathcal{V}_{\mathcal{F}}$ a set of feature values. If \mathcal{F} is a finite set of probe functions for features of elements in U , the pair (U, \mathcal{F}) is called a *data table*, or *information system* (IS).

For each subset $B \subseteq \mathcal{F}$ of probe functions, define the binary relation $\sim_B = \{(x, x') \in U \times U : \forall f \in B, f(x) = f(x')\}$. Since each \sim_B is an equivalence relation, for $B \subset \mathcal{F}$ and $x \in U$, let $[x]_B$ denote the equivalence class, or *block*, containing x ; that is,

$$[x]_B = \{x' \in U : \forall f \in B, f(x') = f(x)\} \subseteq U.$$

If $(x, x') \in \sim_B$ (also written $x \sim_B x'$), then, x and x' are said to be *indiscernible* with respect to all feature probe functions in B or, simply, *B-indiscernible*.

³ The term *feature* identifies something about the appearance of an object that is observable. By contrast, an *attribute* maps each object to a single value and, philosophically, is understood to be a property of an object – something essential or inherent in an object.

Information about a sample $X \subseteq U$ can be approximated from information contained in B by constructing a B -lower approximation

$$B_* X = \bigcup_{x:[x]_B \subseteq X} [x]_B,$$

and a B -upper approximation

$$B^* X = \bigcup_{x:[x]_B \cap X \neq \emptyset} [x]_B.$$

The B -lower approximation $B_* X$ is a collection of blocks of sample elements that can be classified with full certainty as members of X using the knowledge represented by feature probe functions in B . By contrast, the B -upper approximation $B^* X$ is a collection of blocks of sample elements representing both certain and possibly uncertain knowledge about X . Whenever $B_* X \subsetneq B^* X$, the sample X has been classified imperfectly and is considered a rough set. In this chapter, only B -lower approximations are used.

30.4 Approximation Spaces

This section gives a brief introduction to approximation spaces. The basic model for an approximation space was introduced by Pawlak in 1981 [7, 8], elaborated in [1, 2, 4, 5, 9, 41], generalized in [23, 25], extended in [13, 14], and applied in a number of ways (see, e.g., [12, 15–20, 22, 29, 30, 32–34]). An approximation space serves as a formal counterpart of perception or observation [41] and provides a framework for approximate reasoning about vague concepts.

To be precise about what an *approximation space* is, some definitions are required. A *neighborhood function* on a set U is a function $N : U \rightarrow \mathcal{P}(U)$ that assigns to each $x \in U$ some subset of U containing x . A particular kind of neighbourhood function on U is determined by any partition $\xi : U = U_1 \cup \dots \cup U_d$, where for each $x \in U$, the ξ -neighborhood of x , denoted $N_\xi(x)$, is the U_i that contains x . In terms of equivalence relations in Section 30.3, for some fixed $B \subseteq \mathcal{F}$ and any $x \in U$, $[x]_B = N_B(x)$ naturally defines a neighborhood function N_B . In effect, the neighborhood function N_B defines an *indiscernibility* relation, which defines for every object x a set of similarly defined objects, i.e., objects whose feature function value sets agree precisely (see, e.g., [49]). An *overlap function* v on U is any function $v : \mathcal{P}(U) \times \mathcal{P}(U) \rightarrow [0, 1]$ that reflects the degree of overlap between two subsets of U .

A *generalized approximation space* (GAS) is a tuple (U, \mathcal{F}, N, v) , where U is a non-empty set of objects, \mathcal{F} is a set of probe functions representing features of objects in U , N is a neighborhood function on U , and v is an overlap function on U . In this work, only indiscernibility relations determine N .

A set $X \subseteq U$ is *definable* in a GAS if and only if X is the union of some equivalence classes in the codomain of the neighborhood function. Specifically, any information system (U, \mathcal{F}) and any $B \subseteq \mathcal{F}$ naturally defines parameterized approximation spaces $AS_B = (U, \mathcal{F}, N_B, v)$, where $N_B(x) = [x]_B$, a B -indiscernibility class in a partition of U .

A standard example (see, e.g., [23]) of an overlap function is *standard rough inclusion*, defined by $v_{SRI}(X, Y) = \frac{|X \cap Y|}{|X|}$ for non-empty X . Then $v_{SRI}(X, Y)$ measures the portion of X that is included in Y . An analogous notion is used in this work. If U is a set of objects with observable behaviors, let $Y \subseteq U$ represent a kind of ‘standard’ for evaluating sets of objects with similar behaviors. For any $X \subseteq U$, we are interested in how well X ‘covers’ Y , and so we consider another form of overlap function, namely, *standard rough coverage* v_{SRC} , defined by

$$v_{SRC}(X, Y) = \begin{cases} \frac{|X \cap Y|}{|Y|} & \text{if } Y \neq \emptyset, \\ 1 & \text{if } Y = \emptyset. \end{cases} \quad (7)$$

In other words, $v_{SRC}(X, Y)$ returns the fraction of Y that is covered by X . In the case where $X = Y$, then $v_{SRC}(X, Y) = 1$. The minimum coverage value $v_{SRC}(X, Y) = 0$ is obtained when $X \cap Y = \emptyset$. One might note that for non-empty sets, $v_{SRC}(X, Y) = v_{SRI}(Y, X)$.

30.4.1 Deriving Ethogram-Based Average Rough Coverage

This section illustrates how to derive average rough coverage using an ethogram. During a swarm episode, an ethogram is constructed, which provides the basis for an approximation space and the derivation of the degree that a block of equivalent behaviors is covered by a set of behaviors representing a standard (see, e.g., [16, 34, 40]). Let $x_i, s, PC, a, p(s, a), r, d$ denote i th observed behavior, current state, proximate cause [40], possible action in current state, action preference, reward for an action in previous state, and decision (1 = choose action, 0 = reject action), respectively. It should also be observed that probe function $PC : X \rightarrow Causes$ (set of proximate causes), in general, maps X to a set $Causes$ with many possible values representing immediate causes (stimuli) leading an observed behavior, even though only one $PC(x)$ value is given in the snapshot in Table 30.1; i.e., $PC(x) = 3$. The same situation also holds true for state probe function $s : X \rightarrow States$, which maps X to a set $States$ representing many different object states, but only one value of $s(x)$ is represented in Table 30.1.

Assume, for example, $B_a(x) = \{y \in U \mid xIND(B \cup \{a\})y\}$, where U is a set of observed behaviors. Let $\mathcal{B} = \{B_a(x) \mid x \in S\}$ denote a set of blocks representing actions in a set of sample behaviors $S \subseteq U$. Then \bar{v}_a is defined as average rough coverage as shown in (8):

$$\bar{v}_a = \frac{1}{card(\mathcal{B})} \sum_{i=1}^{card(\mathcal{B})} v_B(B_a(x_i), B_*D), \quad (8)$$

where $B_a(x_i) \in \mathcal{B}$. Computing the average lower rough coverage value for action blocks extracted from an ethogram implicitly measures the extent that past actions have been rewarded. What follows is a simple example of how to set up a lower approximation space relative to an ethogram. The calculations are performed on the feature values shown in Table 30.1:

$$\begin{aligned} B &= \{s_i, PC_i, a_i, p(s, a)_i, r_i\}, \\ D &= \{x \in U \mid d(x) = 1\} = \{x_1, x_3, x_5, x_7, x_8\}, \\ B_a(x) &= \{y \in U_{beh} \mid xIND(B \cup \{a\})y\}, \text{ hence} \\ B_{a=4}(x_0) &= \{x_0, x_2, x_4, x_6, x_8\}, B_{a=5}(x_1) = \{x_1\}, \\ B_{a=5}(x_3) &= \{x_3\}, B_{a=5}(x_5) = \{x_5\}, B_{a=5}(x_7) = \{x_7\}, B_{a=5}(x_9) = \{x_9\} \\ B_*D &= \{B_a(x) \mid B_a(x) \subseteq D\} = \{x_1, x_3, x_5, x_7\}, \\ v_B(B_{a=4}(x_0), B_*D) &= 0, v_B(B_{a=5}(x_1), B_*D) = 0.25, \\ v_B(B_{a=5}(x_3), B_*D) &= 0.25, v_B(B_{a=5}(x_5), B_*D) = 0.25, \\ v_B(B_{a=5}(x_7), B_*D) &= 0.25, v_B(B_{a=5}(x_9), B_*D) = 0, \\ \bar{v}_a &= \overline{v_B(B_a(x), B_*D)} = 0.1667. \end{aligned}$$

Table 30.1 Sample ethogram

x_i	s	PC	a	$p(s, a)$	r	d
x_0	1	3	4	0.010	0.010	0
x_1	1	3	5	0.010	0.010	1
x_2	1	3	4	0.010	0.010	0
x_3	1	3	5	0.020	0.011	1
x_4	1	3	4	0.010	0.010	0
x_5	1	3	5	0.031	0.012	1
x_6	1	3	4	0.010	0.010	0
x_7	1	3	5	0.043	0.013	1
x_8	1	3	4	0.010	0.010	1
x_9	1	3	5	0.056	0.014	0

30.4.2 Ethogram-Based Learning Cycle

Based on neighborhoods of objects associated with an approximation space, different forms of learning influenced by the perceived behaviors recorded in episode ethograms are possible. In Figure 30.3, a behavior is defined by the tuple

$$(s, a, r, V(s)),$$

where $V(s)$ is the estimated value of expectation $E[R_t]$. A Monte Carlo method [53, 67] is used to estimate $E[R_t]$, which, in its simplest form, is a running average of the rewards received up to the current state.

The set $N_B(x)$ contains a set of percepts. A *percept* is a by-product of perception, i.e., something that has been observed [68]. For example, a member of $N_B(x)$ represents *what has been perceived about objects belonging to a neighborhood*, i.e., observed objects with matching probe function values. Collectively, $N_B(x)$ represents a *perceptual granule*, a product of perceiving. Perception is defined as the extraction and use of information about one's environment [69]. This basic idea is represented in the *sample objects*, *probe function measurements*, *perceptual neighborhoods*, and *judgmental percepts* columns in Figure 30.3. In this chapter, we focus on the perception of acceptable objects. (This is reflected in the *neighborhood coverage* and *average coverage* columns in Figure 30.3.)

Ethology, the comparative study of behavior, applies to the behavior of animals and humans all those questions asked and methodologies used as a matter of course in all branches of biology since Charles Darwin's time (Konrad Z. Lorenz, *The Foundations of Ethology*, 1981.)

Remark 1: Ethology-Inspired Approach to Learning. The granular computing approach to learning in Figure 30.3 (see Algorithm 2 and Algorithm 3) is rooted in ethology (see, e.g., [39, 40, 70]). The ethological approach to reinforcement learning can be traced back to the introduction of rough ethology [16] that was followed by a series of studies [12, 20, 29–34, 71]. Central to this approach is the archiving of observed behaviors in an ethogram, which has the form of a decision system commonly used in rough set theory [34]. That is, an ethogram represents the decision system $(\mathcal{O}, \mathcal{F}, \{d\})$, where $\mathcal{O}, \mathcal{F}, d$ represent a non-empty set of objects, set of functions representing object features, and decision, respectively. The decision d is a function

$$d : X \times \mathcal{F} \rightarrow \mathfrak{N},$$

where $X \subseteq \mathcal{O}$.

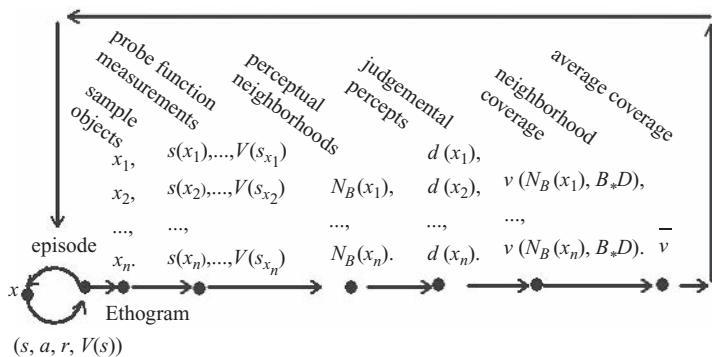


Figure 30.3 Approximation-space-based learning cycle

Assume that $\phi = \{\phi_1, \dots, \phi_L\}$ is a given set of functions representing either features or attributes, where $\phi_i : \mathcal{O} \rightarrow V_i$, and V_i is a value set for ϕ_i for $i = 1, \dots, L$. In combination, the functions representing object features provide a basis for an object description in the form of a vector $\phi : \mathcal{O} \rightarrow V_1 \times \dots \times V_L$ containing measurements (returned values) associated with each functional value $\phi_i(x)$, $x \in \mathcal{O}$ in (9).

$$\textbf{Object description: } \phi(x) = (\phi_1(x), \phi_2(x), \phi_3(x), \dots, \phi_L(x)). \quad (9)$$

Example 1: Sample Object Description. By way of illustration, consider the behavior of an organism (living object) represented by a tuple

$$(s, a, r, V(s), \dots),$$

where $s, a, r, V(s)$ denote organism functions representing state, action, reward for an action, and value of state, respectively. Typically, $V(s) \approx \sum_i r_i$, where r_i is the reward observed in state s_i at instant i of a given time window for an action performed in state s_{i-1} and the sum is over all instants i of the time window. In combination, tuples of behavior function values form the following description of an object x relative to its observed behavior:

$$\textbf{Organism behavior: } \phi(x) = (s(x), a(x), r(x), V(s(x))).$$

For example, in [16, 29], a set of objects X with observed interesting (i.e., acceptable) behavior is approximated after the set of available sample objects has been granulated using rough set approximation methods. Observed organism behavior is episodic and behavior tuples are stored in a decision table called an ethogram, where each observed behavior is assessed with an acceptability decision; i.e., $d(x, \phi) = 1$ (acceptable) and $d(x, \phi) = 0$ (unacceptable) based on evaluation of $V(s)$ for each behavior.

Algorithm 1: Actor–critic method

```

Input : States  $s \in S$ , Actions  $a \in A(s)$ , Initialized  $\alpha, \gamma$ .
Output: Policy  $\pi(s, a)$  //where  $\pi(s, a)$  is a policy in state  $s$  that controls the selection of a particular
action in state  $s$ .
for (all  $s \in S, a \in A(s)$ ) do
     $p(s, a) \leftarrow 0;$ 
     $\pi(s, a) \leftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}};$ 
     $C(s) \leftarrow 0;$ 
end
while True do
    Initialize  $s, T_m$ ;
    for ( $t = 0; t < T_m; t = t + 1$ ) do
        Choose  $a$  from  $s$  using  $\pi(s, a)$ ;
        Take action  $a$ , observe  $r, s'$ ;
         $C(s) \leftarrow C(s) + 1;$ 
         $V(s) \leftarrow \frac{C(s)-1}{C(s)} \cdot V_{C(s)-1}(s) + \frac{1}{C(s)} \cdot r;$ 
         $\delta = r + \gamma V(s') - V(s);$ 
         $p(s, a) \leftarrow p(s, a) + \beta \delta;$ 
         $\pi(s, a) \leftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}};$ 
         $s \leftarrow s';$ 
    end
end

```

30.5 Actor–Critic Methods

AC methods are TD learning methods with a separate memory structure to represent policy independent of the value function used (see Figure 30.1). The AC method considered in this section is an extension of reinforcement comparison in [38]. This extension results from two main ingredients not present in the conventional approach to AC method, namely,

1. *Application of Rule 2 (δ Twiddle Rule) in Algorithm 3.* This hearkens back to original work by Selfridge [35] and Watkins [36] on a control strategy that appears to be a fundamental part of the behavior of biological organisms.
2. *Construction of ethograms during each episode in Algorithm 2 and Algorithm 3.* The basic framework for an ethogram comes from ethology introduced by Tinbergen [40] and Lorenz [70] and later elaborated in the context of approximation spaces (see, e.g., [16, 34]). Each ethogram leads to a particular approximation space and the computation of average rough coverage values relative to acceptable actions performed during an episode (see Figure 30.3).

The following notation is needed (here and in subsequent sections). Let S be a set of possible states, let s denote a (current) state, and for each $s \in S$, let $A(s)$ denote the set of actions available in state s . Put $A = \bigcup_{s \in S} A(s)$, the collection of all possible actions. Let a denote a possible action in the current state; let s' denote the subsequent state after action a (i.e., s' is the state in the next time step); let $p(s, a)$ denote an action preference (for action a in state s); let r denote the reward for an action while in state s .

Begin by fixing a number $\gamma \in (0, 1]$, called a *discount rate*, a number picked that diminishes the estimated value of the next state; in a sense, γ captures the confidence in the expected value of the next state. Let $C(s)$ denote the number of times the actor has observed state s . As is common (e.g., see [38]), define the estimated value function $V(s)$ to be the average of the rewards received while in state s . This average may be calculated by (10).

$$V(s) = \frac{n-1}{n} V_{n-1}(s) + \frac{1}{n} \cdot r, \quad (10)$$

where $V_{n-1}(s)$ denotes $V(s)$ for the previous occurrence of state s . After each action selection, the critic (represented as δ) evaluates the quality of the selected action using

$$\delta \leftarrow r + \gamma V(s') - V(s),$$

which is the error (labeled the TD error) between successive estimates of the expected value of a state. If $\delta > 0$, then it can be said that the expected return received from taking action a at time t is larger than the expected return in state s resulting in an increase to action preference $p(s, a)$. Conversely, if $\delta < 0$, the action a produced a return that is worse than expected and $p(s, a)$ is decreased [47].

The preferred action a in state s is calculated using

$$p(s, a) \leftarrow p(s, a) + \beta \delta,$$

where β is the actor's learning rate. The policy $\pi(s, a)$ is employed by an actor to choose actions stochastically using the Gibbs softmax method [72] (see also [38]).

$$\pi(s, a) \leftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}}.$$

Algorithm 1 gives the AC method that is an extension of the reinforcement comparison method given in [38]. It is assumed that the behavior represented by Algorithm 1 is episodic (with length T_m , an abuse of notation used [73] for terminal state, the last state in an episode) and the while loop in the algorithm is executed continually over the entire learning period, not just for a fixed number of episodes.

If $\delta > 0$, then it can be said that the expected return received from taking action a_t is larger than the expected return in state s_t , which results in an increase to action preference $p(s, a)$. Conversely, if $\delta < 0$, the action a_t produced a return that is worse than expected and $p(s, a)$ is decreased [47]. The RT method [35, 36] is a control strategy inspired by behavior that has been observed in biological organisms such as *E. coli* (*Escherichia coli*) bacteria and silk moths, where an organism continues its current action until the strength of a signal obtained from the environment falls below an acceptable level and then it ‘twiddles’ (i.e., works out a new action strategy). This idea can be applied to the value δ in Algorithm 1. When $\delta < 0$ occurs too often, then it can be said that the agent is performing below expectation and that a ‘twiddle’ is necessary to improve the current situation.

Algorithm 1 gives the AC method that is an extension of the reinforcement comparison method given in [38]. It is assumed that the behavior represented by Algorithm 1 is episodic and is executed continuously.

30.5.1 AC Methods Using Rough Coverage

This section introduces what is known as a rough-coverage actor–critic (RAC) method. The preceding section is just one example of AC methods [38]. In fact, common variations include additional factors which vary the amount of credit assigned to selected actions. This is most commonly seen in calculating the preference, $p(s, a)$. The rough inclusion form of the AC method calculates preference values as shown in (11):

$$p(s, a) \leftarrow p(s, a) + \beta [\delta - \bar{v}_a], \quad (11)$$

where \bar{v}_a is reminiscent of the idea of a reference reward used during reinforcement comparison. Recall that incremental reinforcement comparison uses an incremental average of all recently received rewards as suggested in [38]. By contrast, rough coverage reinforcement comparison (RCRC) uses average

Algorithm 2: Rough coverage actor–critic method

Input : States $s \in S$, Actions $a \in A(s)$, Initialized $\alpha, \gamma, \beta, \bar{v}_a$.

Output: Ethogram.

for (all $s \in S, a \in A(s)$) **do**

$p(s, a) \leftarrow 0;$
 $\pi(s, a) \leftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}};$
 $C(s) \leftarrow 0;$

end

while *True* **do**

Initialize s, T_m ;

for ($t = 0; t < T_m; t = t + 1$) **do**

Choose a from s using $\pi(s, a)$;
 Take action a , observe r, s' ;
 $C(s) \leftarrow C(s) + 1$;
 $V(s) \leftarrow \frac{C(s)-1}{C(s)} \cdot V_{C(s)-1}(s) + \frac{1}{C(s)} \cdot r$;
 $\delta = r + \gamma V(s') - V(s)$;
 $p(s, a) \leftarrow p(s, a) + \beta [\delta - \bar{v}_a]$;
 $\pi(s, a) \leftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}}$;

$s \leftarrow s'$;

end

Extract ethogram table $IS_{swarm} = (U_{beh}, A, d)$;

Discretize feature values in IS_{swarm} ;

Compute \bar{v}_a as in Eq. 8 using IS_{swarm} ;

end

rough coverage of selected blocks in the lower approximation of a set [34]. Intuitively, this means action probabilities are now governed by the coverage of an action by a set of equivalent actions which represent a standard. Rough coverage values are defined within a lower approximation space. Algorithm 2 is the RAC learning algorithm used in the ecosystem for AC methods using lower rough coverage.

In Algorithm 2, $\frac{1}{C(s)}$ is a very primitive model for learning rate. A more advanced, differential learning rate model based on average coverage is proposed in [12].

30.5.2 Twiddling and Information Granulation

In this work, each instance of a twiddle (effort to improve its behavior) by an organism leads to granulation of available information, which is stored in an ethogram. The granules derived from an ethogram are in the form of neighborhoods (classes of equivalent behaviors). The basic idea is to measure the degree of overlap of each neighborhood with the objects in a decision class containing objects that have been judged to be acceptable. The average degree of overlap of the blocks of past behaviors provides a basis for a twiddle factor \bar{v}_a defined by (8).

Remark 2: Approach to Granulation. *The approach to information granulation in Algorithm 2 and in Algorithm 3 works on two different levels. The coverage value*

$$v([x]_B, B_*D), [x]_B = N_B(x), x \in X,$$

*is computed relative to each of the elementary sets (granules which are classes in the indiscernibility relation \sim_B partition of X (perceptual neighborhood level)). The computation of coverage values also utilizes a complex information granule represented by the lower approximation B_*D . This approach to information granulation during reinforcement learning is reflected in the cycle shown in Figure 30.3.*

30.5.3 Run-and-Twiddle Actor–Critic Method

This section briefly presents an RT form of AC method in Algorithm 3 that appears to outperform the conventional AC method in Algorithm 1. Both methods use preference values to compute action selection probabilities. A ‘twiddle’ entails advancing the window of the ethogram (recorded behavior patterns of ecosystem organisms) and recalibrating $\bar{v}_a \forall a \in A$. This form of twiddling mimics, for example, the behavior of *E. coli* bacteria (diminishing food supply results in change in movement) or male silk moth following the perfume emitted by a female silk moth (diminishing perfume signal results in a change of the search path) [35]. A threshold th is used in Algorithm 3 to limit the number of times that $\delta < 0$ before the end of an episode. In addition, a basic assumption in Algorithm 3 is that the number of episodes in the behavior of an organism is unbounded. For this reason, the outer loop in Algorithm 3 is governed by the condition *True*.

Remark 3: Approach to RT. *The approach to RT in Algorithm 3 differs from the approach in [29], where Watkins’ condition $V(s') < V(s)$ is used to determine when to twiddle, i.e., pause and recalibrate \bar{v}_a . By contrast, the condition $\delta < 0$ is used in Algorithm 3 to determine when to twiddle. Notice, also, that many variations of Algorithm 3 are possible. One variation, in particular, is of interest. That is, it is possible to move the if – then block immediately after computing δ so the preference $p(s, a)$ and consequent $\pi(s, a)$ are computed after a new \bar{v}_a has been computed.*

Experiments with the AC method in Algorithm 1, the RAC method in Algorithm 2 and the RT AC method in Algorithm 3 have been compared. The test results for all three forms of AC methods are given in Figures 30.4 and 30.5, which suggests that the RT AC method does better than the other two forms of AC method in adjusting the action policy to yield favorable results. In effect, the test results suggest that

Algorithm 3: RT actor–critic method

Input : States $s \in S$, Actions $a \in A(s)$, Initialized $\alpha, \gamma, \bar{v}_a, th$.

Output: Ethogram.

```

for (all  $s \in S, a \in A(s)$ ) do
     $p(s, a) \leftarrow 0;$ 
     $\pi(s, a) \leftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}};$ 
     $C(s) \leftarrow 0;$ 
end
while True do
    Initialize  $s$ ;
     $v \leftarrow 0$ ;
    for ( $t = 0; t < T_m; t = t + 1$ ) do
        Choose  $a$  from  $s$  using  $\pi(s, a)$ ;
        Take action  $a$ , observe  $r, s'$ ;
         $C(s) \leftarrow C(s) + 1$ ;
         $V(s) \leftarrow \frac{C(s)-1}{C(s)} \cdot V_{C(s)-1}(s) + \frac{1}{C(s)} \cdot r$ ;
         $\delta = r + \gamma V(s') - V(s)$ ;
         $p(s, a) \leftarrow p(s, a) + \delta \bar{v}_a$ ;
         $\pi(s, a) \leftarrow \frac{e^{p(s,a)}}{\sum_{b=1}^{|A(s)|} e^{p(s,b)}}$ ;
        if  $\delta < 0$  then
             $v \leftarrow v + 1$ ;
            if  $v = th$  then
                Extract ethogram table  $DT_{swarm} = (U_{beh}, A, d)$ ;
                Discretize feature values in  $DT_{swarm}$ ;
                Compute  $\bar{v}_a \forall a \in A$ ;
            end
        end
         $s \leftarrow s'$ ;
    end
end

```

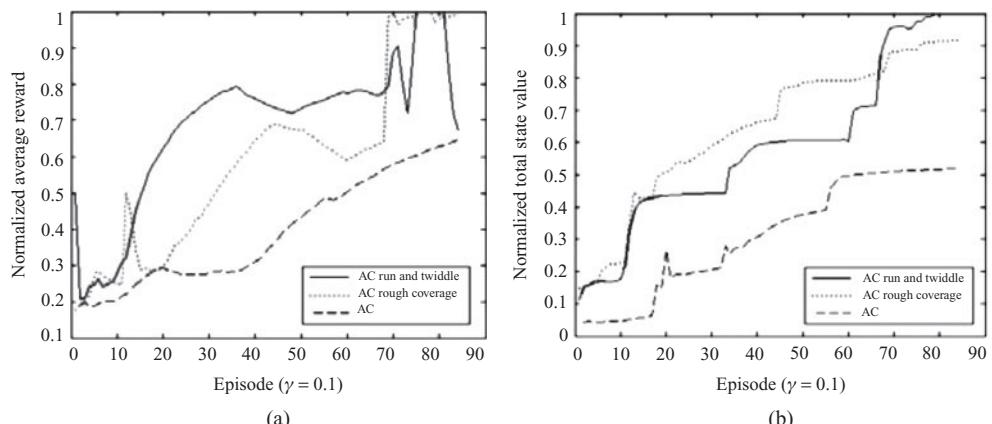


Figure 30.4 Actor–critic method test results, $\gamma = 0.1$: (a) average rewards ($\gamma = 0.1$), (b) state values ($\gamma = 0.1$)

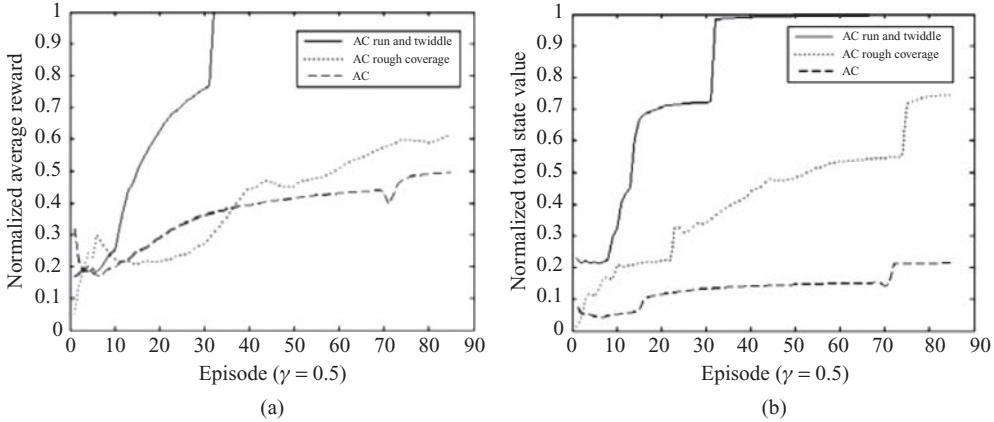


Figure 30.5 RT actor–critic method test results, $\gamma = 0.5$: (a) average rewards, (b) state values ($\gamma = 0.5$)

it is beneficial to granulate the information extracted from successive ethograms during an episode and renew the value of \bar{v}_a whenever the occurrence of an underlying semimartingale leads to negative values of δ in the critic and the advent of twiddling (Rule 1). Experimental evidence suggests that Rule 2 will often be applied repeatedly during each episode. By contrast, notice that \bar{v}_a is computed only once during each episode in the rough coverage algorithm presented in Algorithm 2. The details about the design of the artificial ecosystem and construction of ethograms provide a basis for the experimental results reported in this chapter, can be found in [16, 20, 31, 32, 34].

30.6 Comparison of the AC Methods

In this section, an explanation of why the rough coverage forms of AC represented by Algorithm 2 (RCAC) and Algorithm 3 RT (RTAC) outperform the conventional AC approach. Thanks to granular computing made possible by the approximation space derived from an ethograms at the end of each episode, it can be shown that both methods are more discriminating than the conventional actor critic. To see this, consider the following cases for RCAC. We know that β, δ, \bar{v}_a each have values in the interval $[0, 1]$. There are two cases to consider.

Case 1. Assume $\delta \geq \bar{v}_a$, and without loss of generality (wlog), assume $\beta \geq \delta$. Then observe

$$\beta \geq \delta \geq \bar{v}_a, \quad (12)$$

$$\beta \geq \delta - \bar{v}_a \geq 0, \quad (13)$$

$$\beta \cdot \delta \geq \beta \cdot (\delta - \bar{v}_a) \geq 0, \quad (14)$$

$$p(s, a) + \beta \cdot \delta \geq p(s, a) + \beta \cdot (\delta - \bar{v}_a) \geq p(s, a). \quad (15)$$

Hence, for \bar{v}_a (i.e., action a has low average acceptability during an episode), this means that the policy value for

$$\pi(s, a) \leftarrow \frac{e^{p(s, a)}}{\sum_{b=1}^{|A(s)|} e^{p(s, b)}}$$

will be higher in the conventional case (i.e., using $p(s, a) + \beta \cdot \delta$) than the case where

$$p(s, a) + \beta \cdot (\delta - \bar{v}_a),$$

which provides a basis for a policy decision to reject an action a that has low acceptability, i.e., low average coverage relative a standard for acceptability provided by the lower approximation B_*D in (8). In effect, action a is almost surely less likely to be chosen in the RCAC approach.

Case 2. Assume $\delta \leq \bar{v}_a$, and wlog, assume $\beta > 0$. Then observe

$$\delta \leq \quad \quad \quad \delta, \quad (16)$$

$$\delta - \bar{v}_a \leq \quad \quad \quad \delta, \quad (17)$$

$$\beta \cdot (\delta - \bar{v}_a) \leq \quad \quad \quad \beta \cdot \delta, \quad (18)$$

$$p(s, a) + \beta \cdot (\delta - \bar{v}_a) \leq p(s, a) + \beta \cdot \delta. \quad (19)$$

Hence, for \bar{v}_a (i.e., action a has high average acceptability during an episode), this means that for the policy

$$\pi(s, a) \leftarrow \frac{e^{p(s, a)}}{\sum_{b=1}^{|A(s)|} e^{p(s, b)}},$$

the standard AC policy value will again be higher or the same as the RCAC policy (i.e., using $p(s, a) + \beta \cdot \delta$). In effect, a favorable action a will almost most surely be chosen in both the conventional AC and RCAC approaches.

Consideration of case 1 and case 2 leads to Theorem 1.

Theorem 1. Less acceptable actions are less likely to be chosen in the RC actor–critic approach than in the conventional actor critic.

A similar line of reasoning for RTAC leads to Theorem 2.

Theorem 2. Less acceptable actions are less likely to be chosen in the RT actor–critic approach than in the conventional actor critic.

It can be observed that actions that have lower average coverage are considered less acceptable and are less likely to be chosen in either the RC or the RT AC approach than in the conventional AC approach. In effect, Theorems 1 and 2 provide a basis for explaining why the approximation-space-based AC method does better than the conventional AC approach. This also explains why the RC AC provides a better control strategy than the conventional AC in, for example, controlling movements of a device such as a digital camera during target tracking in noisy environments (see, e.g., [30]), since less acceptable actions are not as likely to be chosen.

30.7 Conclusion

Information granulation is at work in the rough coverage as well as in the RT AC algorithms presented in this chapter. The averaging of the degrees overlap of each of the information granules extracted from an ethogram during an episode in the behavior of an organism provides a basis for improving an organism's behavior. In both cases, the designs of the two variants of the AC method considered in this chapter are extensions of the reinforcement comparison method introduced by Sutton and Barto [38]. Behavior modification in the RT AC algorithm results from the occurrence of an interrupted (i.e., terminated) semimartingale, which leads to recalibration of average action coverages relative to a standard of behavior derived from a particular approximation space extracted from an ethogram that reveals acceptable as well as unacceptable actions. Both rough coverage and RT forms of AC have short-term memory constituted by ethogram (behavior pattern) tables during episodes in the lifespan of organisms

in an ecosystem. Future work will include consideration of other forms of adaptive learning control strategies.

Acknowledgments

The author gratefully acknowledges the suggestions and observations made by the anonymous reviewers, David Gunderson and Andrzej Skowron concerning this work, and the implementation of the AC algorithms by Christopher Henry and Dan Lockery. This research has been supported by Natural Sciences and Engineering Research Council of Canada (NSERC) grant 185986 and research grants T277, T137, T247, T260 from Manitoba Hydro.

References

- [1] A. Gomolińska. Approximation spaces based on similarity and dissimilarity. In: G. Lindemann, H. Schlingloff, H.-D. Burkhard, L. Czaja, W. Penczek, A. Salwicki, A. Skowron, and Z. Suraj (eds), *Concurrency, Specification and Programming (CS&P'06)*. Infomatik-Berichte, Nr. 206, Humboldt University, Berlin, 2006, pp. 446–457.
- [2] A. Gomolińska. Possible rough ingredients of concepts in approximation spaces. *Fundam. Inf.* 72 (2006) 139–154.
- [3] A. Gomolińska. Satisfiability and meaning of formulas and sets of formulas in approximation spaces. *Fundam. Inf.* 67 (1–3) (2005) 77–92.
- [4] A. Gomolińska. Rough validity, confidence, and coverage of rules in approximation spaces. *Transactions on Rough Sets III, LNCS 3400*. Springer, Heidelberg, 2005, pp. 57–81.
- [5] A. Gomolińska. Satisfiability and meaning in approximation spaces. In: G. Lindemann, H.-D. Burkhard, L. Czaja, A. Skowron, H. Schlingloff, and Z. Suraj (eds), *Concurrency, Specification and Programming (CS&P'2004)*. Infomatik-Berichte, Nr. 170, ISSN 0863-095X, Humboldt-Universität zu Berlin, 2004, pp. 229–240.
- [6] C. Henry and J.F. Peters. Image pattern recognition using near sets. In: *Proceedings of Eleventh International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC 2007), Joint Rough Set Symposium (JRS 2007), Toronto May 14–16, 2007*. Springer, Berlin, 2007, Lecture Notes in Artificial Intelligence, Vol. 4482, pp. 475–482.
- [7] Z. Pawlak. *Classification of Objects by Means of Attributes*. Polish Academy of Sciences Report 429, Institute for Computer Science. March, 1981.
- [8] Z. Pawlak. *Rough Sets*. Polish Academy of Sciences Report 431, Institute for Computer Science. March, 1981.
- [9] Z. Pawlak. Rough sets. *Int. J. Comput. Inf. Sci.* 11 (1982) 341–356.
- [10] Z. Pawlak. *Rough Sets – Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.
- [11] Z. Pawlak and A. Skowron. Rudiments of rough sets. *Inf. Sci. Int. J.* 177(1) (2007) 3–27.
- [12] J.F. Peters. Toward approximate adaptive learning. In: *International Conference on Rough Sets and Emerging Intelligent Systems Paradigms in Memoriam Zdzisław Pawlak (RSEISP'07)*, Lectures Notes in Artificial Intelligence 4585, Warsaw, June 28–30, 2007, pp. 57–68.
- [13] J.F. Peters, A. Skowron, and J. Stepaniuk. Nearness of objects: Extension of approximation space model. *Fundam. Inf.* 79 (2007) 1–24.
- [14] J.F. Peters. Near sets. Special theory about nearness of objects. *Fundam. Inf.* 75(1–4) (2007) 407–433.
- [15] J.F. Peters. Near sets. Toward approximation space-based object recognition. In: *Proceedings of Eleventh International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC 2007), Joint Rough Set Symposium (JRS 2007), Toronto May 14–16, 2007*. Springer, Berlin, 2007, Lecture Notes in Artificial Intelligence, Vol. 4482, pp. 22–33.
- [16] J.F. Peters. Rough ethology: Towards a biologically-inspired study of collective behavior in intelligent systems with approximation spaces. *Trans. Rough Sets LNCS 3400* (2005) 153–174.
- [17] J.F. Peters and C. Henry. Approximation spaces in off-policy Monte Carlo learning. *Eng. Appl. Artif. Intell.* 20(5) (2007) 667–675.
- [18] J.F. Peters. Approximation space for intelligent system design patterns. *Eng. Appl. Artif. Intell.* 17(4) (2004) 1–8.
- [19] J.F. Peters. Approximation spaces for hierarchical intelligent behavioral system models. In: B.D.-Kepliż, A. Jankowski, A. Skowron, and M. Szczuka (eds.), *Monitoring, Security and Rescue Techniques in Multiagent Systems, Advances in Soft Computing*. Physica-Verlag, Heidelberg, 2004, pp. 13–30.
- [20] J.F. Peters and C. Henry. Reinforcement learning with approximation spaces. *Fundam. Inf.* 71(2–3) (2006) 323–349.

- [21] S. Ramanna, J.F. Peters, and A. Skowron. Generalized conflict and resolution model with approximation spaces. In: S. Greco, Y. Hata, S. Hirano, M. Inuiguchi, S. Miyamoto, and H.S. Nguyen (eds), *Rough Sets and Current Trends in Computing (RSCTC'2006)*, LNAI 4259. Springer, Berlin, Heidelberg, New York, 2006, pp. 274–283.
- [22] A. Skowron, R. Swiniarski, and P. Synak. Approximation spaces and information granulation. *Trans. Rough Sets III* (2005) 175–189.
- [23] A. Skowron and J. Stepaniuk. Generalized approximation spaces. In: T.Y. Lin and A.M. Wildberger (eds), *Soft Computing*. Simulation Councils, San Diego, 1995, pp. 18–21.
- [24] A. Skowron, J. Stepaniuk, J.F. Peters, and R. Swiniarski. Calculi of approximation spaces. *Fundam. Inf.* 72(1–3) (2006) 363–378.
- [25] J. Stepaniuk. Approximation spaces, reducts and representatives. In: L. Polkowski and A. Skowron (eds), *Rough Sets in Knowledge Discovery 2, Studies in Fuzziness and Soft Computing 19*. Springer-Verlag, Heidelberg, 1998, pp. 109–126.
- [26] M. Wolski. Similarity as nearness: Information quanta, approximation spaces and nearness structures. In: G. Lindemann, H. Schlingloff, H.-D. Burkhard, L. Czaja, W. Penczek, A. Salwicki, A. Skowron, and Z. Suraj, (eds), *Concurrency, Specification and Programming (CS&P'06)*. Informatik-Berichte, Nr. 206, Humboldt University, Berlin, 2006, pp. 424–433.
- [27] J.F. Peters. Classification of objects by means of features. In: D. Fogel, G. Greenwood, and T. Cholewo (eds), *Proceedings of 2007 IEEE Symposium Series on Foundations of Computational Intelligence (IEEE SSCI 2007)*. IEEE, Honolulu, Hawaii, 2007, p. 18.
- [28] J.F. Peters, A. Skowron, and J. Stepaniuk. Nearness in approximation spaces. In: G. Lindemann, H. Schlinglof, H.-D. Burchard, L. Czaja, W. Penczek, A. Sawicki, A. Skowron, and Z. Suraj (eds), *Concurrency, Specification & Programming 2006 (CS & P 2006)*, Wandlitz, Germany, September 27–29, 2006, Vol. 3. Humboldt-Universität zu Berlin, Informatik-Berichte, Vol. 206, 2006, pp. 434–445.
- [29] J.F. Peters, C. Henry, and D. Gunderson. Biologically-inspired adaptive learning control strategies. *Int. J. Hybrid Intell. Syst.* 4 (2007) 1–14.
- [30] J.F. Peters, M. Borkowski, C. Henry, D. Lockery, D. Gunderson, and S. Ramanna. Line-crawling bots that inspect electric power transmission line equipment. In: *Proceedings of 3rd International Conference on Autonomous Robots and Agents 2006 (ICARA 2006)*, Palmerston North, NZ, 2006, pp. 39–44.
- [31] J.F. Peters. Approximation spaces in off-policy Monte Carlo learning. Plenary paper in T. Burczynski, W. Cholewa, and W. Moczulski (eds), *Recent Methods in Artificial Intelligence Methods*, AI-METH Series. Gliwice, 2005, pp. 139–144.
- [32] J.F. Peters, D. Lockery, and S. Ramanna. Monte Carlo off-policy reinforcement learning: A rough set approach. In: *Proceedings of Fifth International Conference on Hybrid Intelligent Systems*, Rio de Janeiro, Brazil, November 6–9, 2005, pp. 187–192.
- [33] J.F. Peters, C. Henry, and S. Ramanna. Reinforcement learning with pattern-based rewards. In: *Proceedings of Fourth International IASTED Conference on Computational Intelligence (CI 2005)*, Calgary, Alberta, Canada, July 4–6, 2005, pp. 267–272.
- [34] J.F. Peters, C. Henry, and S. Ramanna. Rough Ethograms: Study of intelligent system behavior. In: M.A. Kłopotek, S. Wierzchoń, and K. Trojanowski (eds), *New Trends in Intelligent Information Processing and Web Mining (IIS05)*, Gdańsk, Poland, June 13–16, 2005, pp. 117–126.
- [35] O.G. Selfridge. Some themes and primitives in ill-defined systems. In: O.G. Selfridge, E.L. Rissland, and M.A. Arbib (eds), *Adaptive Control of Ill-Defined Systems*. Plenum Press, London, 1984.
- [36] C.J.C.H. Watkins. *Learning from Delayed Rewards*. Ph.D. Thesis, supervisor: Richard Young. King's College, University of Cambridge, UK, May 1989.
- [37] C.J.C.H. Watkins and P. Dayan. Technical note: Q-learning. *Mach. Learn.* 8 (1992) 279–292.
- [38] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
- [39] N. Tinbergen. *Social Behaviour in Animals*, 2nd ed. The Scientific Book Club, London, 1953, 1965.
- [40] N. Tinbergen. On aims and methods of ethology. *Zeitschrift für Tierpsychologie* 20 (1963) 410–433.
- [41] E. Orlowska. *Semantics of Vague Concepts. Applications of Rough Sets*. Polish Academy of Sciences Report 469. Institute for Computer Science, March, 1982.
- [42] A.G. Barto, R.S. Sutton, and C.W. Anderson. Neuronlike elements that can solve difficult problems. *IEEE Trans. Syst. Man Cybern.* 13 (1983) 834–846.
- [43] H.R. Berenji. A convergent actor–critic-based FRL algorithm with application to power management of wireless transmitters. *IEEE Trans. Fuzzy Syst.* 11(4) (2003) 478–485.
- [44] B.P. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.
- [45] V.R. Konda and J.N. Tsitsiklis. Actor–critic algorithms. *Advances in Neural Information Processing Systems* 12 (2000) 1008–1014.

- [46] M.T. Rosenstein. *Learning to Exploit Dynamics for Robot Motor Coordination*. Ph.D Thesis, supervisor: A.G. Barto, University of Massachusetts Amherst, 2003.
- [47] P. Wawrzynski. *Intensive Reinforcement Learning*. Ph.D. Dissertation, supervisor: Andrzej Pacut. Institute of Control and Computational Engineering, Warsaw University of Technology, May 2005.
- [48] J.F. Peters and S. Ramanna. Measuring acceptance of intelligent system models. In: M. Gh. Negoita *et al.* (eds), *Knowledge-Based Intelligent Information and Engineering Systems*, Lecture Notes in Artificial Intelligence, 3213, Part I, 2004, pp. 764–771.
- [49] J.F. Peters, A. Skowron, P. Synak, and S. Ramanna. Rough sets and information granulation. In: T. Bilgic, D. Baets, and O. Kaynak (eds), Tenth International Fuzzy Systems Association. *World Congress IFSA, Istanbul, Turkey, Lecture Notes in Artificial Intelligence* 2715. Physica-Verlag, Heidelberg, 2003, pp. 370–377.
- [50] D.L. Doob. *Stochastic Processes*. Wiley, NY, 1953.
- [51] D. Williams. *Probability with Martingales*. Cambridge University Press, UK, 1991.
- [52] M. Mitzenmacher and E. Upfal. *Probability and Computing. Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, 2005.
- [53] J.M. Hammersley and D.C. Handscomb. *Monte Carlo Methods*. Methuen & Co Ltd, London, 1964.
- [54] Z. Pawlak. Rough classification. *Int. J. Man-Mach. Stud.* 20(5) (1984) 469–483.
- [55] Z. Pawlak. On conflicts. *Int. J. Man-Mach. Stud.* 21 (1984) 127–134.
- [56] Z. Pawlak. *On Conflicts (in Polish)*. Polish Scientific Publishers, Warsaw, 1987.
- [57] Z. Pawlak. Anatomy of conflict. *Bull. Eur. Assoc. Theor. Comput. Sci.* 50 (1993) 234–247.
- [58] Z. Pawlak. An inquiry into anatomy of conflicts. *J. Inf. Sci.* 109 (1998) 65–78.
- [59] Z. Pawlak and A. Skowron. Rough sets: Some extensions. *Inf. Sci. Int. J.* 177(1) (2007) 28–40.
- [60] Z. Pawlak and A. Skowron. Rough sets and Boolean reasoning. *Inf. Sci. Int. J.* 177(1) (2007) 41–73.
- [61] L. Polkowski. Rough sets. *Mathematical Foundations*. Springer-Verlag, Heidelberg, 2002.
- [62] L. Polkowski and A. Skowron (eds). *Rough Sets in Knowledge Discovery 2, Studies in Fuzziness and Soft Computing 19*. Springer-Verlag, Heidelberg, 1998.
- [63] A.N. Whitehead. *Process and Reality*. Macmillan, UK, 1929.
- [64] M. Pavel. *Fundamentals of Pattern Recognition*, 2nd ed. Marcel Dekker, Inc., New York, 1993.
- [65] S.Z. Der and R. Chellappa. *Probe Based Recognition of Targets in Infrared Images*. Research Report CAR-TR-693. Center for Automation Research, November 1993.
<https://drum.umd.edu/dspace/bitstream/1903/398/2/CS-TR-3174.pdf>, accessed 2008.
- [66] P.M. Glover, M.M. Castano-Briones, A. Bassett, and Z. Pikramenou. Ligand design for luminescent lanthanide complexes: From DNA recognition to sensing. *Chem. Listy* 98 (2004) s1–s120.
- [67] R.Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley & Sons, Toronto, 1981.
- [68] *The Oxford English Dictionary*. Oxford University Press, London, 1933.
- [69] R. Audi (ed.). *The Cambridge Dictionary of Philosophy*, 2nd ed. Cambridge University Press, UK, 1999.
- [70] K.Z. Lorenz. *The Foundations of Ethology*. Springer, Wien, 1981.
- [71] J.F. Peters, C. Henry, and S. Ramanna. Reinforcement learning in swarms that learn. In: *Proceedings of 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2005)*, Compiègne University of Technology, France, September 19–22, 2005, pp. 400–406.
- [72] J.W. Gibbs. *Elementary Principles in Statistical Mechanics*. Dover, NY, 1960.
- [73] D. Precup. *Temporal Abstraction in Reinforcement Learning*. Ph.D. dissertation. University of Massachusetts Amherst, May 2000.

31

Fuzzy Linear Programming

Jaroslav Ramík

31.1 Introduction

In mathematical programming problems preferences between alternatives are described by means of objective functions on a given set of alternatives. The values of the objective function describe effects from the alternatives; the more preferable alternatives have higher values than the less preferable ones. For example, in economic problems these values may reflect profits obtained in various means of production. The set of feasible alternatives in mathematical optimization problems is described by means of constraints – equations or inequalities – representing relevant relationships between alternatives. The results of the analysis depend largely on how adequately various factors of the real system are reflected in the description of the objective function(s) and the constraints. Mathematical formulation of the objective function and of the constraints in mathematical optimization problems usually includes some parameters; e.g., in problems of resource allocation the parameters may represent economic values such as costs of various types of production, shipment costs, etc. The values of such parameters depend on multiple factors usually not included in the formulation of the problem. Trying to make the model more representative, we often include the corresponding complex relations, causing the model to become more cumbersome and analytically unsolvable. Some attempts to increase ‘precision’ of the model will be of no practical value due to the impossibility of measuring the parameters accurately. On the other hand, the model with fixed values of its parameters may be too crude, since these values are often chosen in an arbitrary way.

An alternative approach is based on introducing into the model a more adequate representation of expert understanding of the nature of the parameters in an adequate form. In some approaches it has been done in the form of intervals, or convex polyhedral sets have been considered. Here, the parameters can be expressed in a more general form of fuzzy subsets of their possible values, in the form of information granules being treated as conceptual entities. As such, they result through some abstraction and afterward are used as fundamental building blocks for modeling purposes. In this way we obtain a new type of mathematical optimization problem containing fuzzy coefficients and fuzzy relations.

Considering linear optimization problems such treatment forms the essence of fuzzy linear programming (FLP) investigated in this chapter. As we show, for a special form of fuzzy parameters, the usual real numbers, our new formulation of the fuzzy linear optimization problem coincides with the corresponding classical formulations.

FLP problems and related ones have been extensively analyzed in many works published in papers and books displaying a variety of formulations and approaches. Most approaches to FLP problems are based on the straightforward use of the intersection of fuzzy sets representing goals and constraints. The resulting membership function is then maximized. This approach has been mentioned originally by

Bellman and Zadeh [1]. Later on many papers were devoted to the problem of linear programming with fuzzy coefficients, known under different names, mostly as fuzzy linear programming, but sometimes as possibilistic linear programming, flexible linear programming, vague linear programming, inexact linear programming, etc. For an extensive bibliography, see the overview in [2].

Here we present an approach based on a systematic extension of the traditional formulation of the LP problem. This approach is based on previous works of the author of this chapter, see [3–14], or, recent works [15, 16], and also on the works of many other authors, e.g., [17, 34].

In this overview chapter, among others we demonstrate that FLP essentially differs from stochastic programming; FLP has its own structure and tools for investigating broad classes of optimization problems. FLP is also different from parametric linear programming. Problems of parametric linear programming are in essence deterministic optimization problems with special variables called the parameters. The main interest in parametric linear programming is focused on finding functional relationships between the values of parameters and optimal solutions of linear programming problem.

An appropriate treatment of FLP problems requires proper application of special tools in a logically consistent manner. An important role in this treatment is played by generalized concave membership functions and fuzzy relations. The following treatment is based on the substance partly investigated in [39].

The schedule of this chapter is the following. First we formulate an optimization problem and, particularly, FLP problem associated with a collection of instances of the classical linear programming problem. After that we define a feasible solution of FLP problem and deal with the problem of ‘optimal solution’ of FLP problems. Two approaches are introduced: the first one – satisficing solution – is based on external goals modeled by fuzzy quantities and the second approach is based on the concept of efficient (non-dominated) solution. Second, our interest is focused on the problem of duality in FLP problems. The proofs of the propositions and theorems are not supplied here; the reader can find them mostly in [35], or [15]. The chapter is closed with a numerical example.

31.2 Fuzzy Sets, Fuzzy Relations

In this section we summarize basic notions and results from fuzzy set theory that will be useful in this chapter. Throughout this section, X is a non-empty set.

Definition 1. A fuzzy subset A of X is given by the membership function of A , $\mu_A : X \rightarrow [0, 1]$. The value $\mu_A(x)$ is called membership degree of x in the fuzzy set A . A fuzzy subset A of X is called a fuzzy set. The class of all fuzzy subsets of X is denoted by $\mathcal{F}(X)$.

In Definition 1 crisp fuzzy subsets of X and ‘classic’ subsets of X are in one-to-one correspondence. In this way, ‘classic’ subsets of X are isomorphically embedded into fuzzy subsets of X .

Definition 2. Let A be a fuzzy subset of X . The core of A , $\text{Core}(A)$, is defined by

$$\text{Core}(A) = \{x \in X \mid \mu_A(x) = 1\}. \quad (1)$$

The complement of A , $\mathcal{C}A$, is defined by

$$\mu_{\mathcal{C}A}(x) = 1 - \mu_A(x). \quad (2)$$

If the core of A is non-empty, then A is said to be normalized. The support of A , $\text{Supp}(A)$, is defined by

$$\text{Supp}(A) = \text{Cl}(\{x \in X \mid \mu_A(x) > 0\}). \quad (3)$$

Here, by Cl we denote the topological closure.

The height of A , $\text{Hgt}(A)$, is defined by

$$\text{Hgt}(A) = \sup\{\mu_A(x) \mid x \in X\}. \quad (4)$$

Note that if A is normalized, then $\text{Hgt}(A) = 1$, but not vice versa. The upper-level set of the membership function μ_A of A at $\alpha \in [0, 1]$ is denoted by $[A]_\alpha$ and called the α -cut of A ; that is,

$$[A]_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\}. \quad (5)$$

The strict upper-level set of the membership function μ_A of A at $\alpha \in [0, 1)$ is denoted by $(A)_\alpha$ and called the strict α -cut of A ; that is,

$$(A)_\alpha = \{x \in X \mid \mu_A(x) > \alpha\}. \quad (6)$$

Fuzzy sets can be equivalently characterized by their families of α -cuts (see e.g. [36]). α -cuts or strict α -cuts can be viewed as information granules being understood as conceptual entities.

Definition 3. Let $X \subseteq \mathbb{R}^m$ – the m -dimensional Euclidean space. A fuzzy subset A of X is called closed, bounded, compact, or convex if $[A]_\alpha$ is a closed, bounded, compact, or convex subset of X for every $\alpha \in (0, 1]$, respectively.

Now, we shall investigate fuzzy subsets of the real line; i.e., $X = \mathbb{R}$ and $\mathcal{F}(X) = \mathcal{F}(\mathbb{R})$.

Definition 4.

- (i) A fuzzy set A is called a fuzzy interval if for all $\alpha \in [0, 1] : [A]_\alpha$ is non-empty and convex subset of \mathbb{R} . The set of all fuzzy intervals is denoted by $\mathcal{F}_I(\mathbb{R})$.
- (ii) A fuzzy interval A is called a fuzzy number if its core is a singleton. The set of all fuzzy numbers will be denoted by $\mathcal{F}_N(\mathbb{R})$.

Notice that the membership function $\mu_A : \mathbb{R} \rightarrow [0, 1]$ of a fuzzy interval A is quasiconcave on \mathbb{R} . The following definitions will be useful.

Definition 5. Let $X \subseteq \mathbb{R}$. A function $f : \mathbb{R} \rightarrow [0, 1]$ is called

- (i) quasiconcave on X if

$$f(\lambda x + (1 - \lambda)y) \geq \min\{f(x), f(y)\}, \quad (7)$$

for every $x, y \in X$ and every $\lambda \in (0, 1)$ with $\lambda x + (1 - \lambda)y \in X$;

- (ii) strictly quasiconcave on X if

$$f(\lambda x + (1 - \lambda)y) > \min\{f(x), f(y)\}, \quad (8)$$

for every $x, y \in X$, $x \neq y$ and every $\lambda \in (0, 1)$ with $\lambda x + (1 - \lambda)y \in X$;

- (iii) semistrictly quasiconcave on X if f is quasiconcave on X and (8) holds for every $x, y \in X$ and every $\lambda \in (0, 1)$ with $\lambda x + (1 - \lambda)y \in X$, $f(\lambda x + (1 - \lambda)y) > 0$ and $f(x) \neq f(y)$.

Notice that membership functions of crisp subsets of \mathbb{R} are quasiconcave, but not strictly quasiconcave; they are, however, semistrictly quasiconcave on \mathbb{R} .

Definition 6. A fuzzy subset A of \mathbb{R} is called the fuzzy quantity if A is normal and compact with semistrictly quasiconcave membership function μ_A . The set of all fuzzy quantities is denoted by $\mathcal{F}_0(\mathbb{R})$.

By the definition $\mathcal{F}_0(\mathbb{R}) \subseteq \mathcal{F}_I(\mathbb{R})$, moreover, $\mathcal{F}_0(\mathbb{R})$ contains real numbers, intervals, triangular fuzzy numbers, bell-shaped fuzzy numbers, etc.

Now, let X and Y be non-empty sets. In the set theory, a binary relation R between the elements of the sets X and Y is defined as a subset of the Cartesian product $X \times Y$; that is, $R \subseteq X \times Y$.

A *valued relation* R on $X \times Y$ is a fuzzy subset of $X \times Y$. A *valued relation* R on X is a valued relation on $X \times X$.

Any binary relation R , $R \subseteq X \times Y$, is isomorphically embedded into the class of valued relations by its characteristic function χ_R , which is its membership function. In this sense, any binary relation is valued.

Let R be a valued relation on $X \times Y$. In FLP problems, we shall consider fuzzy relations assigning to every pair of fuzzy subsets a real number from interval $[0, 1]$. In other words, we consider valued relations \tilde{R} on $\mathcal{F}(X) \times \mathcal{F}(Y)$ such that $\mu_{\tilde{R}} : \mathcal{F}(X) \times \mathcal{F}(Y) \rightarrow [0, 1]$.

Convention: The elements $x \in X$ and $y \in Y$ are considered as fuzzy subsets of X and Y with the characteristic functions χ_x and χ_y as the membership functions. In this way we obtain the isomorphic embedding of X into $\mathcal{F}(X)$ and Y into $\mathcal{F}(Y)$, and in this sense we write $X \subseteq \mathcal{F}(X)$ and $Y \subseteq \mathcal{F}(Y)$, respectively.

Evidently, the usual binary relations $=$, $<$, and \leq can be understood as the valued relations.

Now, we define fuzzy relations which will be used for comparing the left and right sides of the constraints in optimization problems.

Definition 7. A fuzzy subset of $\mathcal{F}(X) \times \mathcal{F}(Y)$ is called a *fuzzy relation* on $X \times Y$. The set of all fuzzy relations on $\mathcal{F}(X) \times \mathcal{F}(Y)$ is denoted by $\mathcal{F}(\mathcal{F}(X) \times \mathcal{F}(Y))$. A fuzzy relation on $X \times X$ is called a *fuzzy relation* on X .

Definition 8. Let R be a valued relation on $X \times Y$. A fuzzy relation \tilde{R} on $X \times Y$ given by the membership function $\mu_{\tilde{R}} : \mathcal{F}(X) \times \mathcal{F}(Y) \rightarrow [0, 1]$ is called a *fuzzy extension* of relation R , if for each $x \in X$, $y \in Y$, it holds

$$\mu_{\tilde{R}}(x, y) = \mu_R(x, y). \quad (9)$$

On the left side of (9), x and y are understood as fuzzy subsets of X and Y defined by the membership functions identical with the characteristic functions of singletons $\{x\}$ and $\{y\}$, respectively.

Definition 9. Let $\Psi : \mathcal{F}(X \times Y) \rightarrow \mathcal{F}(\mathcal{F}(X) \times \mathcal{F}(Y))$ be a mapping. Let for all $R \in \mathcal{F}(X \times Y)$, $\Psi(R)$ be a fuzzy extension of relation R . Then the mapping Ψ is called a *fuzzy extension* of valued relations.

Definition 10. Let $\Phi, \Psi : \mathcal{F}(X \times Y) \rightarrow \mathcal{F}(\mathcal{F}(X) \times \mathcal{F}(Y))$ be mappings. We say that the mapping Φ is dual to Ψ if

$$\Phi(\mathcal{C}R) = \mathcal{C}\Psi(R) \quad (10)$$

holds for all $R \in \mathcal{F}(X \times Y)$. For Φ dual to Ψ , $R \in \mathcal{F}(X \times Y)$ a valued relation, the fuzzy relation $\Phi(R)$ is called *dual to fuzzy relation* $\Psi(R)$.

Proposition 11. A mapping Φ is dual to Ψ if and only if the mapping Ψ is dual to Φ .

The analogical statement holds for the dual fuzzy relations $\Phi(R)$ and $\Psi(R)$.

Now, we are going to define special mappings – important fuzzy extensions of valued relations. Recall the concept of t -norm, and t -conorm.

A class of functions $T : [0, 1]^2 \rightarrow [0, 1]$ that are commutative, associative, non-decreasing in every variable and satisfy the following boundary condition:

$$T(a, 1) = a \text{ for all } a \in [0, 1]$$

are called the *triangular norms* or *t -norms*. The most popular three examples of t -norms are $T_M(a, b) = \min\{a, b\}$, $T_P(a, b) = a \cdot b$, and $T_L(a, b) = \max\{0, a + b - 1\}$. They are called *minimum t -norm* T_M , *product t -norm* T_P , *Lukasiewicz t -norm* T_L , respectively.

A class of functions closely related to the class of t -norms is the class of functions $S : [0, 1]^2 \rightarrow [0, 1]$ that are commutative, associative, non-decreasing in every variable and satisfy the following boundary condition

$$S(a, 0) = a \text{ for all } a \in [0, 1].$$

The functions that satisfy all these properties are called the *triangular conorms* or *t -conorms* (see, e.g., [37]). For example, $S_M(a, b) = \max\{a, b\}$, $S_P(a, b) = a + b - a \cdot b$, $S_L(a, b) = \min\{1, a + b\}$, are the t -conorms. S_M , S_P , S_L are called the *maximum*, *probabilistic sum*, *bounded sum*, respectively. It can easily be verified that for each t -norm T , the function $T^* : [0, 1]^2 \rightarrow [0, 1]$ defined for all $a, b \in [0, 1]$ by

$$T^*(a, b) = 1 - T(1 - a, 1 - b)$$

is a t -conorm. The converse statement is also true. Namely, if S is a t -conorm, then the function $S^* : [0, 1]^2 \rightarrow [0, 1]$ defined for all $a, b \in [0, 1]$ by

$$S^*(a, b) = 1 - S(1 - a, 1 - b)$$

is a t -norm. The t -conorm T^* and t -norm S^* are called *dual* to the t -norm T and t -conorm S , respectively. It may easily be verified that

$$T_M^* = S_M, \quad T_P^* = S_P, \quad T_L^* = S_L.$$

A triangular norm T is said to be *strict* if it is continuous and strictly monotone. It is said to be *Archimedean* if for all $x, y \in (0, 1)$ there exists a positive integer n such that $T^{n-1}(x, \dots, x) < y$. Here, by commutativity and associativity we can define the extension to more than two arguments by the formula

$$T^{n-1}(x_1, x_2, \dots, x_n) = T(T^{n-2}(x_1, x_2, \dots, x_{n-1}), x_n),$$

where $T^1(x_1, x_2) = T(x_1, x_2)$.

Notice that if T is strict, then T is Archimedean.

Definition 12. An additive generator of a t -norm T is a strictly decreasing function $f : [0, 1] \rightarrow [0, +\infty]$ which is right continuous at 0, satisfies $f(1) = 0$, and is such that for all $x, y \in [0, 1]$ we have

$$f(x) + f(y) \in \text{Ran}(f) \cup [f(0), +\infty],$$

$$T(x, y) = f^{(-1)}(f(x) + f(y)),$$

where $\text{Ran}(f) = \{y \in \mathbb{R} \mid y = f(x), x \in [0, 1]\}$.

Triangular norms (t -conorms) constructed by means of additive (multiplicative) generators are always Archimedean. This property and some other properties of t -norms are summarized in [37].

Definition 13. Let T be a t -norm and S be a t -conorm. Let R be a valued relation on X . Fuzzy extensions $\Phi^T(R)$ and $\Phi^S(R)$ of a valued relation R on X defined for all fuzzy sets A, B with the membership functions $\mu_A : X \rightarrow [0, 1]$, $\mu_B : Y \rightarrow [0, 1]$, respectively, by

$$\mu_{\Phi^T(R)}(A, B) = \sup\{T(\mu_R(x, y), T(\mu_A(x), \mu_B(y))) \mid x, y \in X\}, \quad (11)$$

$$\mu_{\Phi^S(R)}(A, B) = \inf\{S(S(1 - \mu_A(x), 1 - \mu_B(y)), \mu_R(x, y)) \mid x, y \in X\}. \quad (12)$$

are called a T -fuzzy extension of relation R and S -fuzzy extension of relation R , respectively.

It can easily be verified that the T -fuzzy extension of relation R and S -fuzzy extension of relation R are fuzzy extensions of relation R given by Definition 8.

In the following proposition we prove a duality result between fuzzy extensions of valued relations. In a special case, particularly, $T = \min$ and $S = \max$, the analogical results can be also found in [38].

Proposition 14. *Let T be a t-norm and S be a t-conorm dual to T . Then Φ^T is dual to Φ^S .*

Definition 15. *Let R be \leq , i.e., R be a classical binary relation ‘less or equal’ on \mathbb{R} , and let $T = \min$, $S = \max$. We denote $\Phi^T(R)$ and $\Phi^S(R)$ from (11) and (12) by $\tilde{\leq}^{\min}$ and $\tilde{\leq}^{\max}$, respectively. From (11) and (12) we obtain two fuzzy extensions of relation \leq by*

$$\mu_{\tilde{\leq}^{\min}}(A, B) = \sup\{\min(\mu_A(x), \mu_B(y), \mu_R(x, y)) \mid x, y \in \mathbb{R}\}, \quad (13)$$

$$\mu_{\tilde{\leq}^{\max}}(A, B) = \inf\{\max(1 - \mu_A(x), 1 - \mu_B(y), \mu_R(x, y)) \mid x, y \in \mathbb{R}\}. \quad (14)$$

We equivalently write $A \tilde{\leq}^{\min} B$ and $A \tilde{\leq}^{\max} B$, instead of $\mu_{\tilde{\leq}^{\min}}(A, B)$ and $\mu_{\tilde{\leq}^{\max}}(A, B)$, respectively. By $A \tilde{\leq}^{\min} B$, we mean $B \tilde{\geq}^{\min} A$.

The following results are crucial for studying FLP problems.

Theorem 16. *Let R be \leq and let $T = \min$, $S = \max$. Let $A, B \in \mathcal{F}(\mathbb{R})$ be normal and compact fuzzy sets, $\alpha \in (0, 1)$.*

Then

- (i) $\mu_{\tilde{\leq}^{\min}}(A, B) \geq \alpha$ if and only if $\inf[A]_\alpha \leq \sup[B]_\alpha$,
- (ii) $\mu_{\tilde{\leq}^{\max}}(A, B) \geq \alpha$ if and only if $\sup(A)_{1-\alpha} \leq \inf(B)_{1-\alpha}$.

Let T be a t-norm and S be a t-conorm.

Definition 17.

1. A mapping $\Psi^{T,S} : \mathcal{F}(X \times Y) \rightarrow \mathcal{F}(\mathcal{F}(X) \times \mathcal{F}(Y))$ is defined for every valued relation $R \in \mathcal{F}(X \times Y)$ and for all fuzzy sets $A \in \mathcal{F}(X)$, $B \in \mathcal{F}(Y)$ by

$$\mu_{\Psi^{T,S}(R)}(A, B) = \sup\{\inf\{T(\mu_A(x), S(\mu_{CB}(y), \mu_R(x, y))) \mid y \in Y\} \mid x \in X\}. \quad (15)$$

2. A mapping $\Psi_{T,S} : \mathcal{F}(X \times Y) \rightarrow \mathcal{F}(\mathcal{F}(X) \times \mathcal{F}(Y))$ is defined for every valued relation $R \in \mathcal{F}(X \times Y)$ and for all fuzzy sets $A \in \mathcal{F}(X)$, $B \in \mathcal{F}(Y)$ by

$$\mu_{\Psi_{T,S}(R)}(A, B) = \inf\{\sup\{S(T(\mu_A(x), \mu_R(x, y)), \mu_{CB}(y)) \mid x \in X\} \mid y \in Y\}. \quad (16)$$

3. A mapping $\Psi^{S,T} : \mathcal{F}(X \times Y) \rightarrow \mathcal{F}(\mathcal{F}(X) \times \mathcal{F}(Y))$ is defined for every valued relation $R \in \mathcal{F}(X \times Y)$ and for all fuzzy sets $A \in \mathcal{F}(X)$, $B \in \mathcal{F}(Y)$ by

$$\mu_{\Psi^{S,T}(R)}(A, B) = \sup\{\inf\{T(S(\mu_{CA}(x), \mu_R(x, y)), \mu_B(y)) \mid x \in X\} \mid y \in Y\} \quad (17)$$

4. A mapping $\Psi_{S,T} : \mathcal{F}(X \times Y) \rightarrow \mathcal{F}(\mathcal{F}(X) \times \mathcal{F}(Y))$ is defined for every valued relation $R \in \mathcal{F}(X \times Y)$ and for all fuzzy sets $A \in \mathcal{F}(X)$, $B \in \mathcal{F}(Y)$ by

$$\mu_{\Psi_{S,T}(R)}(A, B) = \inf\{\sup\{S(\mu_{CA}(x), T(\mu_B(y), \mu_R(x, y))) \mid y \in Y\} \mid x \in X\}. \quad (18)$$

The previous four fuzzy relations are also fuzzy extensions of valued relations by Definition 9.

31.3 Fuzzy Linear Programming Problems

Now, we turn to optimization theory and consider the following *optimization problem*:

$$\begin{aligned} & \text{maximize (minimize)} \quad f(x) \\ & \text{subject to} \quad x \in X, \end{aligned} \tag{19}$$

where f is a real-valued function on \mathbb{R}^n called the *objective function* and X is a non-empty subset of \mathbb{R}^n given by means of real-valued functions g_1, g_2, \dots, g_m on \mathbb{R}^n , the set of all solutions of the system

$$\begin{aligned} g_i(x) &= b_i, \quad i = 1, 2, \dots, m_1, \\ g_i(x) &\leq b_i, \quad i = m_1 + 1, m_1 + 2, \dots, m, \\ x_j &\geq 0, \quad j = 1, 2, \dots, n, \end{aligned}$$

called the *constraints*. The elements of X are called *feasible solutions* of (19), and the feasible solution x^* where f attains its global maximum (or minimum) over X is called the *optimal solution*.

Most frequent optimization problems are linear ones. In this chapter we are concerned with FLP problem related to linear programming problems in the following form.

Let $\mathcal{M} = \{1, 2, \dots, m\}$ and $\mathcal{N} = \{1, 2, \dots, n\}$, where m and n are positive integers. Then for each $c = (c_1, c_2, \dots, c_n)^T \in \mathbb{R}^n$ and $a_i = (a_{i1}, a_{i2}, \dots, a_{in})^T \in \mathbb{R}^n$, $i \in \mathcal{M}$, the functions $f(\cdot, c)$ and $g(\cdot, a_i)$ defined on \mathbb{R}^n by

$$f(x, c_1, \dots, c_n) = c_1x_1 + \dots + c_nx_n, \tag{20}$$

$$g_i(x, a_{i1}, \dots, a_{in}) = a_{i1}x_1 + \dots + a_{in}x_n, \quad i \in \mathcal{M}, \tag{21}$$

are linear on \mathbb{R}^n . For each $c \in \mathbb{R}^n$ and $a_i \in \mathbb{R}^n$, $i \in \mathcal{M}$, we consider the *linear programming problem* (classical linear programming)

$$\begin{aligned} & \text{maximize (minimize)} \quad c_1x_1 + \dots + c_nx_n \\ & \text{subject to} \quad a_{i1}x_1 + \dots + a_{in}x_n \leq b_i, \quad i \in \mathcal{M}, \\ & \quad x_j \geq 0, \quad j \in \mathcal{N}. \end{aligned} \tag{22}$$

The set of all feasible solutions of problem (22) is denoted by X ; that is,

$$X = \{x \in \mathbb{R}^n \mid a_{i1}x_1 + \dots + a_{in}x_n \leq b_i, \quad i \in \mathcal{M}, \quad x_j \geq 0, \quad j \in \mathcal{N}\}. \tag{23}$$

Assumptions and remarks.

1. Let f, g_i be linear functions defined by (20) and (21), respectively. From now on, the parameters c_j , a_{ij} , and b_i will be considered as *fuzzy quantities*, that is, normal and compact fuzzy subsets of the Euclidean space \mathbb{R} with semistrictly quasiconcave membership function (see Definition 6). This assumption makes it possible to include classical linear programming problems into fuzzy linear programming ones. The fuzzy quantities are denoted with the tilde above the corresponding symbol. We also have $\mu_{\tilde{c}_j} : \mathbb{R} \rightarrow [0, 1]$, $\mu_{\tilde{a}_{ij}} : \mathbb{R} \rightarrow [0, 1]$ and $\mu_{\tilde{b}_i} : \mathbb{R} \rightarrow [0, 1]$, $i \in \mathcal{M}$, $j \in \mathcal{N}$ – membership functions of the fuzzy parameters \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i , respectively. The fuzzy quantities being real numbers will not be denoted with the tilde.

2. Let \tilde{R}_i , $i \in \mathcal{M}$, be fuzzy relations on \mathbb{R} . They will be used for ‘comparing the left and right sides’ of the constraints. Primarily, we shall study the case of $\tilde{R}_i = \tilde{R}$, for all $i \in \mathcal{M}$; i.e., all fuzzy relations in the constraints are the same.

3. The ‘optimization,’ i.e., ‘maximization’ or ‘minimization’ of the objective function requires a special treatment, as the set of fuzzy values of the objective function is not linearly ordered. In order to ‘maximize’ the objective function we shall define a suitable concept of ‘optimal solution.’ It will be done by two distinct approaches: Applying the first approach an exogenously given fuzzy goal $\tilde{d} \in \mathcal{F}(\mathbb{R})$ and special

fuzzy relation \tilde{R}_0 on \mathbb{R} is introduced. In the second approach we define α -efficient (α -non-dominated) solution of the FLP problem. Some other approaches can be found in the literature (see [3, 16, 20]).

The FLP problem associated with linear programming problem (22) is defined as follows:

$$\begin{aligned} & \text{'maximize' ('minimize')} \quad \tilde{c}_1x_1 + \cdots + \tilde{c}_n x_n \\ & \text{subject to} \quad (\tilde{a}_{i1}x_1 + \cdots + \tilde{a}_{in}x_n)\tilde{R}_i \tilde{b}_i, \quad i \in \mathcal{M}, \\ & \quad x_j \geq 0, \quad j \in \mathcal{N}. \end{aligned} \quad (24)$$

Here, $\tilde{R}_i, i \in \mathcal{M}$, are fuzzy relations on \mathbb{R} . The objective function values and the left-hand side values of the constraints of (24) are obtained by the *extension principle*:

For given $\tilde{c}_1, \dots, \tilde{c}_n \in \mathcal{F}_0(\mathbb{R})$, $\tilde{f}(x, \tilde{c}_1, \dots, \tilde{c}_n)$ is the fuzzy extension of $f(x, c_1, \dots, c_n)$ with the membership function defined for each $t \in \mathbb{R}$ by

$$\mu_{\tilde{f}}(t) = \begin{cases} \sup \left\{ T(\mu_{\tilde{c}_1}(c_1), \dots, \mu_{\tilde{c}_n}(c_n)) \mid \begin{array}{l} c_1, \dots, c_n \in \mathbb{R}, \\ c_1x_1 + \cdots + c_nx_n = t \end{array} \right\} & \text{if } f^{-1}(x; t) \neq \emptyset, \\ 0 & \text{otherwise,} \end{cases}$$

where $f^{-1}(x, t) = \{(c_1, \dots, c_n)^T \in \mathbb{R}^n \mid f(x, c_1, \dots, c_n) = t\}$.

Particularly, for $f(x, c_1, \dots, c_n) = c_1x_1 + \cdots + c_nx_n$, the fuzzy set $\tilde{f}(x, \tilde{c}_1, \dots, \tilde{c}_n)$ will be denoted as $\tilde{c}_1x_1 + \cdots + \tilde{c}_nx_n$; i.e.,

$$\tilde{f}(x, \tilde{c}_1, \dots, \tilde{c}_n) = \tilde{c}_1x_1 + \cdots + \tilde{c}_nx_n. \quad (25)$$

Similarly, the membership function of $\tilde{g}_i(x, \tilde{a}_{i1}, \dots, \tilde{a}_{i1})$ is defined for each $t \in \mathbb{R}$ by

$$\mu_{\tilde{g}_i}(t) = \begin{cases} \sup \left\{ T(\mu_{\tilde{a}_{i1}}(a_1), \dots, \mu_{\tilde{a}_{in}}(a_n)) \mid \begin{array}{l} a_1, \dots, a_n \in \mathbb{R}, \\ a_1x_1 + \cdots + a_nx_n = t \end{array} \right\} & \text{if } g_i^{-1}(x; t) \neq \emptyset, \\ 0 & \text{otherwise,} \end{cases}$$

where

$$g_i^{-1}(x, t) = \{(a_1, \dots, a_n)^T \in \mathbb{R}^n \mid a_1x_1 + \cdots + a_nx_n = t\}.$$

Here, the fuzzy set $\tilde{g}_i(x, \tilde{a}_{i1}, \dots, \tilde{a}_{i1})$ is denoted as $\tilde{a}_{i1}x_1 + \cdots + \tilde{a}_{in}x_n$; i.e.,

$$\tilde{g}_i(x, \tilde{a}_{i1}, \dots, \tilde{a}_{i1}) = \tilde{a}_{i1}x_1 + \cdots + \tilde{a}_{in}x_n$$

for every $i \in \mathcal{M}$ and for each $x \in \mathbb{R}^n$. The following proposition can easily be derived from the definition.

Proposition 18. Let $\tilde{a}_j \in \mathcal{F}_0(\mathbb{R}), x_j \geq 0, j \in \mathcal{N}$. Then $\tilde{a}_{i1}x_1 + \cdots + \tilde{a}_{in}x_n$ defined by the extension principle is again a fuzzy quantity.

In (24) the value $\tilde{a}_{i1}x_1 + \cdots + \tilde{a}_{in}x_n \in \mathcal{F}_0(\mathbb{R})$ is ‘compared to’ the fuzzy quantity $\tilde{b}_i \in \mathcal{F}_0(\mathbb{R})$ by fuzzy relation $\tilde{R}_i, i \in \mathcal{M}$.

Usually, the fuzzy relations \tilde{R}_i on \mathbb{R} for comparing the left and right sides of the constraints of (24) are extensions of a valued relation on \mathbb{R} , particularly, the binary inequality relations ‘ \leq ’ or ‘ \geq ’.

If \tilde{R}_i is the T -fuzzy extension of relation $R_i, i \in \mathcal{M}$, then the membership function of the i th constraint is as follows:

$$\mu_{\tilde{R}_i}(\tilde{a}_{i1}x_1 + \cdots + \tilde{a}_{in}x_n, \tilde{b}_i) = \sup\{T(\mu_{\tilde{a}_{i1}x_1 + \cdots + \tilde{a}_{in}x_n}(u), \mu_{\tilde{b}_i}(v)) \mid u R_i v\}.$$

For aggregating fuzzy constraints in FLP problem (24), we need some operators with reasonable properties. Such operators should assign to each tuple of elements a unique real number. For this purpose, t -norms or t -conorms can be applied. However, we know some other useful operators generalizing usual t -norms or t -conorms. Clearly, between arbitrary interval $[a, b]$ in \mathbb{R} and the unit interval $[0, 1]$ there exists a one-to-one correspondence. Hence, each result for operators on the interval $[a, b]$ can be transformed into a result for operators on $[0, 1]$ and vice versa. Moreover, the aggregation operators on $[0, 1]$ should be sufficiently general, at least from theoretical point of view. In many cases, general aggregation operators can be derived from n -ary operations on $[0, 1]$.

Definition 19. An aggregation operator G is a sequence $\{G_n\}_{n=1}^{\infty}$ of mappings (called aggregating mappings) $G_n : [0, 1]^n \rightarrow [0, 1]$, satisfying the following properties:

- (i) $G_1(x) = x$ for each $x \in [0, 1]$;
- (ii) $G_n(x_1, x_2, \dots, x_n) \leq G_n(y_1, y_2, \dots, y_n)$, whenever $x_i \leq y_i$ for each $i = 1, 2, \dots, n$, and every $n = 2, 3, \dots$;
- (iii) $G_n(0, 0, \dots, 0) = 0$ and $G_n(1, 1, \dots, 1) = 1$ for every $n = 2, 3, \dots$

Condition (i) says that G_1 is a unary identity operation, (ii) means that aggregating mapping G_n is monotone, particularly non-decreasing in all of its arguments x_i , and condition (iii) represents the boundary conditions.

Examples of aggregation operators (see e.g. [33, 35]):

- (1) t -norms and t -conorms;
- (2) usual averages: the arithmetic mean, geometric mean, harmonic mean, and root-power mean;
- (3) k -order statistic aggregation operators;
- (4) order weighted averaging (OWA) operators;
- (5) Sugeno and Choquet integrals.

31.4 Feasible Solution

Let us begin with the concept of feasible solution of an FLP problem (24).

Definition 20. Let $g_i, i \in \mathcal{M}$, be linear functions defined by (21). Let $\mu_{\tilde{a}_{ij}} : \mathbb{R} \rightarrow [0, 1]$ and $\mu_{\tilde{b}_i} : \mathbb{R} \rightarrow [0, 1]$, $i \in \mathcal{M}, j \in \mathcal{N}$, be membership functions of fuzzy quantities \tilde{a}_{ij} and \tilde{b}_i , respectively. Let $\tilde{R}_i, i \in \mathcal{M}$, be fuzzy relations on \mathbb{R} . Let G_A be an aggregation operator and T be a t -norm.

A fuzzy set \tilde{X} , the membership function $\mu_{\tilde{X}}$ of which is defined for all $x \in \mathbb{R}^n$ by

$$\mu_{\tilde{X}}(x) = \begin{cases} G_A(\mu_{\tilde{R}_1}(\tilde{a}_{11}x_1 + \dots + \tilde{a}_{1n}x_n, \tilde{b}_1), \dots, \\ \mu_{\tilde{R}_m}(\tilde{a}_{m1}x_1 + \dots + \tilde{a}_{mn}x_n, \tilde{b}_m)) & \text{if } x_j \geq 0 \text{ for all } j \in \mathcal{N}, \\ 0 & \text{otherwise,} \end{cases} \quad (26)$$

is called the feasible solution of the FLP problem (24).

For $\alpha \in (0, 1)$, a vector $x \in [\tilde{X}]_{\alpha}$ is called the α -feasible solution of the FLP problem (24).

A vector $\bar{x} \in \mathbb{R}^n$ such that $\mu_{\tilde{X}}(\bar{x}) = \text{Hgt}(\tilde{X})$ is called the max-feasible solution.

By the definition the feasible solution \tilde{X} of an FLP problem is a fuzzy set. On the other hand, the α -feasible solution is a vector belonging to the α -cut of the feasible solution \tilde{X} and the same is true for the max-feasible solution – a special α -feasible solution with $\alpha = \text{Hgt}(\tilde{X})$.

Given a feasible solution \tilde{X} and $\alpha \in (0, 1]$ (the degree of possibility, feasibility, satisfaction etc.), any vector $x \in \mathbb{R}^n$ satisfying $\mu_{\tilde{X}}(x) \geq \alpha$ is the α -feasible solution of the corresponding FLP problem.

For $i \in \mathcal{M}$, \tilde{X}_i denotes the fuzzy subset of \mathbb{R}^n with the membership function $\mu_{\tilde{X}_i}$ defined for all $x \in \mathbb{R}^n$ as

$$\mu_{\tilde{X}_i}(x) = \mu_{\tilde{R}_i}(\tilde{a}_{i1}x_1 \tilde{+} \cdots \tilde{+} \tilde{a}_{in}x_n, \tilde{b}_i). \quad (27)$$

Fuzzy set (27) is interpreted as i th fuzzy constraint. All fuzzy constraints \tilde{X}_i are aggregated into the feasible solution (26) by the aggregation operator G_A . Usually, $G_A = \min$ is used for aggregating the constraints; similarly, the t -norm $T = \min$ is used for extending arithmetic operations ‘ $\tilde{+}$ ’.

Clearly, if a_{ij} and b_i are real parameters, then the feasible solution is also real. Moreover, if for all $i \in \mathcal{M}$, \tilde{R}_i are T -fuzzy extensions of valued relations R_i and for two collections of fuzzy parameters it holds $\tilde{a}'_{ij} \subseteq \tilde{a}''_{ij}$ and $\tilde{b}'_i \subseteq \tilde{b}''_i$, then the same holds for the feasible solutions, i.e., $\tilde{X}' \subseteq \tilde{X}''$ (see also Proposition 25 below).

Now, we derive special formulas which allow for computing an α -feasible solution $x \in [\tilde{X}]_\alpha$ of the FLP problem (24). For this purpose, the following notation is useful. Given $\alpha \in (0, 1]$, $i \in \mathcal{M}$, $j \in \mathcal{N}$, let $\tilde{a} \in \mathcal{F}_0(\mathbb{R})$. We denote

$$\tilde{a}^L(\alpha) = \inf \{t \in \mathbb{R} | t \in [\tilde{a}]_\alpha\} = \inf[\tilde{a}]_\alpha, \quad \tilde{a}^R(\alpha) = \sup \{t | t \in [\tilde{a}]_\alpha\} = \sup[\tilde{a}]_\alpha. \quad (28)$$

Theorem 21. Let \tilde{a}_{ij} and \tilde{b}_i be fuzzy quantities and $x_j \geq 0$ for all $i \in \mathcal{M}$, $j \in \mathcal{N}$, $\alpha \in (0, 1)$. Let $\tilde{\leq}^{\min}$ and $\tilde{\leq}^{\max}$ be fuzzy extensions of the binary relation \leq . Then for $i \in \mathcal{M}$ it holds

(i) $\mu_{\tilde{\leq}^{\min}}(\tilde{a}_{i1}x_1 \tilde{+} \cdots \tilde{+} \tilde{a}_{in}x_n, \tilde{b}_i) \geq \alpha$ if and only if

$$\sum_{j \in \mathcal{N}} \tilde{a}_{ij}^L(\alpha)x_j \leq \tilde{b}_i^R(\alpha),$$

(ii) $\mu_{\tilde{\leq}^{\max}}(\tilde{a}_{i1}x_1 \tilde{+} \cdots \tilde{+} \tilde{a}_{in}x_n, \tilde{b}_i) \geq \alpha$ if and only if

$$\sum_{j \in \mathcal{N}} \tilde{a}_{ij}^R(1 - \alpha)x_j \leq \tilde{b}_i^L(1 - \alpha).$$

Notice that semistrict quasiconcavity of fuzzy quantities is a property securing validity of the equivalence (ii) in Theorem 21, which plays a key role in deriving duality principle in FLP we shall deal later on.

In the following example we apply Theorem 21 to a broad and practical class of so-called $(\mathcal{L}, \mathcal{R})$ -fuzzy quantities with membership functions given by shifts and contractions of special generator functions.

Example 22. Let $l, r \in \mathbb{R}$ with $l \leq r$, let $\gamma, \delta \in [0, +\infty)$ and let \mathcal{L}, \mathcal{R} be non-increasing, upper-semicontinuous, semistrictly quasiconcave functions mapping interval $[0, +\infty)$ into $[0, 1]$, i.e. $\mathcal{L}, \mathcal{R} : [0, +\infty) \rightarrow [0, 1]$. Moreover, assume that $\mathcal{L}(0) = \mathcal{R}(0) = 1$ and $\lim_{x \rightarrow +\infty} \mathcal{L}(x) = \lim_{x \rightarrow +\infty} \mathcal{R}(x) = 0$, for each $x \in \mathbb{R}$

$$\mu_A(x) = \begin{cases} \mathcal{L}\left(\frac{l-x}{\gamma}\right) & \text{if } x \in (l - \gamma, l), \gamma > 0, \\ 1 & \text{if } x \in [l, r], \\ \mathcal{R}\left(\frac{x-r}{\delta}\right) & \text{if } x \in (r, r + \delta), \delta > 0, \\ 0 & \text{otherwise.} \end{cases}$$

We shall write $A = (l, r, \gamma, \delta)_{\mathcal{LR}}$, the fuzzy quantity A is called an $(\mathcal{L}, \mathcal{R})$ -fuzzy interval, and the set of all $(\mathcal{L}, \mathcal{R})$ -fuzzy intervals will be denoted by $\mathcal{F}_{\mathcal{LR}}(\mathbb{R})$. Observe that $\text{Core}(A) = [l, r]$ and $[A]_\alpha$ is a compact interval for every $\alpha \in (0, 1]$. It is obvious that the class of $(\mathcal{L}, \mathcal{R})$ -fuzzy intervals extends the class of closed intervals $[a, b] \subseteq \mathbb{R}$ including the case $a = b$, i.e., real numbers. Similarly, if the membership

functions of \tilde{a}_{ij} and \tilde{b}_i are given analytically by

$$\mu_{\tilde{a}_{ij}}(x) = \begin{cases} \mathcal{L}\left(\frac{l_{ij}-x}{\gamma_{ij}}\right) & \text{if } x \in [l_{ij} - \gamma_{ij}, l_{ij}), \gamma_{ij} > 0, \\ 1 & \text{if } x \in [l_{ij}, r_{ij}], \\ \mathcal{R}\left(\frac{x-r_{ij}}{\delta_{ij}}\right) & \text{if } x \in (r_{ij}, r_{ij} + \delta_{ij}], \delta_{ij} > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (29)$$

and

$$\mu_{\tilde{b}_i}(x) = \begin{cases} \mathcal{L}\left(\frac{l_i-x}{\gamma_i}\right) & \text{if } x \in [l_i - \gamma_i, l_i), \gamma_i > 0, \\ 1 & \text{if } x \in [l_i, r_i], \\ \mathcal{R}\left(\frac{x-r_i}{\delta_i}\right) & \text{if } x \in (r_i, r_i + \delta_i], \delta_i > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (30)$$

for each $x \in \mathbb{R}$, $i \in \mathcal{M}$, $j \in \mathcal{N}$. Then the values of (28) can be computed as

$$\begin{aligned} \tilde{a}_{ij}^L(\alpha) &= l_{ij} - \gamma_{ij} \mathcal{L}^{(-1)}(\alpha), & \tilde{a}_{ij}^R(\alpha) &= r_{ij} + \delta_{ij} \mathcal{R}^{(-1)}(\alpha), \\ \tilde{b}_i^L(\alpha) &= l_i - \gamma_i \mathcal{L}^{(-1)}(\alpha), & \tilde{b}_i^R(\alpha) &= r_i + \delta_i \mathcal{R}^{(-1)}(\alpha), \end{aligned}$$

where $\mathcal{L}^{(-1)}$ and $\mathcal{R}^{(-1)}$ are pseudo-inverse functions of \mathcal{L} and \mathcal{R} defined by $\mathcal{L}^{(-1)}(\alpha) = \sup\{x | \mathcal{L}(x) \geq \alpha\}$ and $\mathcal{R}^{(-1)}(\alpha) = \sup\{x | \mathcal{R}(x) \geq \alpha\}$, respectively.

Let $G_A = \min$. By Theorem 21, the α -cut $[\tilde{X}]_\alpha$ of the feasible solution of (24) with $\tilde{R}_i = \tilde{\leq}^{\min}$, $i \in \mathcal{M}$, can be obtained by solving the system of inequalities

$$\sum_{j \in \mathcal{N}} (l_{ij} - \gamma_{ij} \mathcal{L}^{(-1)}(\alpha)) x_j \leq r_i + \delta_i \mathcal{R}^{(-1)}(\alpha), \quad i \in \mathcal{M}. \quad (31)$$

On the other hand, the α -cut $[\tilde{X}]_\alpha$ of the feasible solution of (24) with $\tilde{R}_i = \tilde{\leq}^{\max}$, $i \in \mathcal{M}$, can be obtained by solving the system of inequalities

$$\sum_{j \in \mathcal{N}} (r_{ij} + \delta_{ij} \mathcal{R}^{(-1)}(\alpha)) x_j \leq l_i - \gamma_i \mathcal{L}^{(-1)}(\alpha), \quad i \in \mathcal{M}. \quad (32)$$

Moreover, by (31) and (32), $[\tilde{X}]_\alpha$ is the intersection of a finite number of half spaces, hence a convex polyhedral set.

31.5 ‘Optimal’ Solution

The ‘optimization,’ i.e., ‘maximization’ or ‘minimization,’ of the objective function requires a special approach, as the set of fuzzy values of the objective function is not linearly ordered. In order to ‘maximize’ the objective function we shall introduce a suitable concept of ‘optimal solution.’ It shall be done by two distinct approaches, namely, (1) satisficing solution and (2) α -efficient solution.

31.5.1 Satisficing Solution

We assume the existence of an exogenously given goal $\tilde{d} \in \mathcal{F}(\mathbb{R})$. The fuzzy value \tilde{d} is compared to fuzzy values $\tilde{c}_1 x_1 \tilde{+} \dots \tilde{+} \tilde{c}_n x_n$ of the objective function by a given fuzzy relation \tilde{R}_0 . In this way the fuzzy objective function is treated as another constraint

$$(\tilde{c}_1 x_1 \tilde{+} \dots \tilde{+} \tilde{c}_n x_n) \tilde{R}_0 \tilde{d}.$$

Satisficing solution is then obtained by a modification of definition of feasible solution.

Definition 23. Let f, g_i be linear functions defined by (20) and (21). Let $\mu_{\tilde{c}_j} : \mathbb{R} \rightarrow [0, 1]$, $\mu_{\tilde{a}_{ij}} : \mathbb{R} \rightarrow [0, 1]$ and let $\mu_{\tilde{b}_i} : \mathbb{R} \rightarrow [0, 1]$, $i \in \mathcal{M}$, $j \in \mathcal{N}$, be membership functions of fuzzy quantities \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i , respectively. Moreover, let $\tilde{d} \in \mathcal{F}_I(\mathbb{R})$ be a fuzzy interval, called the fuzzy goal. Let \tilde{R}_i , $i \in \{0\} \cup \mathcal{M}$, be fuzzy relations on \mathbb{R} and T be a t-norm, G and G_A be aggregation operators.

A fuzzy set \tilde{X}^* with the membership function $\mu_{\tilde{X}^*}$ defined for all $x \in \mathbb{R}^n$ by

$$\mu_{\tilde{X}^*}(x) = G_A(\mu_{\tilde{R}_0}(\tilde{c}_1 x_1 \tilde{+} \cdots \tilde{+} \tilde{c}_n x_n, \tilde{d}), \mu_{\tilde{X}}(x)),$$

where $\mu_{\tilde{X}}(x)$ is the membership function of the feasible solution, is called the satisficing solution of FLP problem (24).

For $\alpha \in (0, 1]$, a vector $x \in [\tilde{X}^*]_\alpha$ is called the α -satisficing solution of FLP problem (24).

A vector $x^* \in \mathbb{R}^n$ with the property

$$\mu_{\tilde{X}^*}(x^*) = \text{Hgt}(\tilde{X}^*) \quad (33)$$

is called the max-satisficing solution.

By Definition 23 any satisficing solution of the FLP problem is a fuzzy set. On the other hand, the α -satisficing solution belongs to the α -cut $[\tilde{X}^*]_\alpha$. Likewise, the max-satisficing solution is an α -satisficing solution with $\alpha = \text{Hgt}(\tilde{X}^*)$.

The t-norm T is used for extending arithmetic operations, the aggregation operator G for joining the individual constraints into the feasible solution and G_A is applied for aggregating the fuzzy set of the feasible solution and fuzzy set of the objective \tilde{X}_0 defined by the membership function

$$\mu_{\tilde{X}_0}(x) = \mu_{\tilde{R}_0}(\tilde{c}_1 x_1 \tilde{+} \cdots \tilde{+} \tilde{c}_n x_n, \tilde{d}),$$

for all $x \in \mathbb{R}^n$. The membership function of optimal solution \tilde{X}^* is defined for all $x \in \mathbb{R}^n$ by

$$\mu_{\tilde{X}^*}(x) = G_A(\mu_{\tilde{X}_0}(x), \mu_{\tilde{X}}(x)).$$

If (24) is a maximization problem ‘the higher value is better,’ then the membership function μ_d of the fuzzy goal \tilde{d} is supposed to be increasing or non-decreasing. If (24) is a minimization problem ‘the lower value is better,’ then the membership function μ_d of \tilde{d} is decreasing or non-increasing. The fuzzy relation \tilde{R}_0 for comparing $\tilde{c}_1 x_1 \tilde{+} \cdots \tilde{+} \tilde{c}_n x_n$ and \tilde{d} is supposed to be a fuzzy extension of \geq or \leq .

Formally, Definitions 20 and 23 are similar. In other words, the concepts of feasible solution is similar to the concept of optimal solution. Therefore, we can take advantage of the properties of feasible solution studied in the preceding section.

Observe that in case of real parameters c_j, a_{ij} , and b_i , the set of all max-optimal solutions given by (33) coincides with the set of all optimal solutions of the classical linear programming problem. We have the following result.

Proposition 24. Let $c_j, a_{ij}, b_i \in \mathbb{R}$ be real numbers or intervals for all $i \in \mathcal{M}$, $j \in \mathcal{N}$. Let $\tilde{d} \in \mathcal{F}(\mathbb{R})$ be a fuzzy goal with a strictly increasing membership function $\mu_{\tilde{d}}$. Let for $i \in \mathcal{M}$, \tilde{R}_i be a fuzzy extension of relation ‘ \leq ’ on \mathbb{R} and \tilde{R}_0 be a T -fuzzy extension of relation ‘ \geq ’. Let T , G , and G_A be t-norms. Then the set of all max-satisficing solutions of (24) coincides with the set of all optimal solutions X^* of linear programming problem (22).

Proposition 25. Let $\tilde{c}'_j, \tilde{a}'_{ij}$, and \tilde{b}'_i , and $\tilde{c}''_j, \tilde{a}''_{ij}$, and \tilde{b}''_i be two collections of fuzzy quantities – parameters of FLP problem (24), $i \in \mathcal{M}$, $j \in \mathcal{N}$. Let T , G , G_A be t-norms. Let \tilde{R}_i , $i \in \{0\} \cup \mathcal{M}$, be T -fuzzy extensions of valued relations R_i on \mathbb{R} , and $\tilde{d} \in \mathcal{F}_I(\mathbb{R})$ be a fuzzy goal.

If \tilde{X}^{**} is the satisficing solution of FLP problem (24) with the parameters $\tilde{c}'_j, \tilde{a}'_{ij}$, and \tilde{b}'_i , \tilde{X}^{***} is the satisficing solution of the FLP problem with the parameters $\tilde{c}''_j, \tilde{a}''_{ij}$, and \tilde{b}''_i such that for all $i \in \mathcal{M}$, $j \in \mathcal{N}$,

$$\tilde{c}'_j \subseteq \tilde{c}''_j, \tilde{a}'_{ij} \subseteq \tilde{a}''_{ij} \text{ and } \tilde{b}'_i \subseteq \tilde{b}''_i,$$

then it holds

$$\tilde{X}^{*'} \subseteq \tilde{X}^{**}.$$

Further on, we extend Theorem 21 to the case of satisficing solution of an FLP problem. For this purpose we introduce the following notation. Given $\alpha \in (0, 1]$, $j \in \mathcal{N}$, let

$$\begin{aligned}\tilde{c}_j^L(\alpha) &= \inf\{c \mid c \in [\tilde{c}_j]_\alpha\}, \\ \tilde{c}_j^R(\alpha) &= \sup\{c \mid c \in [\tilde{c}_j]_\alpha\}, \\ \tilde{d}^L(\alpha) &= \inf\{d \mid d \in [\tilde{d}]_\alpha\}, \\ \tilde{d}^R(\alpha) &= \sup\{d \mid d \in [\tilde{d}]_\alpha\}.\end{aligned}$$

Theorem 26. Let \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i be fuzzy quantities, $i \in \mathcal{M}$, $j \in \mathcal{N}$. Let $\tilde{d} \in \mathcal{F}(\mathbb{R})$ be a fuzzy goal with the membership function $\mu_{\tilde{d}}$ satisfying the following conditions

$$\begin{aligned}\mu_{\tilde{d}} &\text{ is upper semicontinuous,} \\ \mu_{\tilde{d}} &\text{ is strictly increasing,} \\ \lim_{t \rightarrow -\infty} \mu_{\tilde{d}}(t) &= 0.\end{aligned}\tag{34}$$

For $i \in \mathcal{M}$, let \tilde{R}_i be the T -fuzzy extension of the binary relation \leq on \mathbb{R} and \tilde{R}_0 be the T -fuzzy extension of the binary relation \geq on \mathbb{R} . Let $T = G = G_A = \min$. Let \tilde{X}^* be a satisficing solution of FLP problem (24) and let $\alpha \in (0, 1)$.

A vector $x = (x_1, \dots, x_n) \geq 0$ belongs to $[\tilde{X}^*]_\alpha$ if and only if

$$\begin{aligned}\sum_{j=1}^n \tilde{c}_j^R(\alpha)x_j &\geq \tilde{d}^L(\alpha), \\ \sum_{j=1}^n \tilde{a}_{ij}^L(\alpha)x_j &\leq \tilde{b}_i^R(\alpha), \quad i \in \mathcal{M}.\end{aligned}$$

If the membership functions of the fuzzy parameters \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i can be formulated in an explicit form, e.g., as $(\mathcal{L}, \mathcal{R})$ -fuzzy quantities, see (30), then we can find a max-satisficing solution as the optimal solution of some associated classical optimization problem.

Proposition 27. Let

$$\mu_{\tilde{X}_0}(x) = \mu_{\tilde{R}_0}(\tilde{c}_1x_1 + \dots + \tilde{c}_nx_n, \tilde{d})$$

be the membership function of the fuzzy objective and let

$$\mu_{\tilde{X}_i}(x) = \mu_{\tilde{R}_i}(\tilde{a}_{i1}x_1 + \dots + \tilde{a}_{in}x_n, \tilde{b}_i), \quad i \in \mathcal{M},$$

be the membership functions of the fuzzy constraints, $x = (x_1, \dots, x_n) \in \mathbb{R}^n$. Let $T = G = G_A = \min$ and assume that (34) holds for fuzzy goal \tilde{d} . Then the vector $(t^*, x^*) \in \mathbb{R}^{n+1}$ is an optimal solution of the optimization problem

$$\begin{aligned}&\text{maximize} \quad t \\ &\text{subject to} \quad \mu_{\tilde{X}_i}(x) \geq t, \quad i \in \{0\} \cup \mathcal{M}, \\ &\quad x_j \geq 0, \quad j \in \mathcal{N}\end{aligned}\tag{35}$$

if and only if $x^* \in \mathbb{R}^n$ is a max-satisficing solution of FLP problem (24).

In practice, one of possible ways of how to choose the appropriate fuzzy goal \tilde{d} is to set in advance two values: the lower and upper limits and approximate the corresponding membership function of \tilde{d} by the linear function. It means that in case of the ‘maximization problem’ the lower limit is the highest value with the membership grade of the goal equal to zero, and, on the other hand, the upper limit is the lowest value where the membership grade is equal to 1 (‘full satisfaction’). The resulting membership function of \tilde{d} is then non-decreasing and piecewise linear. For minimization problems it can be done similarly.

31.5.2 α -efficient Solution

Now, let \tilde{a} and \tilde{b} be fuzzy quantities and \tilde{R} be a fuzzy relation on \mathbb{R} , $\alpha \in (0, 1]$. We write

$$\tilde{a} \lesssim_{\alpha}^{\tilde{R}} \tilde{b}, \text{ if } \mu_{\tilde{R}}(\tilde{a}, \tilde{b}) \geq \alpha.$$

We also write

$$\tilde{a} \prec_{\alpha}^{\tilde{R}} \tilde{b}, \text{ if } \tilde{a} \lesssim_{\alpha}^{\tilde{R}} \tilde{b} \text{ and } \mu_{\tilde{R}}(\tilde{b}, \tilde{a}) < \alpha.$$

Notice that $\lesssim_{\alpha}^{\tilde{R}}$ is a binary relation on the set of all fuzzy quantities $\mathcal{F}_0(\mathbb{R})$. If \tilde{a} and \tilde{b} are real numbers a and b , respectively, and \tilde{R} is a fuzzy extension of relation \leq , then $\tilde{a} \lesssim_{\alpha}^{\tilde{R}} \tilde{b}$ if and only if $a \leq b$.

Now, modifying the well-known concept of efficient (nondominated) solution of linear programming problem we define ‘maximization’ (or ‘minimization’) of the objective function of FLP problem (24).

Definition 28. Let \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i , $i \in \mathcal{M}$, $j \in \mathcal{N}$, be fuzzy quantities on \mathbb{R} . Let \tilde{R}_i , $i \in 0, 1, 2, \dots, m$, be fuzzy relations on \mathbb{R} and $\alpha \in (0, 1]$. Let $x = (x_1, \dots, x_n)^T$ be an α -feasible solution of (24) and denote $\tilde{c}^T x = \tilde{c}_1 x_1 + \dots + \tilde{c}_n x_n$. The vector $x \in \mathbb{R}^n$ is an α -efficient solution of (24) with maximization of the objective function if there is no $x' \in [\tilde{X}]_{\alpha}$ such that $\tilde{c}^T x' \prec_{\alpha}^{\tilde{R}_0} \tilde{c}^T x$. Similarly, the vector x is an α -efficient solution of (24) with minimization of the objective function if there is no $x' \in [\tilde{X}]_{\alpha}$ such that $\tilde{c}^T x' \prec_{\alpha}^{\tilde{R}_0} \tilde{c}^T x$.

Notice that any α -efficient solution of the FLP problem is an α -feasible solution of the FLP problem with some additional property. If all coefficients of FLP problem (24) are real numbers, then the α -efficient solution of the FLP problem is equivalent to the classical optimal solution of the corresponding linear programming problem.

In practice, the level α of efficiency of the solution (e.g., the degree of possibility or necessity) is chosen by the decision maker according to the nature of the problem in advance and depends on the required efficiency of the solution. Usual values range from 0.6 to 0.9.

In the following theorem we show some necessary and sufficient conditions for α -efficient solution of (24) in case of special fuzzy extensions of the binary relation \leq .

Theorem 29. Let \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i , $i \in \mathcal{M}$, $j \in \mathcal{N}$, be fuzzy quantities, $\alpha \in (0, 1)$.

(i) Let $\tilde{R}_i = \tilde{\leq}^{\min}$; i.e., \tilde{R}_i be a fuzzy extension of the binary relation \leq on \mathbb{R} defined by (13) and (14) for all $i \in 0, 1, 2, \dots, m$. Let $x^* = (x_1^*, \dots, x_n^*)^T$, $x_j^* \geq 0$, $j \in \mathcal{N}$, be an α -feasible solution of (24). Then the vector $x^* \in \mathbb{R}^n$ is an α -efficient solution of (24) with maximization of the objective function if and only if x^* is an optimal solution of the following linear programming problem:

$$\begin{aligned} &\text{maximize} && \tilde{c}_1^R(\alpha)x_1 + \dots + \tilde{c}_n^R(\alpha)x_n \\ &\text{subject to} && \tilde{a}_{i1}^L(\alpha)x_1 + \dots + \tilde{a}_{in}^L(\alpha)x_n \leq \tilde{b}_i^R(\alpha), \quad i \in \mathcal{M}, \\ & && x_j \geq 0, \quad j \in \mathcal{N}. \end{aligned}$$

(ii) Let $\tilde{R}_0 = \tilde{\leq}^{\min}$, $\tilde{R}_i = \tilde{\leq}^{\max}$, $i \in 1, 2, \dots, m$. Let $x^* = (x_1^*, \dots, x_n^*)^T$, $x_j^* \geq 0$, $j \in \mathcal{N}$, be an α -feasible solution of (24). Then the vector $x^* \in \mathbb{R}^n$ is an α -efficient solution of (24) with maximization of the

objective function if and only if x^* is an optimal solution of the following linear programming problem:

$$\begin{aligned} & \text{maximize} && \tilde{c}_1^R(\alpha)x_1 + \cdots + \tilde{c}_n^R(\alpha)x_n \\ & \text{subject to} && \tilde{a}_{i1}^R(\alpha)x_1 + \cdots + \tilde{a}_{in}^R(\alpha)x_n \leq \tilde{b}_i^L(\alpha), \quad i \in \mathcal{M}, \\ & && x_j \geq 0, \quad j \in \mathcal{N}. \end{aligned}$$

(iii) Let $\tilde{R}_0 = \tilde{\leq}^{\max}$, $\tilde{R}_i = \tilde{\leq}^{\min}$, $i \in 1, 2, \dots, m$. Let $x^* = (x_1^*, \dots, x_n^*)^T$, $x_j^* \geq 0$, $j \in \mathcal{N}$, be an α -feasible solution of (24). Then the vector $x^* \in \mathbb{R}^n$ is an α -efficient solution of (24) with maximization of the objective function if and only if x^* is an optimal solution of the following linear programming problem:

$$\begin{aligned} & \text{maximize} && \tilde{c}_1^L(\alpha)x_1 + \cdots + \tilde{c}_n^L(\alpha)x_n \\ & \text{subject to} && \tilde{a}_{i1}^L(\alpha)x_1 + \cdots + \tilde{a}_{in}^L(\alpha)x_n \leq \tilde{b}_i^R(\alpha), \quad i \in \mathcal{M}, \\ & && x_j \geq 0, \quad j \in \mathcal{N}. \end{aligned}$$

(iv) Let $\tilde{R}_i = \tilde{\leq}^{\max}$, $i \in 0, 1, 2, \dots, m$. Let $x^* = (x_1^*, \dots, x_n^*)^T$, $x_j^* \geq 0$, $j \in \mathcal{N}$, be an α -feasible solution of (24). Then the vector $x^* \in \mathbb{R}^n$ is an α -efficient solution of (24) with maximization of the objective function if and only if x^* is an optimal solution of the following linear programming problem:

$$\begin{aligned} & \text{maximize} && \tilde{c}_1^L(\alpha)x_1 + \cdots + \tilde{c}_n^L(\alpha)x_n \\ & \text{subject to} && \tilde{a}_{i1}^R(\alpha)x_1 + \cdots + \tilde{a}_{in}^R(\alpha)x_n \leq \tilde{b}_i^L(\alpha), \quad i \in \mathcal{M}, \\ & && x_j \geq 0, \quad j \in \mathcal{N}. \end{aligned}$$

In the following section we shall investigate duality – a fundamental concept of linear optimization. Again we shall distinguish the above-mentioned two approaches to ‘optimality’ in FLP.

31.6 Duality in FLP

In this section we generalize the well-known concept of duality in linear programming for FLP problems. Some results of this section can also be found in [39]. We derive some weak and strong duality theorems which extend the known results for linear programming problems.

Consider the following FLP problem:

$$\begin{aligned} & \text{‘maximize’} && \tilde{c}_1 x_1 \tilde{+} \cdots \tilde{+} \tilde{c}_n x_n \\ & \text{subject to} && (\tilde{a}_{i1} x_1 \tilde{+} \cdots \tilde{+} \tilde{a}_{in} x_n) \tilde{R} \tilde{b}_i, \quad i \in \mathcal{M}, \\ & && x_j \geq 0, \quad j \in \mathcal{N}, \end{aligned} \tag{36}$$

where \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i are normal fuzzy quantities with membership functions $\mu_{\tilde{c}_j} : \mathbb{R} \rightarrow [0, 1]$, $\mu_{\tilde{a}_{ij}} : \mathbb{R} \rightarrow [0, 1]$ and $\mu_{\tilde{b}_i} : \mathbb{R} \rightarrow [0, 1]$, $i \in \mathcal{M}$, $j \in \mathcal{N}$.

Let $\Phi : \mathcal{F}(\mathbb{R} \times \mathbb{R}) \rightarrow \mathcal{F}(\mathcal{F}(\mathbb{R}) \times \mathcal{F}(\mathbb{R}))$ be a mapping and $\Psi : \mathcal{F}(\mathbb{R} \times \mathbb{R}) \rightarrow \mathcal{F}(\mathcal{F}(\mathbb{R}) \times \mathcal{F}(\mathbb{R}))$ be the dual mapping to mapping Φ . Let R be a valued relation on \mathbb{R} and let $\tilde{R} = \Phi(R)$, $\tilde{R}^D = \Psi(R)$ (see Definition 10). Then \tilde{R} and \tilde{R}^D are dual fuzzy relations.

FLP problem (36) will be called the *primal FLP problem (P)*.

The *dual FLP problem (D)* is defined as

$$\begin{aligned} & \text{‘minimize’} && \tilde{b}_1 y_1 \tilde{+} \cdots \tilde{+} \tilde{b}_m y_m \\ & \text{subject to} && \tilde{c}_j \tilde{R}^D (\tilde{a}_{1j} y_1 \tilde{+} \cdots \tilde{+} \tilde{a}_{mj} y_m), \quad j \in \mathcal{N}, \\ & && y_i \geq 0, \quad i \in \mathcal{M}. \end{aligned} \tag{37}$$

The pair of FLP problems (36) and (37) is called the *primal – dual pair of FLP problems*.

Let R be the binary operation \leq and let $T = \min$, $S = \max$. Let $\tilde{\leq}^{\min}$ and $\tilde{\leq}^{\max}$ be fuzzy extensions defined by (13) and (14), respectively. Since T is the dual t -norm to S , by Definition 10, $\tilde{\leq}^{\max}$ is the dual fuzzy relation to $\tilde{\leq}^{\min}$. We obtain the primal–dual pair of FLP problems as follows:

(P)

$$\begin{aligned} & \text{‘maximize’} \quad \tilde{c}_1 x_1 + \cdots + \tilde{c}_n x_n \\ & \text{subject to} \quad \tilde{a}_{i1} x_1 + \cdots + \tilde{a}_{in} x_n \tilde{\leq}^{\min} \tilde{b}_i, \quad i \in \mathcal{M}, \\ & \quad x_j \geq 0, \quad j \in \mathcal{N}. \end{aligned} \quad (38)$$

(D)

$$\begin{aligned} & \text{‘minimize’} \quad \tilde{b}_1 y_1 + \cdots + \tilde{b}_m y_m \\ & \text{subject to} \quad \tilde{c}_j \tilde{\leq}^{\max} \tilde{a}_{1j} y_1 + \cdots + \tilde{a}_{mj} y_m, \quad j \in \mathcal{N}, \\ & \quad y_i \geq 0, \quad i \in \mathcal{M}. \end{aligned} \quad (39)$$

Let the feasible solution of the primal FLP problem (P) be denoted by \tilde{X} and the feasible solution of the dual FLP problem (D) by \tilde{Y} . Clearly, \tilde{X} is a fuzzy subset of \mathbb{R}^n and \tilde{Y} is a fuzzy subset of \mathbb{R}^m .

Notice that in the crispnon-fuzzy case, i.e., when the parameters \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i are real numbers, by Theorem 21 the relations $\tilde{\leq}^{\min}$ and $\tilde{\leq}^{\max}$ coincide with \leq ; hence, (P) and (D) is a primal – dual pair of linear programming problems in the classical sense. The following proposition is a useful modification of Theorem 21.

Proposition 30. *Let \tilde{c}_j and \tilde{a}_{ij} be fuzzy quantities and let $y_i \geq 0$ for all $i \in \mathcal{M}$, $j \in \mathcal{N}$, $\alpha \in (0, 1)$. Let $\tilde{\leq}^{\max}$ be a fuzzy extension of the binary relation \geq defined by Definition 15. Then for $j \in \mathcal{N}$ it holds $\mu_{\tilde{\leq}^{\max}}(\tilde{a}_{1j} y_1 + \cdots + \tilde{a}_{mj} y_m, \tilde{c}_j) \geq 1 - \alpha$ if and only if*

$$\sum_{i \in \mathcal{M}} \tilde{a}_{ij}^L(\alpha) y_i \geq \tilde{c}_j^R(\alpha).$$

In the following theorem we prove the weak form of the duality theorem for FLP problems.

Theorem 31. First weak duality theorem: *Let \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i be fuzzy quantities for all $i \in \mathcal{M}$ and $j \in \mathcal{N}$. Let $A = T_M = \min$, $S = S_M = \max$, and $\alpha \in (0, 1)$. Let \tilde{X} be a feasible solution of FLP problem (36) and \tilde{Y} be a feasible solution of FLP problem (37).*

If a vector $x = (x_1, \dots, x_n)^T \geq 0$ belongs to $[\tilde{X}]_\alpha$ and $y = (y_1, \dots, y_m)^T \geq 0$ belongs to $[\tilde{Y}]_{1-\alpha}$, then

$$\sum_{j \in \mathcal{N}} \tilde{c}_j^R(\alpha) x_j \leq \sum_{i \in \mathcal{M}} \tilde{b}_i^R(\alpha) y_i.$$

Theorem 32. Second weak duality theorem: *Let \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i be fuzzy quantities for all $i \in \mathcal{M}$ and $j \in \mathcal{N}$. Let $A = T_M = \min$, $S = S_M = \max$, and $\alpha \in (0, 1)$. Let \tilde{X} be a feasible solution of FLP problem (36) and \tilde{Y} be a feasible solution of FLP problem (37).*

If for some $x = (x_1, \dots, x_n)^T \geq 0$ belonging to $[\tilde{X}]_\alpha$ and $y = (y_1, \dots, y_m)^T \geq 0$ belonging to $[\tilde{Y}]_{1-\alpha}$ it holds

$$\sum_{j \in \mathcal{N}} \tilde{c}_j^R(\alpha) x_j = \sum_{i \in \mathcal{M}} \tilde{b}_i^R(\alpha) y_i,$$

then x is an α -efficient solutions of FLP problem (P) and y is an $(1 - \alpha)$ -efficient solutions of FLP problem (D).

Remark. 1. In the case of real parameters, Theorems 31 and 32 are standard linear programming weak duality theorems.

2. The result of the first weak duality theorem is independent of the ‘maximization’ or ‘minimization’ approach.

3. By analogy we can easily formulate the primal–dual pair of FLP problems interchanging the fuzzy relations \leq^{\min} and \leq^{\max} in the objective functions and/or constraints of (36) and (37). Then the weak duality theorems should be appropriately modified.

4. Let $\alpha \geq 0.5$. It is clear that $[\tilde{Y}]_\alpha \subseteq [\tilde{Y}]_{1-\alpha}$. In the weak duality theorems we can change the assumptions as follows: $x \in [\tilde{X}]_\alpha$ and $y \in [\tilde{Y}]_\alpha$. Evidently, the statements of the theorems will remain unchanged.

Let us turn to the *strong duality*. We start with ‘satisficing’ approach to ‘maximization’ or ‘minimization’.

For this purpose, we assume the existence of exogenously given additional fuzzy goals $\tilde{d} \in \mathcal{F}(\mathbb{R})$ and $\tilde{h} \in \mathcal{F}(\mathbb{R})$. The fuzzy goal \tilde{d} is compared to fuzzy values $\tilde{c}_1 x_1 + \dots + \tilde{c}_n x_n$ of the objective function of the primal FLP problem (*P*) by fuzzy relation \leq^{\min} . On the other hand, the fuzzy goal \tilde{h} is compared to fuzzy values $\tilde{b}_1 y_1 + \dots + \tilde{b}_m y_m$ of the objective function of the dual FLP problem (*D*) by fuzzy relation \leq^{\max} . In this way we treat the fuzzy objectives as constraints

$$\tilde{c}_1 x_1 + \dots + \tilde{c}_n x_n \leq^{\min} \tilde{d}, \quad \tilde{b}_1 y_1 + \dots + \tilde{b}_m y_m \leq^{\max} \tilde{h}.$$

By \tilde{X}^* we denote the satisficing solution of the primal FLP problem (*P*), defined by Definition 23, by \tilde{Y}^* , the satisficing solution of the dual FLP problem (*D*) is denoted. Clearly, \tilde{X}^* is a fuzzy subset of \mathbb{R}^n and \tilde{Y}^* is a fuzzy subset of \mathbb{R}^m ; moreover, $\tilde{X}^* \subseteq \tilde{X}$ and $\tilde{Y}^* \subseteq \tilde{Y}$.

Theorem 33. First strong duality theorem: *Let \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i be fuzzy quantities for all $i \in \mathcal{M}$ and $j \in \mathcal{N}$. Let $\tilde{d}, \tilde{h} \in \mathcal{F}(\mathbb{R})$ be fuzzy goals with the membership functions $\mu_{\tilde{d}}$ and $\mu_{\tilde{h}}$ satisfying the following conditions*

- both $\mu_{\tilde{d}}$ and $\mu_{\tilde{h}}$ are upper semicontinuous,*
- $\mu_{\tilde{d}}$ is strictly increasing and $\mu_{\tilde{h}}$ is strictly decreasing,*
- $$\lim_{t \rightarrow -\infty} \mu_{\tilde{d}}(t) = \lim_{t \rightarrow +\infty} \mu_{\tilde{h}}(t) = 0.$$

Let $G = T = \min$ and $S = \max$. Let \leq^{\min} be the T -fuzzy extension of the binary relation \leq on \mathbb{R} and \leq^{\max} be the S -fuzzy extension of the relation \leq on \mathbb{R} . Let \tilde{X}^ be a satisficing solution of FLP problem (38), \tilde{Y}^* be a satisficing solution of FLP problem (39), and $\alpha \in (0, 1)$.*

If a vector $x^ = (x_1^*, \dots, x_n^*)^T \geq 0$ belongs to $[\tilde{X}^*]_\alpha$, then there exists a vector $y^* = (y_1^*, \dots, y_m^*)^T \geq 0$ which belongs to $[\tilde{Y}^*]_{1-\alpha}$, and*

$$\sum_{j \in \mathcal{N}} \tilde{c}_j^R(\alpha) x_j^* = \sum_{i \in \mathcal{M}} \tilde{b}_i^R(\alpha) y_i^*. \quad (40)$$

Notice that in the non-fuzzy case, (40) is the standard strong duality result for linear programming. Now we turn to the α -efficient approach to optimization of FLP problems.

By X_α^* we denote the α -efficient solution of the primal FLP problem (*P*), defined by Definition 28, analogically, by Y_α^* the α -efficient solution of the dual FLP problem (*D*) is denoted.

Theorem 34. Second strong duality theorem: *Let \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i be fuzzy quantities for all $i \in \mathcal{M}$ and $j \in \mathcal{N}$. Let \leq^{\min} and \leq^{\max} be fuzzy extensions of the binary relation \leq , and $\alpha \in (0, 1)$. If $[\tilde{X}]_\alpha$ and $[\tilde{Y}]_{1-\alpha}$ are non-empty, then there exists x^* – an α -efficient solutions of FLP problem (*P*), and y^* – an $(1-\alpha)$ -efficient solutions of FLP problem (*D*) such that*

$$\sum_{j \in \mathcal{N}} \tilde{c}_j^R(\alpha) x_j^* = \sum_{i \in \mathcal{M}} \tilde{b}_i^R(\alpha) y_i^*.$$

Particularly, in the non-fuzzy case, Theorem 34 is in fact the strong duality result for standard linear programming. The question arises how the theorems could be modified for more general t -norms and t -conorms.

31.7 Extended Operations

Up till now, in Proposition 18, formulas (25) and (26) and many others we have used addition of fuzzy values by the t -norm $T_M = \min$. In this section, we shall investigate addition of fuzzy quantities using a more general t -norm T ; particularly, we denote

$$\tilde{f} = \tilde{c}_1 x_1 \tilde{+}_T \cdots \tilde{+}_T \tilde{c}_n x_n, \quad (41)$$

and

$$\tilde{g}_i = \tilde{a}_{i1} x_1 \tilde{+}_T \cdots \tilde{+}_T \tilde{a}_{in} x_n, \quad (42)$$

for each $x \in \mathbb{R}^n$, where $\tilde{c}_j, \tilde{a}_{ij} \in \mathcal{F}(\mathbb{R})$, for all $i \in \mathcal{M}, j \in \mathcal{N}$. The extended addition $\tilde{+}_T$ in (41) and (42) is defined by using of the extension principle. The membership functions of (41) and (42) is defined as follows:

$$\mu_{\tilde{f}}(t) = \sup\{T(\mu_{\tilde{c}_1}(c_1), \dots, \mu_{\tilde{c}_n}(c_n)) | t = c_1 x_1 + \cdots + c_n x_n\}, \quad (43)$$

$$\mu_{\tilde{g}_i}(t) = \sup\{T(\mu_{\tilde{a}_{i1}}(a_{i1}), \dots, \mu_{\tilde{a}_{in}}(a_{in})) | t = a_{i1} x_1 + \cdots + a_{in} x_n\}. \quad (44)$$

Formulas (41), (42) or (43), (44) can be difficult to obtain; however, in some special cases analytical formulas can be derived.

For the sake of brevity we deal only with (41); formula (42) can be obtained analogously. We derive special formulas for a broad class of fuzzy values (i.e., coefficients of the FLP problem) generated by the same functions.

Let $\Phi, \Psi : (0, +\infty) \rightarrow [0, 1]$ be non-increasing, semistrictly quasiconcave, and upper-semicontinuous functions. Given $\gamma, \delta \in (0, +\infty)$, define functions $\Phi_\gamma, \Psi_\delta : (0, +\infty) \rightarrow [0, 1]$ for $x \in (0, +\infty)$ by

$$\Phi_\gamma(x) = \Phi\left(\frac{x}{\gamma}\right), \quad \Psi_\delta(x) = \Psi\left(\frac{x}{\delta}\right).$$

Let $l_j, r_j \in \mathbb{R}$ such that $l_j \leq r_j$, let $\gamma_j, \delta_j \in (0, +\infty)$, and let

$$\tilde{c}_j = (l_j, r_j, \Phi_{\gamma_j}, \Psi_{\delta_j}), \quad j \in \mathcal{N},$$

denote fuzzy intervals with the membership functions given by

$$\mu_{\tilde{c}_j}(x) = \begin{cases} \Phi_{\gamma_j}(l_j - x) & \text{if } x \in (-\infty, l_j), \\ 1 & \text{if } x \in [l_j, r_j], \\ \Psi_{\delta_j}(x - r_j) & \text{if } x \in (r_j, +\infty). \end{cases} \quad (45)$$

The following proposition shows that $\tilde{c}_1 x_1 \tilde{+}_T \cdots \tilde{+}_T \tilde{c}_n x_n$ is a closed fuzzy quantity of the same type for particular t -norms T . The proof is straightforward and is omitted here.

Proposition 35. Let $\tilde{c}_j = (l_j, r_j, \Phi_{\gamma_j}, \Psi_{\delta_j})$, $j \in \mathcal{N}$, be fuzzy quantities with the membership functions given by (45). For $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, $x_j \geq 0$ for all $j \in \mathcal{N}$, define I_x by

$$I_x = \{j \mid x_j > 0, j \in \mathcal{N}\}.$$

Then

$$\begin{aligned} \tilde{c}_1 x_1 \tilde{+}_{T_M} \cdots \tilde{+}_{T_M} \tilde{c}_n x_n &= (l, r, \Phi_{l_M}, \Psi_{r_M}), \\ \tilde{c}_1 x_1 \tilde{+}_{T_D} \cdots \tilde{+}_{T_D} \tilde{c}_n x_n &= (l, r, \Phi_{l_D}, \Psi_{r_D}), \end{aligned} \quad (46)$$

where T_M is the minimum t-norm, T_D is the drastic product, and

$$\begin{aligned} l &= \sum_{j \in I_x} l_j x_j, \quad r = \sum_{j \in I_x} r_j x_j, \\ l_M &= \sum_{j \in I_x} \frac{\gamma_j}{x_j}, \quad r_M = \sum_{j \in I_x} \frac{\delta_j}{x_j}, \\ l_D &= \max \left\{ \frac{\gamma_j}{x_j} \mid j \in I_x \right\}, \quad r_D = \max \left\{ \frac{\delta_j}{x_j} \mid j \in I_x \right\}. \end{aligned}$$

If all \tilde{c}_j are $(\mathcal{L}, \mathcal{R})$ -fuzzy intervals, then an analogous and more specific result can be obtained. Let $l_j, r_j \in \mathbb{R}$ with $l_j \leq r_j$, let $\gamma_j, \delta_j \in [0, +\infty)$, and let \mathcal{L}, \mathcal{R} be non-increasing, semistrictly quasiconcave, upper-semicontinuous functions from $(0, 1]$ into $[0, +\infty)$. Moreover, assume that $\mathcal{L}(1) = \mathcal{R}(1) = 0$ and define $\mathcal{L}(0) = \lim_{x \rightarrow 0} \mathcal{L}(x)$, $\mathcal{R}(0) = \lim_{x \rightarrow 0} \mathcal{R}(x)$.

Let

$$\tilde{c}_j = (l_j, r_j, \gamma_j, \delta_j)_{\mathcal{LR}}$$

be an $(\mathcal{L}, \mathcal{R})$ -fuzzy interval given by the membership function defined for each $x \in \mathbb{R}$ and for every $j \in \mathcal{N}$, by

$$\mu_{\tilde{c}_j}(x) = \begin{cases} \mathcal{L}^{(-1)}\left(\frac{l_j - x}{\gamma_j}\right) & \text{if } x \in (l_j - \gamma_j, l_j), \gamma_j > 0, \\ 1 & \text{if } x \in [l_j, r_j], \\ \mathcal{R}^{(-1)}\left(\frac{x - r_j}{\delta_j}\right) & \text{if } x \in (r_j, r_j + \delta_j), \delta_j > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (47)$$

where $\mathcal{L}^{(-1)}, \mathcal{R}^{(-1)}$ are pseudo-inverse functions of \mathcal{L}, \mathcal{R} , respectively. We obtain the following result:

Proposition 36. Let $\tilde{c}_j = (l_j, r_j, \gamma_j, \delta_j)_{\mathcal{LR}}, j \in \mathcal{N}$, be $(\mathcal{L}, \mathcal{R})$ -fuzzy intervals with the membership functions given by (47) and let $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, $x_j \geq 0$ for all $j \in \mathcal{N}$. Then

$$\begin{aligned} \tilde{c}_1 x_1 \tilde{+}_{T_M} \cdots \tilde{+}_{T_M} \tilde{c}_n x_n &= (l, r, A_M, B_M)_{\mathcal{LR}}, \\ \tilde{c}_1 x_1 \tilde{+}_{T_D} \cdots \tilde{+}_{T_D} \tilde{c}_n x_n &= (l, r, A_D, B_D)_{\mathcal{LR}}, \end{aligned} \quad (48)$$

where T_M is the minimum t-norm, T_D is the drastic product, and

$$\begin{aligned} l &= \sum_{j \in \mathcal{N}} l_j x_j, \quad r = \sum_{j \in \mathcal{N}} r_j x_j, \\ A_M &= \sum_{j \in \mathcal{N}} \gamma_j x_j, \quad B_M = \sum_{j \in \mathcal{N}} \delta_j x_j, \\ A_D &= \max\{\gamma_j \mid j \in \mathcal{N}\}, \quad B_D = \max\{\delta_j \mid j \in \mathcal{N}\}. \end{aligned}$$

The results (46) and (48) in Proposition 35 and 36, respectively, can be extended as follows (see also [12]).

Proposition 37. Let T be a continuous Archimedean t-norm with an additive generator f . Let $\Phi : (0, +\infty) \rightarrow [0, 1]$ be defined for each $x \in (0, +\infty)$ as

$$\Phi(x) = f^{(-1)}(x).$$

Let $\tilde{c}_j = (l_j, r_j, \Phi_{\gamma_j}, \Phi_{\delta_j}), j \in \mathcal{N}$, be closed fuzzy intervals with the membership functions given by (45) and let $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, $x_j \geq 0$ for all $j \in \mathcal{N}$, $I_x = \{j \mid x_j > 0, j \in \mathcal{N}\}$. Then

$$\tilde{c}_1 x_1 \tilde{+}_T \cdots \tilde{+}_T \tilde{c}_n x_n = (l, r, \Phi_{l_D}, \Phi_{r_D}),$$

where

$$\begin{aligned} l &= \sum_{j \in I_x} l_j x_j, \quad r = \sum_{j \in I_x} r_j x_j, \\ l_D &= \max \left\{ \frac{\gamma_j}{x_j} \mid j \in I_x \right\}, \quad r_D = \max \left\{ \frac{\delta_j}{x_j} \mid j \in I_x \right\}. \end{aligned}$$

For a continuous Archimedean t-norm T and closed fuzzy intervals \tilde{c}_j satisfying the assumptions of Proposition 37, we easily obtain

$$\tilde{c}_1 x_1 \tilde{+}_T \cdots \tilde{+}_T \tilde{c}_n x_n = \tilde{c}_1 x_1 \tilde{+}_{T_D} \cdots \tilde{+}_{T_D} \tilde{c}_n x_n, \quad (49)$$

which means that we obtain the same fuzzy linear function based on an arbitrary t-norm T' such that $T' \leq T$.

The result which follows generalizes a result concerning the addition of closed fuzzy intervals based on continuous Archimedean t -norms.

Proposition 38. *Let T be a continuous Archimedean t -norm with an additive generator f . Let $K : [0, +\infty) \rightarrow [0, +\infty)$ be continuous convex function with $K(0) = 0$. Let $\alpha \in (0, +\infty)$ and*

$$\Phi_\alpha(x) = f^{(-1)} \left(\alpha K \left(\frac{x}{\alpha} \right) \right) \quad (50)$$

for all $x \in [0, +\infty)$. Let $\tilde{c}_j = (l_j, r_j, \Phi_{\gamma_j}, \Phi_{\delta_j})$, $j \in \mathcal{N}$, be closed fuzzy intervals with the membership functions given by (45) and let $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, $x_j \geq 0$ for all $j \in \mathcal{N}$, $I_x = \{j \mid x_j > 0, j \in \mathcal{N}\}$. Then

$$\tilde{c}_1 x_1 \tilde{+}_T \cdots \tilde{+}_T \tilde{c}_n x_n = (l, r, \Phi_{l_K}, \Phi_{r_K}),$$

where

$$\begin{aligned} l &= \sum_{j \in I_x} l_j x_j, \quad r = \sum_{j \in I_x} r_j x_j, \\ l_K &= \sum_{j \in I_x} \frac{\gamma_j}{x_j}, \quad r_K = \sum_{j \in I_x} \frac{\delta_j}{x_j}. \end{aligned}$$

31.8 Special Models of FLP

Three types of FLP problem known from the literature are investigated in this section. We start with the oldest version of FLP problem, originally called fuzzy (linear) programming problem (see [34]). Later on, e.g., in [24], this problem was named flexible linear programming problem.

31.8.1 Flexible Linear Programming

Flexible linear programming is referred to the approach to linear programming problems allowing for a kind of flexibility of the objective function and constraints in standard linear programming problem (22). Consider

$$\begin{aligned} &\text{maximize} && c_1 x_1 + \cdots + c_n x_n \\ &\text{subject to} && a_{i1} x_1 + \cdots + a_{in} x_n \leq b_i, \quad i \in \mathcal{M}, \\ & && x_j \geq 0, \quad j \in \mathcal{N}. \end{aligned} \quad (51)$$

The values of parameters c_j , a_{ij} , and b_i in (51) are supposed to be subjected to some uncertainty. By non-negative values p_i , $i \in \{0\} \cup \mathcal{M}$, admissible violations of the objective and constraints are (subjectively) chosen and introduced to the original model (51).

An aspiration level $d_0 \in \mathbb{R}$ is (subjectively) determined such that the decision maker (DM) is fully satisfied on condition the value of the objective function is greater than or equal to d_0 . On the other hand, if the objective function attains a value smaller than $d_0 - p_0$, then DM is fully dissatisfied. Within the interval $(d_0 - p_0, d_0)$, the satisfaction of DM increases (e.g., linearly) from 0 to 1. Under these assumptions a membership function $\mu_{\tilde{d}}$ of the fuzzy goal \tilde{d} could be defined as follows:

$$\mu_{\tilde{d}}(t) = \begin{cases} 1 & \text{if } t \geq d_0, \\ 1 + \frac{t-d_0}{p_0} & \text{if } d_0 - p_0 \leq t < d_0, \\ 0 & \text{otherwise.} \end{cases} \quad (52)$$

Now, let for the i th constraint function of (51), $i \in \mathcal{M}$, a right-hand side $b_i \in \mathbb{R}$ is known such that then the DM is fully satisfied on the condition that left-hand side is less than or equal to this value. On the other hand, if the objective function is greater than $b_i + p_i$, then the DM is fully dissatisfied. Within the interval $(b_i, b_i + p_i)$, the satisfaction of DM decreases (linearly) from 1 to 0. Under these assumptions the membership function $\mu_{\tilde{b}_i}$ of the fuzzy right-hand side \tilde{b}_i is defined as

$$\mu_{\tilde{b}_i}(t) = \begin{cases} 1 & \text{if } t \leq b_i, \\ 1 - \frac{t-b_i}{p_i} & \text{if } b_i \leq t < b_i + p_i, \\ 0 & \text{otherwise.} \end{cases} \quad (53)$$

The relationship between the objective function and constraints in the flexible linear programming problem is symmetric; i.e., there is no a difference between the former and the latter. ‘Maximization’ is understood as finding a vector $x \in \mathbb{R}^n$ such that the membership grade of the intersection of fuzzy sets (52) and (53) is maximized. This problem is equivalent to the following optimization problem:

$$\begin{aligned} & \text{maximize } \lambda \\ & \text{subject to } \mu_{\tilde{d}} \left(\sum_{j \in \mathcal{N}} c_j x_j \right) \geq \lambda, \\ & \mu_{\tilde{b}_i} \left(\sum_{j \in \mathcal{N}} a_{ij} x_j \right) \geq \lambda, \quad i \in \mathcal{M}, \\ & 0 \leq \lambda \leq 1, \\ & x_j \geq 0, \quad j \in \mathcal{N}. \end{aligned} \quad (54)$$

Problem (54) can easily be transformed to the equivalent linear programming problem:

$$\begin{aligned} & \text{maximize } \lambda \\ & \text{subject to } \sum_{j \in \mathcal{N}} c_j x_j \geq d_0 + \lambda p_0, \\ & \sum_{j \in \mathcal{N}} a_{ij} x_j \leq b_i + (1 - \lambda) p_i, \quad i \in \mathcal{M}, \\ & 0 \leq \lambda \leq 1, \\ & x_j \geq 0, \quad j \in \mathcal{N}. \end{aligned} \quad (55)$$

Now, consider a more specific FLP problem:

$$\begin{aligned} & \text{maximize } c_1 x_1 + \cdots + c_n x_n \\ & \text{subject to } a_{i1} x_1 + \cdots + a_{in} x_n \geq^T \tilde{b}_i, \quad i \in \mathcal{M}, \\ & x_j \geq 0, \quad j \in \mathcal{N}, \end{aligned} \quad (56)$$

where c_j , a_{ij} , and b_i are real numbers, whereas \tilde{d} and \tilde{b}_i are fuzzy quantities defined by (52) and (53). Moreover, $\tilde{\leq}^T$ is a T -fuzzy extension of the usual inequality relation \leq , with $T = \min$. It turns out that the vector $x \in \mathbb{R}^n$ is an optimal solution of flexible linear programming problem (55) if and only if it is a max-satisficing solution of FLP problem (56). This result follows directly from Proposition 27.

31.8.2 Interval Linear Programming

In this subsection we apply the results of this chapter to a special case of the FLP problem – interval linear programming (ILP) problem. By ILP we understand the following FLP problem:

$$\begin{aligned} & \text{maximize} && \tilde{c}_1 x_1 + \cdots + \tilde{c}_n x_n \\ & \text{subject to} && \tilde{a}_{i1} x_1 + \cdots + \tilde{a}_{in} x_n \tilde{R} \tilde{b}_i, \quad i \in \mathcal{M}, \\ & && x_j \geq 0, \quad j \in \mathcal{N}, \end{aligned} \quad (57)$$

where \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i are considered to be compact intervals in \mathbb{R} ; i.e., $\tilde{c}_j = [\underline{c}_j, \bar{c}_j]$, $\tilde{a}_{ij} = [\underline{a}_{ij}, \bar{a}_{ij}]$, and $\tilde{b}_i = [\underline{b}_i, \bar{b}_i]$, where \underline{c}_j , \bar{c}_j , \underline{a}_{ij} , \bar{a}_{ij} , and \underline{b}_i , \bar{b}_i are lower and upper bounds of the corresponding intervals, respectively. Let the membership functions of \tilde{c}_j , \tilde{a}_{ij} , and \tilde{b}_i be the characteristic functions of the intervals; i.e., $\chi_{[\underline{c}_j, \bar{c}_j]} : \mathbb{R} \rightarrow [0, 1]$, $\chi_{[\underline{a}_{ij}, \bar{a}_{ij}]} : \mathbb{R} \rightarrow [0, 1]$, and $\chi_{[\underline{b}_i, \bar{b}_i]} : \mathbb{R} \rightarrow [0, 1]$, $i \in \mathcal{M}$, $j \in \mathcal{N}$.

Now, we assume that R is the usual binary relation \leq , and $A = T = \min$, $S = \max$. The fuzzy relation \tilde{R} is the fuzzy extension of a valued relation \leq . We shall consider 6 fuzzy relations \tilde{R} -extensions of the binary relation \leq , defined by (13) and (14) and by (15)–(18); i.e.,

$$\tilde{R} \in \left\{ \tilde{\leq}^{\min}, \tilde{\leq}^{\max}, \tilde{\leq}^{T,S}, \tilde{\leq}_{T,S}, \tilde{\leq}^{S,T}, \tilde{\leq}_{S,T} \right\}.$$

Then by Proposition 21 we obtain six types of feasible solutions of ILP problem (57):

$$X_{\tilde{\leq}^{\min}} = \left\{ x \in \mathbb{R}^n \mid \sum_{j=1}^n \underline{a}_{ij} x_j \leq \bar{b}_i, x_j \geq 0, j \in \mathcal{N} \right\}. \quad (58)$$

$$X_{\tilde{\leq}^{\max}} = \left\{ x \in \mathbb{R}^n \mid \sum_{j=1}^n \bar{a}_{ij} x_j \leq \underline{b}_i, x_j \geq 0, j \in \mathcal{N} \right\}. \quad (59)$$

$$X_{\tilde{\leq}^{T,S}} = X_{\tilde{\leq}_{T,S}} = \left\{ x \in \mathbb{R}^n \mid \sum_{j=1}^n \bar{a}_{ij} x_j \leq \bar{b}_i, x_j \geq 0, j \in \mathcal{N} \right\}. \quad (60)$$

$$X_{\tilde{\leq}^{S,T}} = X_{\tilde{\leq}_{S,T}} = \left\{ x \in \mathbb{R}^n \mid \sum_{j=1}^n \underline{a}_{ij} x_j \leq \underline{b}_i, x_j \geq 0, j \in \mathcal{N} \right\}. \quad (61)$$

Clearly, feasible solutions (58)–(61) are usual subsets of \mathbb{R}^n ; moreover, they all are polyhedral.

In order to find, e.g., a satisficing solution of ILP problem (57), we consider a fuzzy goal $\tilde{d} \in \mathcal{F}(\mathbb{R})$ and \tilde{R}_0 , a fuzzy extension of the usual binary relation \geq for comparing the objective with the fuzzy goal.

In the following proposition we show that if the feasible solution of ILP problem is classical then its max-satisficing solution is the same as the set of all classical optimal solutions of the linear programming problem of maximizing a particular non-fuzzy objective over the set of feasible solutions.

Proposition 39. *Let X be a classical feasible solution of ILP problem (57). Let $\tilde{d} \in \mathcal{F}(\mathbb{R})$ be a fuzzy goal with the membership function $\mu_{\tilde{d}}$ satisfying conditions (34). Let $G_A = G = T = \min$ and $S = \max$.*

- (i) If \tilde{R}_0 is $\tilde{\leq}^{\min}$, then the set of all max-satisficing solutions of ILP problem (57) coincides with the set of all optimal solution of the problem

$$\begin{aligned} & \text{maximize} && \sum_{j=1}^n \bar{c}_j x_j \\ & \text{subject to} && x \in X. \end{aligned}$$

- (ii) If \tilde{R}_0 is $\tilde{\leq}^{\max}$, then the set of all max-satisficing solutions of ILP problem (57) coincides with the set of all optimal solution of the problem

$$\begin{aligned} & \text{maximize} && \sum_{j=1}^n \underline{c}_j x_j \\ & \text{subject to} && x \in X. \end{aligned}$$

We close this section with several observations concerning duality of ILP problems.

Let the primal ILP problem (P) be problem (57) with \tilde{R} be $\tilde{\leq}^{\min}$; i.e., (38) holds. Then the dual ILP problem (D) is (39). Clearly, the feasible solution $X_{\tilde{\leq}^{\min}}$ of (P) is defined by (58) and the feasible solution $Y_{\tilde{\leq}^{\max}}$ of the dual problem (D) can be derived from (59) as

$$Y_{\tilde{\leq}^{\max}} = \left\{ y \in \mathbb{R}^m \mid \sum_{i=1}^m \underline{a}_{ij} y_i \geq \bar{c}_j, y_i \geq 0, i \in \mathcal{M} \right\}.$$

Notice that the problems

$$\begin{aligned} & \text{maximize} && \sum_{j=1}^n \bar{c}_j x_j \\ & \text{subject to} && x \in X_{\tilde{\leq}^{\min}} \end{aligned}$$

and

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^m \bar{b}_i y_i \\ & \text{subject to} && y \in Y_{\tilde{\leq}^{\max}} \end{aligned}$$

are dual to each other in the usual sense if and only if $\underline{c}_j = \bar{c}_j$ and $\underline{b}_i = \bar{b}_i$ for all $i \in \mathcal{M}$ and $j \in \mathcal{N}$.

31.8.3 FLP Problems with Centered Coefficients

Interesting class of FLP problems can be obtained if the coefficients of the FLP problem are fuzzy sets called \mathcal{B} -fuzzy intervals (see [39, 40]).

Definition 40. A fuzzy set A given by the membership function $\mu_A : \mathbb{R} \rightarrow [0, 1]$ is called a generator in \mathbb{R} if

- (i) $0 \in \text{Core}(A)$,
- (ii) μ_A is quasiconcave on \mathbb{R} .

Notice that each generator is a special fuzzy interval A that satisfies (i).

Definition 41. A set \mathcal{B} of generators in \mathbb{R} is called a basis of generators in \mathbb{R} if

- (i) $\chi_{\{0\}} \in \mathcal{B}, \chi_{\mathbb{R}} \in \mathcal{B}$,
- (ii) if $f, g \in \mathcal{B}$ then $\max\{f, g\} \in \mathcal{B}$ and $\min\{f, g\} \in \mathcal{B}$.

Definition 42. Let \mathcal{B} be a basis of generators. A fuzzy set A given by the membership function $\mu_A : \mathbb{R} \rightarrow [0, 1]$ is called a \mathcal{B} -fuzzy interval if there exists $a_A \in \mathbb{R}$ and $g_A \in \mathcal{B}$ such that for each $x \in \mathbb{R}$

$$\mu_A(x) = g_A(x - a_A).$$

The set of all \mathcal{B} -fuzzy intervals will be denoted by $\mathcal{F}_{\mathcal{B}}(\mathbb{R})$. Each $A \in \mathcal{F}_{\mathcal{B}}(\mathbb{R})$ is represented by a pair (a_A, g_A) ; we write $A = (a_A, g_A)$. An ordering relation $\leq_{\mathcal{B}}$ is defined on $\mathcal{F}_{\mathcal{B}}(\mathbb{R})$ as follows: For $A, B \in \mathcal{F}_{\mathcal{B}}(\mathbb{R})$, $A = (a_A, g_A)$, and $B = (a_B, g_B)$, we write $A \leq_{\mathcal{B}} B$ if and only if

$$(a_A < a_B) \text{ or } (a_A = a_B \text{ and } g_A \leq g_B). \quad (62)$$

Notice that $\leq_{\mathcal{B}}$ is a partial ordering on $\mathcal{F}_{\mathcal{B}}(\mathbb{R})$. The following proposition is a simple consequence of Definition 41.

Proposition 43. A pair (\mathcal{B}, \leq) , where \mathcal{B} is a basis of generators and \leq is the pointwise ordering of functions, is a lattice with the maximal element $\chi_{\mathbb{R}}$ and minimal element $\chi_{\{0\}}$.

Example 44. The following sets of functions form a basis of generators in \mathbb{R} :

- (i) $\mathcal{B}_D = \{\chi_{\{0\}}, \chi_{\mathbb{R}}\}$ – discrete basis,
- (ii) $\mathcal{B}_I = \{\chi_{[a,b]} \mid -\infty \leq a \leq 0 \leq b \leq +\infty\}$ – interval basis,
- (iii) $\mathcal{B}_G = \{\mu_d \mid \mu_d(x) = g^{(-1)}(|x|/d), x \in \mathbb{R}, d > 0\} \cup \{\chi_{\{0\}}, \chi_{\mathbb{R}}\}$, where $g : (0, 1] \rightarrow [0, +\infty)$ is non-increasing non-constant function, $g(1) = 0$, $g(0) = \lim_{x \rightarrow 0} g(x)$. Evidently, the relation \leq between function values is a linear ordering on \mathcal{B}_G .

Proposition 45. Let $\mathcal{F}_{\mathcal{B}_G}(\mathbb{R})$ be the set of all \mathcal{B}_G -fuzzy intervals, where \mathcal{B}_G is the basis from Example 44. Then the relation $\leq_{\mathcal{B}_G}$ is a linear ordering on $\mathcal{F}_{\mathcal{B}_G}(\mathbb{R})$.

We can extend this result as follows: Let \mathcal{B} be a basis of generators and $\leq_{\mathcal{B}}$ be a partial ordering on the set $\mathcal{F}_{\mathcal{B}}(\mathbb{R})$ defined by (62) in Definition 42. If \mathcal{B} is linearly ordered by \subseteq , then $\mathcal{F}_{\mathcal{B}}(\mathbb{R})$ is linearly ordered by $\leq_{\mathcal{B}}$. It follows that each $\tilde{c} \in \mathcal{F}_{\mathcal{B}}(\mathbb{R})$ can be uniquely represented by a pair (c, μ) , where $c \in \mathbb{R}$ and $\mu \in \mathcal{B}$ such that

$$\mu_{\tilde{c}}(t) = \mu(c - t);$$

therefore, we can write $\tilde{c} = (c, \mu)$.

Let \circ be either addition or multiplication – arithmetic operations on \mathbb{R} and \star be either min or max operations on \mathcal{B} . On $\mathcal{F}_{\mathcal{B}}(\mathbb{R})$ we introduce the following operations:

$$(a, f) \circ^{(\star)} (b, g) = (a \circ b, f \star g)$$

for all $(a, f), (b, g) \in \mathcal{F}_{\mathcal{B}}(\mathbb{R})$.

Evidently, the pairs of operations $(+^{(\min)}, \cdot^{(\min)})$, $(+^{(\max)}, \cdot^{(\max)})$, $(+^{(\max)}, \cdot^{(\min)})$, and $(+^{(\max)}, \cdot^{(\max)})$ are distributive. For more properties, see [23].

Now, consider \mathcal{B} -fuzzy intervals: $\tilde{c}_j = (c_j, f_j)$, $\tilde{a}_{ij} = (a_{ij}, g_{ij})$, $\tilde{b}_i = (b_i, h_i)$, $\tilde{c}_j, \tilde{a}_{ij}, \tilde{b}_i \in \mathcal{F}_{\mathcal{B}}(\mathbb{R})$, $i \in \mathcal{M}$, $j \in \mathcal{N}$. Let \diamond and \star be either min or max operations on \mathcal{B} . Consider the following optimization problem:

$$\begin{aligned} & \text{maximize} && \tilde{c}_1 \cdot^{(\diamond)} \tilde{x}_1 +^{(\star)} \cdots +^{(\star)} \tilde{c}_n \cdot^{(\diamond)} \tilde{x}_n \\ & \text{subject to} && \tilde{a}_{i1} \cdot^{(\diamond)} \tilde{x}_1 +^{(\star)} \cdots +^{(\star)} \tilde{a}_{in} \cdot^{(\diamond)} \tilde{x}_n \leq_{\mathcal{B}} \tilde{b}_i, \quad i \in \mathcal{M}, \\ & && \tilde{x}_j \geq_{\mathcal{B}} \tilde{0}, \quad j \in \mathcal{N}. \end{aligned} \quad (63)$$

In (63), maximization is performed with respect to the ordering $\leq_{\mathcal{B}}$; moreover, $\tilde{x}_j = (x_j, \xi_j)$, where $x_j \in \mathbb{R}$ and $\xi_j \in \mathcal{B}$, $\tilde{0} = (0, \chi_{\{0\}})$. The inequalities $\tilde{x}_j \geq_{\mathcal{B}} \tilde{0}$, $j \in \mathcal{N}$, are equivalent to $x_j \geq 0$, $j \in \mathcal{N}$. Now, we define feasible and optimal solutions.

A *feasible solution* of the problem (63) is a vector

$$(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n) \in \mathcal{F}_{\mathcal{B}}(\mathbb{R}) \times \mathcal{F}_{\mathcal{B}}(\mathbb{R}) \times \dots \times \mathcal{F}_{\mathcal{B}}(\mathbb{R}),$$

satisfying the constraints

$$\begin{aligned} \tilde{a}_{i1} \cdot^{(\diamond)} \tilde{x}_1 +^{(*)} \dots +^{(*)} \tilde{a}_{in} \cdot^{(\diamond)} \tilde{x}_n &\leq_{\mathcal{B}} \tilde{b}_i, \quad i \in \mathcal{M}, \\ \tilde{x}_j &\geq_{\mathcal{B}} \tilde{0}, \quad j \in \mathcal{N}. \end{aligned}$$

The set of all feasible solutions of (63) is denoted by $X_{\mathcal{B}}$.

An *optimal solution* of the problem (63) is a vector

$$(\tilde{x}_1^*, \tilde{x}_2^*, \dots, \tilde{x}_n^*)^T \in \mathcal{F}_{\mathcal{B}}(\mathbb{R}) \times \mathcal{F}_{\mathcal{B}}(\mathbb{R}) \times \dots \times \mathcal{F}_{\mathcal{B}}(\mathbb{R})$$

such that

$$\tilde{z}^* = \tilde{c}_1 \cdot^{(\diamond)} \tilde{x}_1^* +^{(*)} \dots +^{(*)} \tilde{c}_n \cdot^{(\diamond)} \tilde{x}_n^*$$

is the maximal element (with respect to the ordering $\leq_{\mathcal{B}}$) of the set

$$X_{\mathcal{B}}^* = \{\tilde{z} \mid \tilde{z} = \tilde{c}_1 \cdot^{(\diamond)} \tilde{x}_1 +^{(*)} \dots +^{(*)} \tilde{c}_n \cdot^{(\diamond)} \tilde{x}_n, (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)^T \in X_{\mathcal{B}}\}.$$

For each of four combinations of min and max in the operations $\cdot^{(\diamond)}$ and $+^{(*)}$, (63) is a particular optimization problem. We can easily derive the following result.

Proposition 46. Let \mathcal{B} be a linearly ordered basis of generators. Let $(\tilde{x}_1^*, \tilde{x}_2^*, \dots, \tilde{x}_n^*)^T \in \mathcal{F}_{\mathcal{B}}(\mathbb{R})^n$ be an optimal solution of (63), where $\tilde{x}_j^* = (x_j^*, \xi_j^*)$, $j \in \mathcal{N}$. Then the vector $x^* = (x_1^*, \dots, x_n^*)$ is an optimal solution of the following linear programming problem:

$$\begin{aligned} &\text{maximize} && c_1 x_1 + \dots + c_n x_n \\ &\text{subject to} && a_{i1} x_1 + \dots + a_{in} x_n \leq b_i, \quad i \in \mathcal{M}, \\ & && x_j \geq 0, \quad j \in \mathcal{N}. \end{aligned} \tag{64}$$

Now, by A_x we denote the set of indices of all active constraints of (64) at $x = (x_1, \dots, x_n)$; i.e.,

$$A_x = \{i \in \mathcal{M} \mid a_{i1} x_1 + \dots + a_{in} x_n = b_i\}.$$

The following proposition gives a necessary condition for the existence of a feasible solution of (63). The proof can be found in [40].

Proposition 47. Let \mathcal{B} be a linearly ordered basis of generators. Let $(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)^T \in \mathcal{F}_{\mathcal{B}}(\mathbb{R})^n$ be a feasible solution of (63), where $\tilde{x}_j = (x_j, \xi_j)$, $j \in \mathcal{N}$. Then the vector $x = (x_1, \dots, x_n)^T$ is the feasible solution of the linear programming problem (64) and it holds

(i) if $\diamond = \max$ and $\star = \min$, then

$$\min\{a_{ij} \mid j \in \mathcal{N}\} \leq_{\mathcal{B}} b_i \quad \text{for all } i \in A_x;$$

(ii) if $\diamond = \max$ and $\star = \max$, then

$$\max\{a_{ij} \mid j \in \mathcal{N}\} \leq_{\mathcal{B}} b_i \quad \text{for all } i \in A_x.$$

Notice that in this section we have presented an alternative approach to FLP problems. Comparing to the approach presented before, the decision variables x_j considered here have not been non-negative numbers. They have been considered as fuzzy intervals of the same type as the corresponding coefficients of the FLP problem. From the computational point of view this approach is simple as it requires to solve only a classical linear programming problem.

31.9 Illustrating Example

Let us study the following problem, see [15]. An investor has a sum of USD 12 million in the beginning of a monitored term and decides about a participation in two investment projects. The length of both projects are 3 years. Leftover resources in every particular year can be put on time deposit. Returns and costs considered are uncertain and can be formulated as fuzzy numbers. The problem is to find a (non-fuzzy) strategy maximizing quantity of resources at the end of the 3-year term. This optimal investment problem can be formulated by the following FLP model:

$$\begin{aligned} \text{maximize} \quad & \tilde{c}_1 x_1 + \tilde{c}_2 x_2 + (1 + \tilde{u}_3) p_3 \\ \text{subject to} \quad & \begin{aligned} \tilde{a}_{11} x_1 + \tilde{a}_{12} x_2 &+ p_1 & \cong 12, \\ \tilde{a}_{21} x_1 + \tilde{a}_{22} x_2 &+ (1 + \tilde{u}_1) p_1 & \cong 0, \\ \tilde{a}_{31} x_1 + \tilde{a}_{32} x_2 &+ (1 + \tilde{u}_2) p_2 - p_3 & \cong 0, \\ x_1, x_2 & & \leq 1, \\ x_1, x_2, p_1, p_2, p_3 & & \geq 0, \end{aligned} \end{aligned} \quad (65)$$

where we denote

\tilde{c}_i – fuzzy return of the i th project, $i = 1, 2$, at the end of the period;

\tilde{a}_{ij} – fuzzy return/cost of the i th project $i = 1, 2$, j th year, $j = 1, 2, 3$;

\tilde{u}_j – fuzzy interest rate in the j th year, $j = 1, 2, 3$;

x_i – participation measure in the i th project, $i = 1, 2$;

p_j – resource allocation in the j th year, $j = 1, 2, 3$;

\cong – fuzzy equality relation.

Let $\tilde{a} = (a_L, a_C, a_R)$ be a *triangular fuzzy number*, $a_L < a_C < a_R$, where a_L is called the *left value* of \tilde{a} , a_C is called the *central value*, and a_R is the *right value* of \tilde{a} . Then the membership function of \tilde{a} is given by

$$\mu_{\tilde{a}}(t) = \max\{0, \min\{\frac{t - a_L}{a_C - a_L}, \frac{a_R - t}{a_R - a_C}\}\}.$$

If $a_L = a_C = a_R$, we say that $\tilde{a} = (a_L; a_C; a_R)$ is *non-fuzzy* (i.e., ordinary real number), with the membership function identical to the characteristic function χ_{a_C} .

In our problem, the parameters $\tilde{c}_1, \tilde{c}_2, \tilde{a}_{11}, \tilde{a}_{12}, \tilde{a}_{21}, \tilde{a}_{22}, \tilde{a}_{31}, \tilde{a}_{32}, \tilde{u}_1, \tilde{u}_2, \tilde{u}_3$ are supposed to be triangular fuzzy numbers as follows:

$$\begin{aligned} \tilde{c}_1 &= (4, 6, 8) & \tilde{a}_{21} &= (-4, -2, 0) & \tilde{u}_1 &= (0.01, 0.02, 0.03) \\ \tilde{c}_2 &= (3, 5, 7) & \tilde{a}_{22} &= (1, 2, 3) & \tilde{u}_2 &= (0.01, 0.02, 0.03) \\ \tilde{a}_{11} &= (6, 10, 14) & \tilde{a}_{31} &= (6, 8, 10) & \tilde{u}_3 &= (0.01, 0.03, 0.05) \\ \tilde{a}_{12} &= (3, 6, 9) & \tilde{a}_{32} &= (6, 12, 18). \end{aligned}$$

Let $x_1, x_2 \geq 0$, except $x_1 = x_2 = 0$; i.e., we exclude the situation that the investor will participate in no project.

(a) *Membership functions.* By Extension principle, the left-hand sides of the three constraints in (65), denoted by $\tilde{L}_1, \tilde{L}_2, \tilde{L}_3$ are triangular fuzzy numbers as follows:

$$\begin{aligned}\tilde{L}_1 &= (6x_1 + 3x_2 + p_1, 10x_1 + 6x_2 + p_1, 14x_1 + 9x_2 + p_1), \\ \tilde{L}_2 &= (-4x_1 + x_2 + 1.01p_1 - p_2, -2x_1 + 2x_2 + 1.02p_1 - p_2, \\ &\quad 3x_2 + 1.03p_1 - p_2), \\ \tilde{L}_3 &= (6x_1 + 6x_2 + 1.01p_2 - p_3, 8x_1 + 12x_2 + 1.02p_2 - p_3, \\ &\quad 10x_1 + 18x_2 + 1.03p_2 - p_3).\end{aligned}$$

Applying (9), we calculate the membership functions of $\tilde{L}_1, \tilde{L}_2, \tilde{L}_3$:

$$\begin{aligned}\mu_{\tilde{L}_1}(t) &= \max \left\{ 0, \min \left\{ \frac{t - 6x_1 - 3x_2 - p_1}{4x_1 + 3x_2}, \frac{14x_1 + 9x_2 + p_1 - t}{4x_1 + 3x_2} \right\} \right\}, \\ \mu_{\tilde{L}_2}(t) &= \max \left\{ 0, \min \left\{ \frac{t + 4x_1 - x_2 - 1.01p_1 + p_2}{2x_1 + x_2 + 0.01p_1}, \frac{3x_2 + 1.03p_1 - p_2 - t}{2x_1 + x_2 + 0.01p_1} \right\} \right\}, \\ \mu_{\tilde{L}_3}(t) &= \max \left\{ \min \left\{ 0, \frac{t - 6x_1 - 6x_2 - 1.01p_2 + p_3}{2x_1 + 6x_2 + 0.01p_2}, \frac{10x_1 + 18x_2 + 1.03p_2 - p_3 - t}{2x_1 + 6x_2 + 0.01p_2} \right\} \right\}.\end{aligned}$$

Now, we calculate the membership function μ_{\cong} of the fuzzy relation \cong being a fuzzy extension of the valued relation '=':

$$\mu_{\cong}(\tilde{L}_i, \tilde{P}_i) = \sup \{ \min \{ 0, \mu_{\tilde{L}}(u), \mu_{\tilde{P}_i}(v) \} \mid u = v \}, i = 1, 2, 3,$$

where \tilde{P}_i are real numbers with the characteristic functions

$$\begin{aligned}\mu_{\tilde{P}_1}(t) &= \begin{cases} 1 & \text{if } t = 12 \\ 0 & \text{otherwise} \end{cases}, & \mu_{\tilde{P}_2}(t) &= \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases}, \\ \mu_{\tilde{P}_3}(t) &= \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases}.\end{aligned}$$

Particularly,

$$\mu_{\cong}(\tilde{L}_1, \tilde{P}_1) = \mu_{\tilde{L}_1}(12), \mu_{\cong}(\tilde{L}_i, \tilde{P}_i) = \mu_{\tilde{L}_i}(0), i = 2, 3.$$

Notice that for real numbers the fuzzy relation \cong is identical to the ordinary equality relation '='.

(b) *Feasible solution.*

By Definition 20, using $T^A = T = \min$, the feasible solution of FLP problem (65) is a fuzzy set \tilde{X} defined by the membership function:

$$\mu_{\tilde{X}}(x_1, x_2, p_1, p_2, p_3) = \min \{ \mu_{\tilde{L}_1}(12), \mu_{\tilde{L}_2}(0), \mu_{\tilde{L}_3}(0) \}.$$

For $\alpha \in (0, 1]$, α -feasible solution is the set of all vectors

$$x = (x_1, x_2, p_1, p_2, p_3)$$

such that

$$\min \{ \mu_{\tilde{L}_1}(12), \mu_{\tilde{L}_2}(0), \mu_{\tilde{L}_3}(0) \} \geq \alpha. \quad (66)$$

Inequality (66) can be expressed equivalently by the following inequalities:

$$\begin{aligned}
 (6 + 4\alpha)x_1 + (3 + 3\alpha)x_2 + p_1 &\leq 12, \\
 (14 - 4\alpha)x_1 + (9 - 3\alpha)x_2 + p_1 &\geq 12, \\
 (4 - 2\alpha)x_1 - (1 + \alpha)x_2 - (1.01 + 0.01\alpha)p_1 + p_2 &\geq 0, \\
 - 2\alpha x_1 + (3 - \alpha)x_2 + (1.03 - 0.01\alpha)p_1 - p_2 &\geq 0, \\
 (6 + 2\alpha)x_1 + (6 + 6\alpha)x_2 + (1.01 + 0.01\alpha)p_2 - p_3 &\leq 0, \\
 (10 - 2\alpha)x_1 + (18 - 6\alpha)x_2 + (1.03 - 0.01\alpha)p_2 - p_3 &\geq 0, \\
 x_1, x_2 &\leq 1, \\
 x_1, x_2, p_1, p_2, p_3 &\geq 0.
 \end{aligned} \tag{67}$$

(c) *Satisficing solution.*

We choose the appropriate fuzzy goal \tilde{d} setting two values of the objective function: the lower limit 21 and upper limits 27 (see Section 5.1). The lower limit corresponds to the highest value with the membership grade of the goal equal to zero, and upper limit is the lowest value where the membership grade is equal to 1. The resulting membership function of the goal \tilde{d} is then non-decreasing and piecewise linear; i.e., \tilde{d} is given by the membership function: $\mu_{\tilde{d}}(t) = \min\{1, \max\{0, \frac{t-21}{6}\}\}$ for all $t \geq 0$.

For membership function of objective \tilde{Z} we have

$$\mu_{\tilde{Z}}(t) = \max\{0, \min\{\frac{t - 4x_1 - 3x_2 - 1.01p_3}{2x_1 + 2x_2 + 0.02p_3}, \frac{8x_1 + 7x_2 + 1.05p_3 - t}{2x_1 + 2x_2 + 0.02p_3}\}\}.$$

Now, we calculate the membership function $\mu_{\tilde{\geq}}$ of the fuzzy relation $\tilde{\geq}$ being a fuzzy extension of the valued relation ' \geq ':

$$\mu_{\tilde{\geq}}(\tilde{Z}, \tilde{d}) = \sup\{\min\{0, \mu_{\tilde{Z}}(u), \mu_{\tilde{d}}(v)\} \mid u \geq v\}.$$

For the membership function of the objective function we obtain

$$\mu_{\tilde{\geq}}(\tilde{Z}, \tilde{d}) = \max\{0, \min\{\frac{8x_1 + 7x_2 + 1.05p_3 - 21}{2x_1 + 2x_2 + 0.02p_3 + 6}, 1\}\}.$$

For optimal solution \tilde{X}_0 it follows

$$\mu_{\tilde{X}_0}(x) = \min\{\mu_{\tilde{X}}(x_1, x_2, p_1, p_2, p_3), \mu_{\tilde{\geq}}(\tilde{Z}, \tilde{d})\}.$$

For $\alpha \in (0, 1]$, the α -satisficing solution is a set of all vectors $x^0 = (x_1, x_2, p_1, p_2, p_3)$, such that $\mu_{\tilde{X}_0}(x) \geq \alpha$, or, $\mu_{\tilde{X}}(x_1, x_2, p_1, p_2, p_3) \geq \alpha$, and at the same time $\mu_{\tilde{\geq}}(\tilde{Z}, \tilde{d}) \geq \alpha$. The former inequality is equivalent to inequalities (67) and the latter is equivalent to

$$(8 - 2\alpha)x_1 + (7 - 2\alpha)x_2 + (1.05 - 0.02\alpha)p_3 \geq 21 + 6\alpha. \tag{68}$$

Hence, the set of all α -satisficing solutions is a set of all vectors $x^0 = (x_1, x_2, p_1, p_2, p_3)$ satisfying (67) and (68).

In order to find a max-satisficing solution of FLP (65), we apply Proposition 27 by solving the following nonlinear programming problem:

$$\begin{aligned}
 &\text{maximize} && \alpha \\
 &\text{subject to} && (67), (68), \\
 & && 0 \leq x_1, x_2, \alpha \leq 1, \\
 & && p_1, p_2, p_3 \geq 0.
 \end{aligned}$$

By using Excel Solver, we have calculated the following optimal solution:

$x_1 = 0.605$, $x_2 = 1$, $p_1 = 0$, $p_2 = 0.811$, $p_3 = 17.741$, and $\alpha = 0.990$.

For the classical problem, i.e., the usual linear programming problem (65), where parameters \tilde{c}_i , \tilde{a}_{ij} , \tilde{u}_j are real numbers equal to the central values, we obtain the following optimal solution:

$$x_1 = 0.6, x_2 = 1, p_1 = 0, p_2 = 0.8, p_3 = 17.611, \text{ and } z = 26.744.$$

Both solutions are close to each other, which is natural, as the central values of the parameters are applied.

On the other hand we could ask for α -satisficing solution with $\alpha < 1$, e.g., $\alpha = 0.7$, i.e., with a lower level of satisfaction. We have found such solution with the additional property that p_3 is maximized:

$$x_1 = 0.62, x_2 = 1, p_1 = 0, p_2 = 0.811, \alpha = 0.7, \text{ and } p_3 = 17.744.$$

Hence, the fuzzy linear programming problem formulation allows for finding different kinds of ‘optimal’ solutions in environment with uncertain parameters of the model and also enables to take into account additional requirements.

31.10 Conclusion

In this overview chapter devoted to FLP we have proposed a general approach to FLP problems with fuzzy coefficients. A unifying concept of this approach is the concept of fuzzy relation, particularly fuzzy extension of the inequality or equality relation and the concept of aggregation operator. We have formulated FLP problem, defined as a feasible solution of FLP problem and dealt with the problem of ‘optimal solution’ of FLP problems. Two approaches have been introduced: the satisficing solution based on external goals modeled by fuzzy quantities and α -efficient (non-dominated) solution. Then our interest has been focused on the problem of duality in FLP. The chapter has been closed with an illustrating numerical example.

Acknowledgment

This research was partly supported by the Czech Grant Agency, Grant no. 402/06/0431.

References

- [1] R. Bellman and L. Zadeh. Decision making in fuzzy environment. *Manage. Sci.* 17 (1970) 141–164.
- [2] M. Inuiguchi and J. Ramík. Possibilistic linear programming: A brief review of fuzzy mathematical programming and a comparison with stochastic programming in portfolio selection problem. *Fuzzy Sets Syst.* 111 (2000) 3–28.
- [3] J. Ramík and J. Římánek. Inequality relation between fuzzy numbers and its use in fuzzy optimization. *Fuzzy Sets Syst.* 16 (1985) 123–138.
- [4] J. Ramík. Extension principle in fuzzy optimization. *Fuzzy Sets Syst.* 19 (1986) 29–37.
- [5] J. Ramík. An application of fuzzy optimization to optimum allocation of production. In J. Kacprzyk and S.A. Orlovsky (eds), *Proceedings of International*, Academia-Verlag, Laxenburg, Berlin, 1987, IIASA, pp. 227–241.
- [6] J. Ramík. A unified approach to fuzzy optimization. In: M. Sugeno (ed.), *Proceedings of the 2nd IFSA Congress*, Tokyo, 1987, IFSA, pp. 128–130.
- [7] J. Ramík and J. Římánek. The linear programming problem with vaguely formulated relations between the coefficients. In: M. Fedrizzi, J. Kacprzyk, and S.A. Orlovsky (eds), *Interfaces between Artificial Intelligence and Operations Research in Fuzzy Environment*. D. Ricdel Publishing Company, Dordrecht Boston Lancaster Tokyo, 1989, pp. 104–119.
- [8] J. Ramík. Fuzzy preferences in linear programming. In: M. Fedrizzi and J. Kacprzyk (eds), *Interactive Fuzzy Optimization and Mathematical Programming*. Springer-Verlag, Berlin Heidelberg New York, 1990, pp. 114–122.
- [9] J. Ramík. Inequality relations between fuzzy data. In: H. Bandemer (ed.), *Modelling Uncertain Data*, Akademie Verlag, Berlin, 1992, pp. 158–162.
- [10] J. Ramík. Some problems of linear programming with fuzzy coefficients. In: K.-W. Hansmann, A. Bachem, M. Jarke, and A. Marusev, (eds), *Operation Research Proceedings: Papers of the 21st Annual Meeting of DGOR 1992*, Springer-Verlag, Heidelberg, 1993, pp. 296–305.
- [11] J. Ramík and K. Nakamura. Canonical fuzzy numbers of dimension two. *Fuzzy Sets Syst.* 54 (1993) 167–180.

- [12] J. Ramík, K. Nakamura, I. Rozenberg, and I. Miyakawa. Joint canonical fuzzy numbers. *Fuzzy Sets Syst.* 53 (1993) 29–47.
- [13] J. Ramík and H. Rommelfanger. A single- and multi-valued order on fuzzy numbers and its use in linear programming with fuzzy coefficients. *Fuzzy Sets Syst.* 57 (1993) 203–208.
- [14] J. Ramík and H. Rommelfanger. Fuzzy mathematical programming based on some new inequality relations. *Fuzzy Sets Syst.* 81 (1996) 77–88.
- [15] M. Fiedler, J. Nedoma, J. Ramík, J. Rohn, and K. Zimmermann. *Linear Optimization Problems with Inexact Data*. Springer Science + Business Media, New York, 2006.
- [16] J. Ramík. Duality in fuzzy linear programming with possibility and necessity relations. *Fuzzy Sets Syst.* 157(1) (2006) 1283–1302.
- [17] J.J. Buckley. Possibilistic linear programming with triangular fuzzy numbers. *Fuzzy Sets Syst.* 26 (1988) 135–138.
- [18] S. Chanas. Fuzzy programming in multiobjective linear programming – a parametric approach. *Fuzzy Sets Syst.* 29 (1989) 303–313.
- [19] S. Chen and C. Hwang. *Fuzzy Multiple Attribute Decision Making*. Springer-Verlag, Berlin, Heidelberg, New York, 1992.
- [20] M. Delgado, J. Kacprzyk, J.-L. Verdegay, and M.A. Vila. *Fuzzy Optimization – Recent Advances*. Physica-Verlag, Heidelberg, New York, 1994.
- [21] M. Inuiguchi, H. Ichihashi, and Y. Kume. Modality constrained programming problems: A unified approach to fuzzy mathematical programming problems in the setting of possibility theory. *Inf. Sci.* 67 (1993) 93–126.
- [22] M. Inuiguchi and T. Tanino. Scenario decomposition approach to interactive fuzzy numbers in possibilistic linear programming problems. In: R. Felix (ed), Proceedings of EFDAN'99, Dortmund, 2000, FLS Fuzzy Logic Systeme GmbH, pp. 133–142.
- [23] M. Kovacs and L.H. Tran. Algebraic structure of centered M-fuzzy numbers. *Fuzzy Sets. Syst.* 39 (1991) 91–99.
- [24] Y.J. Lai and C.L. Hwang. *Fuzzy Mathematical Programming: Theory and Applications*. Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1992.
- [25] Y.J. Lai and C.L. Hwang. *Multi-Objective Fuzzy Mathematical Programming: Theory and Applications*. Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1993.
- [26] S.A. Orlovsky. Decision making with fuzzy preference relation. *Fuzzy Sets Syst.* 1 (1978) 155–167.
- [27] S.A. Orlovsky. On formalization of a general fuzzy mathematical programming problem. *Fuzzy Sets Syst.* 3 (1980) 311–321.
- [28] J. Ramík and H. Rommelfanger. A new algorithm for solving multi-objective fuzzy linear programming problems. *Found. Comput. Decis. Sci.* 3 (1996) 145–157.
- [29] H. Rommelfanger. *Entscheiden bei Unschärfe – Fuzzy Decision Support Systeme*. Springer-Verlag, Berlin Heidelberg, 1988.
- [30] H. Rommelfanger and R. Slowinski. Fuzzy linear programming with single or multiple objective functions. In: R. Slowinski (ed), *Fuzzy Sets in Decision Analysis, Operations Research and Statistics*, Kluwer Academic Publishers, Boston Dordrecht London, 1998, pp. 179–213.
- [31] M. Sakawa and H. Yano. Interactive decision making for multiobjective programming problems with fuzzy parameters. In: R. Slowinski and J. Teghem (eds), *Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming Under Uncertainty*. Kluwer Academic Publishers, Dordrecht, 1990, pp. 191–220.
- [32] B. Werners. Interactive fuzzy programming system. *Fuzzy Sets Syst.* 23 (1987) 131–147.
- [33] R.R. Yager. On a general class of fuzzy connectives. *Fuzzy Sets Syst.* 4 (1980) 235–242.
- [34] H.-J. Zimmermann. Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets Syst.* 1 (1978) 45–55.
- [35] J. Ramík and M. Vlach. *Generalized Concavity as a Basis for Optimization and Decision Analysis*. Technical Report IS-RR-2001-003. Japan Advanced Institute for Science and Technology, Hokuriku, March 2001.
- [36] J. Ramík and M. Vlach. A non-controversial definition of fuzzy sets. In: J.F. Peters, A. Skowron (eds), *Transactions on rough sets II – Rough sets and fuzzy sets*. Springer-Verlag, Berlin Heidelberg, , 2004, pp. 201–207.
- [37] E.P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*. Kluwer Academic Publishers Series Trends in Logic. Dordrecht Boston London, 2000.
- [38] M. Inuiguchi, H. Ichihashi, and Y. Kume. Some properties of extended fuzzy preference relations using modalities. *Inf. Sci.* 61 (1992) 187–209.
- [39] J. Ramík and M. Vlach. *Generalized Concavity in Optimization and Decision Making*. Kluwer Academic Publishers, Boston Dordrecht London, 2001.
- [40] M. Kovacs. Fuzzy linear programming with centered fuzzy numbers. In: M. Delgado, J. Kacprzyk, J.-L. Verdegay, and M.A. Vila (eds), *Fuzzy Optimization – Recent Advances*. Physica-Verlag, Heidelberg New York, 1994, pp. 135–147.

32

A Fuzzy Regression Approach to Acquisition of Linguistic Rules

Junzo Watada and Witold Pedrycz

32.1 Introduction

As stressed by Zadeh (cf. [1, 2]) in computing with words being cast in the setting of granular computing, fuzzy sets play a pivotal role. ‘The essence of granular computing is to carry out computing that exploits information granules [1–3]. Information granules are regarded as collections of elements that can be perceived and treated together because of their similarity, functional properties, or spatial or temporal adjacency’ [4, 5]. The most common tool for a human to granule knowledge is a natural language in which words are reflective of the individual information granules. In this sense, fuzzy logic becomes instrumental as an effective vehicle to manipulate information granules. Typically, mechanisms of approximate (fuzzy) reasoning are involved here. In this study, we exploit an idea of fuzzy regression along with its optimization capabilities as a vehicle to complete computing with words. More specifically, we intend to abstract the latent structure about the underlying relations between words. Human words can be translated (formalized) into fuzzy sets (fuzzy numbers, to be more specific) which are afterward employed in a fuzzy reasoning scheme. Considering that it is possible (through the mechanism of approximate reasoning) to build a dictionary expressing relationships between fuzzy numbers and words, we can construct the relations existing within the data making use of fuzzy regression analysis [4, 6–8].

It becomes apparent that experts with much professional experiences are capable of making assessment using their intuition and experience. Given this intuitive nature of domain knowledge, the measurements and interpretation of these characteristics inherently involve uncertainty. In such cases, judgments may be expressed by experts using linguistic terms. The difficulty in the direct measurement of certain characteristics makes their estimation highly imprecise and this situation implies the use of fuzzy sets (cf. [9–12]). There have been a number of well-documented cases in which fuzzy regression analysis has been effectively used. One may refer here to Watada *et al.* [13, 14] and Toyoura *et al.* [15] who proposed a model of damage assessment based on information given by experts and being processed through fuzzy multivariate analysis.

To cope with linguistic variables, we define processes of vocabulary translation and vocabulary matching which convert linguistic expressions into membership functions defined in the unit interval. Fuzzy regression analysis [9, 16] is employed to deal with the mapping and assessment process [15, 17, 18] of experts which are realized from linguistic variables of features and characteristics of an objective into the linguistic expression articulating the total assessment.

Throughout this study, we adhere to the standard notation being commonly used in fuzzy sets. In particular, triangular fuzzy numbers are represented as a triple (m, a, b) , where ‘ m ’ denotes a central value and a and b left and right widths (spreads), respectively. This topic is described in Section 32.2, which highlights the underlying concept along with the formulation of the model. Section 32.3 discusses how to obtain a fuzzy regression model for given fuzzy values. Section 32.4 provides an illustrative example. Finally, Section 32.5 concludes the chapter by stressing the characteristics of the model.

32.2 Linguistic Variables and Vocabulary Matching

In making assessments regarding some objects, experts often (a) linguistically evaluate various features and characteristics of the object and (b) carry out an overall assessment of the object totally in a linguistic form. For instance, although it is possible to measure production volume, it is difficult to analytically interpret the obtained numerical value in terms of possible influence. This result might have impacted on future decision making.

Let us consider cases when evaluating some network of sales points, as in the case of a car sale network. Let us consider that we have some evaluations of their generic features such as sales volume, number of customers visiting the sale point, and alike. Furthermore, a total assessment of such sales points is given. The example presented in Table 32.1 illustrates a situation in which individual features are evaluated using linguistic terms, say, a *good* sales trend, etc. To enhance readability, the linguistic terms are emphasized using italics. An overall assessment (concerning the state of a system) could also be articulated in a linguistic manner (say, *very good*, *good*, *average*, *bad*, and alike).

As before, let us consider the sales volume and a number of customers visiting the sales network. Let it be a linguistic variable which assumes values in some set. For instance, we could have

$$\begin{aligned} L_{\text{volume}}(\text{sales volume}) &\text{ is } \textit{extremely bad}, \\ L_{\text{number}}(\text{visiting customers}) &\text{ is } \textit{low}, \end{aligned}$$

where subscripts (number) in L (say, visiting customers) denote the state (variable) of a visiting customers.

The expressions used in Table 32.1, such as, e.g., ‘*good*’, ‘*bad*’, ‘*extremely bad*’ can be defined with fuzzy grades on $[0, 1]$ such as $U_{(\text{good})}$, $U_{(\text{bad})}$, and $U_{(\text{extremely bad})}$. The sales volume can be defined using fuzzy numbers. Denote by ϕ a certain possibility distribution. We can identify the possibility of the state of a sales trend with the degree of its descriptive adjective on $[0, 1]$. For example,

$$\begin{aligned} \pi_{(\text{number})}(\text{visiting customer}) &\equiv \pi_{(\text{state})}(\text{the number}) \\ &\equiv U_{(\text{extremely bad})}. \end{aligned}$$

Table 32.1 Linguistic data given by experts

Training sample	Linguistic variables					Linguistic objective
1	$L_1(1)$...	$L_i(1)$...	$L_K(1)$	$L_0(1)$
2	$L_1(2)$...	$L_i(2)$...	$L_K(2)$	$L_0(2)$
3	‘Good’	...	‘Bad’	...	‘Very bad’	‘Bad’
:	:		:		:	:
ω	$L_1(\omega)$...	$L_i(\omega)$...	$L_K(\omega)$	$L_0(\omega)$
:	:		:		:	:
n	$L_1(n)$...	$L_i(n)$...	$L_K(n)$	$L_0(n)$

While the number of customers is quantified linguistically as $\phi_{(number)}(\text{visiting customers})$, this expresses the state of the visiting number of customers. We can also use the notation $\phi_{(state)}(number)$. Such an evaluation as good, bad, or extremely bad is given based on some approximate value that is U(*extremely bad*).

Let us define the dictionary of such adjectives as *good*, *bad*, and *extremely bad*. They are named descriptive adjective by stressing the usage of evaluation. According to the above explanation, a descriptive adjective should correspond with some fuzzy grade on [0, 1]. For example, in case when the number of customers is '*good*' it means the number of customers per day might be 5000 to 10,000 but in case of '*extremely bad*' it could simply mean the number is under 1000.

We intend to model the experts' assessment process through which experts evaluate the possibility of sales trend on the basis of states of the features of the process and the corresponding characteristics L_i , $i = 1, 2, \dots, K$, where these values L_i are a linguistic term for object i . In other words, we intend to determine the linguistic assessment process F of the linguistic variables L_1, L_2, \dots, L_K , which gives rise to a linguistic value of an objective Z . More formally, we express such mapping in the following manner:

$$\mathbf{L} = \mathbf{F}(\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_K). \quad (1)$$

We also define the descriptive adjectives '*extreme*', '*very*' being used in this description. Let us form a dictionary of the corresponding linguistic expressions L_i and fuzzy grades U_{L_i} . Let us also consider fuzzy grades in the form of a triangular-shaped membership function of a fuzzy number.

We obtain the assessment L_i ($i = 1, 2, \dots, K$), where L_i is understood in terms of U_{L_i} through the use of the linguistic dictionary built by experts.

The linguistic total assessment L_T is derived using linguistic values of the corresponding features, L_i $i = 1, 2, \dots, K$. This assessment is denoted as

$$\mathbf{L}_T = \mathbf{g}(\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_K). \quad (2)$$

The structure of the linguistic assessment has to relate to the numeric evaluations. Subsequently, let us denote this linguistic assessment f as follows:

$$\mathbf{V} = \mathbf{f}(\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_K), \quad (3)$$

where V stands for a numeric evaluation and $U_i, i = 1, 2, \dots, K$. When this numerical nature of the evaluation of a linguistic value is stressed, we will be using pertinent notations such as $U_{L_i}, i = 1, 2, \dots, K$. In other words, a human translates some vague numerical assessment U_i into a linguistic value L_i when he/she offers a quantification, such as '*good*', '*bad*', '*extremely bad*', etc.

The mechanism of human evaluation F can be structured into the following three phases:

1. Translation of attributes from linguistic values L_i into fuzzy grades U_L making use of triangular membership functions

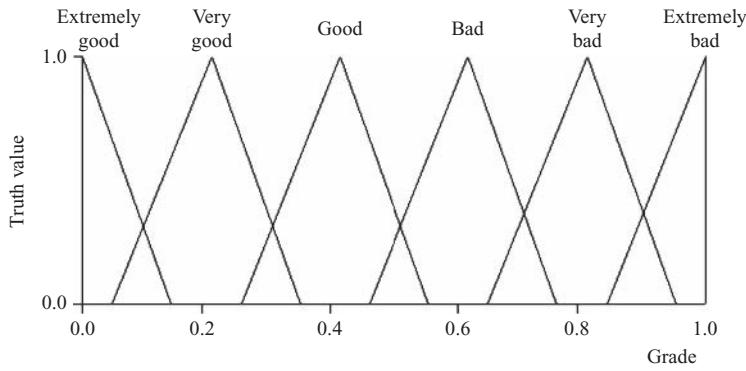
$$\mathbf{U}_{\mathbf{L}_i} \equiv (u_i, c_i^l, c_i^r),$$

where u_i denotes the central (modal) value of the fuzzy set (fuzzy number), c_i^l stands for a left-side bound (lower bound), and c_i^r denotes a right-side bound (upper bound), respectively.

2. Estimation of the total assessment by the fuzzy assessment function

$$\mathbf{V} = f(\mathbf{U}_{\mathbf{L}_1}, \mathbf{U}_{\mathbf{L}_2}, \dots, \mathbf{U}_{\mathbf{L}_K}), \quad (4)$$

which produces a fuzzy grade \mathbf{V} by mapping fuzzy grades of attributes $\mathbf{U}_{\mathbf{L}_i}$, where a suffix \mathbf{L} is not attached to \mathbf{V} because the linguistic expression is unknown. The detailed calculations will be clarified in Section 32.3.



$$\begin{aligned}
 U_{(\text{extremely good})} &= (0.00, 0.00, 0.15) \\
 U_{(\text{very good})} &= (0.20, 0.15, 0.15) \\
 U_{(\text{good})} &= (0.40, 0.15, 0.15) \\
 U_{(\text{bad})} &= (0.40, 0.15, 0.15) \\
 U_{(\text{very bad})} &= (0.80, 0.15, 0.15) \\
 U_{(\text{extremely bad})} &= (0.00, 0.15, 0.15)
 \end{aligned}$$

Figure 32.1 Dictionary of descriptive adjectives

3. Linguistic matching of the fuzzy grade of the objective with the elements of the dictionary in Figure 32.1 wherein the linguistic value \mathbf{Z} is decided for the objective.

Let us define the mechanism of vocabulary matching by expressing it as the following minimax problem:

$$\mathbf{Z}_0 \simeq \max_{\mathbf{W}_i \in \mathbf{D}} \left\{ \max_t \mu_V(t) \wedge \mu_{W_i}(t) \right\}, \quad (5)$$

where $\mathbf{Z}_0 \simeq \max_{\mathbf{W}_i \in \mathbf{D}} f(\mathbf{W}_i)$ denotes that \mathbf{Z}_0 is the word in \mathbf{D} which realizes the maximum value of f , μ_V denotes a membership function of V , and μ_{W_i} denotes a membership function of a word \mathbf{W}_i as included in the dictionary \mathbf{D} . The essence of this procedure is illustrated in Figure 32.2. We assign a word \mathbf{L}_0 ‘very bad’ to the fuzzy grade of the total assessment V showed by a dotted line in Figure 32.2.

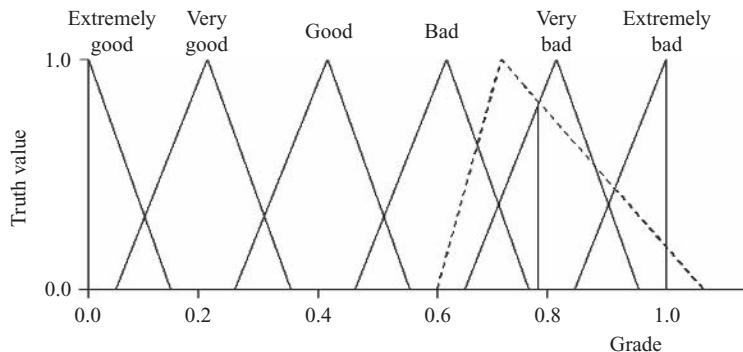


Figure 32.2 Vocabulary matching

Table 32.2 Translated data fuzzy

Training sample	Fuzzy grade of variables			Fuzzy grade of the objective	
1	$\mathbf{U}_{L_1}(1)$...	$\mathbf{U}_{L_i}(1)$...	$\mathbf{U}_{L_K}(1)$
2	$\mathbf{U}_{L_1}(2)$...	$\mathbf{U}_{L_i}(2)$...	$\mathbf{U}_{L_K}(2)$
3	$(0.4, 0.15, 0.15)$...	$(0.6, 0.15, 0.15)$...	$(0.8, 0.15, 0.15)$
⋮	⋮		⋮		⋮
ω	$\mathbf{U}_{L_1}(\omega)$...	$\mathbf{U}_{L_i}(\omega)$...	$\mathbf{U}_{L_K}(\omega)$
⋮	⋮		⋮		⋮
n	$\mathbf{U}_{L_1}(n)$...	$\mathbf{U}_{L_i}(n)$...	$\mathbf{U}_{L_K}(n)$
					$\mathbf{V}_{L_0}(n)$

32.3 Determination of Fuzzy Regression Model

Once the model of experts' assessment process has been formulated, it becomes imperative to determine the fuzzy assessment function f pertaining to the total assessment (refer to Figure 32.2). This is a fuzzy function using which K fuzzy grades of attributes, \mathbf{U}_{L_i} , are transformed into a single fuzzy grade of the total assessment, denoted by \mathbf{V} . This fuzzy regression model is constructed on the basis of some training data $\omega (\omega = 1, 2, \dots, n)$ as provided by experts. The solution to this estimation problem could be offered by the fuzzy regression model presented by Watada *et al.* [16]. Table 32.1 shows the linguistic training data. These data are translated into membership grades expressed in terms of the elements of the dictionary (see Table 32.2). In this table, the fuzzy grades of attributes i and the total assessment of object ω are denoted by $U_{L_i}(\omega)$ and $V_{L_0}(\omega)$, respectively, where $i = 1, 2, \dots, K$ and $\omega = 1, 2, \dots, n$. We should note that the model is estimated using given linguistic data $L_0, L_1, L_2, \dots, L_K$.

Assume that all fuzzy grades have triangular membership functions. The construction of the fuzzy regression model f concerns the following optimization problem:

$$\begin{aligned} \mathbf{V}_{L_0} &= f(\mathbf{U}_{L_1}, \mathbf{U}_{L_2}, \dots, \mathbf{U}_{L_K}) \\ &= \sum_{i=1}^K \mathbf{A}_i \mathbf{U}_{L_i}(\omega) \quad \omega = 1, 2, \dots, n. \end{aligned} \quad (6)$$

The linear equation is an approximate expression of such a latent structure of human assessment. From this linguistic evaluation, it can be expressed using fuzzy numbers. When we emphasize linguistic words, we use the notation \mathbf{V}_{L_0} , which is given value instead that V is estimation. Using 'n' relations forming the training set, we have

$$\mathbf{V}_{L_0}(\omega) = \sum_{i=1}^K \mathbf{A}_i \mathbf{U}_{L_i}(\omega) \quad \omega = 1, 2, \dots, n. \quad (7)$$

In the above expression, we need to determine optimal fuzzy parameters A_i . Two optimization criteria are considered. One criterion concerns the fitness (goodness of fit) of the fuzzy regression model, h . The other one deals with the fuzziness captured by the fuzzy regression model, S . Let us elaborate on the detailed formulation of these criteria.

(i) Fitness.

Assume that an estimated value $\mathbf{V}_{L_0}(\omega)$ is obtained by the fuzzy linear function f and the fitness $b(\omega)$ of $\mathbf{Y}_0(\omega)$ for the sample value $\mathbf{Y}(\omega)$ is $\mathbf{V}_{L_0}(\omega)$ to a sample value $\mathbf{V}_{L_0}(\omega)$ is expressed in the form

$$h(\omega) = \bigvee_{y \in R} \left\{ \mu_{L_0(\omega)}(y) \bigwedge \mu_{L_0(\omega)}(y) \right\}. \quad (8)$$

(ii) Fuzziness.

The fuzziness \mathbf{S}^α included in the fuzzy function at α -level is defined by

$$\mathbf{S}^\alpha = \sum_{i=1}^K (\bar{a}_i - \underline{a}_i), \quad (9)$$

where \bar{a}_i and \underline{a}_i are numbers which specify the corresponding α -level set \mathbf{A}_i^α ; i.e.,

$$\mathbf{A}_i^\alpha = [\bar{a}_i, \underline{a}_i]. \quad (10)$$

Given the triangular membership function, we have

$$\mathbf{A}_i^\alpha \equiv [a_i - (1 - \alpha)c_i^l, a_i + (1 - \alpha)c_i^r]$$

as \mathbf{A}_i is defined by

$$\mathbf{A}_i \equiv (a_i, c_i^l, c_i^r).$$

Note that these two indices are in conflict. Higher values of fitness may (and will) result in excessively high values of fuzziness.

1. Formulation of the problem

We formulate a fuzzy assessment function by minimizing its fuzziness \mathbf{S} under the constraint that an estimated fuzzy grade of total assessment of each sample is fit to the fuzzy grade given by experts with adequateness greater than or equal to the given value h^0 , called fitness standard.

Problem. If data are given such as that listed in Table 32.2, the problem is to determine a fuzzy linear function

$$\mathbf{V}_{\mathbf{L}_0}(\omega) = \sum_{i=1}^K \mathbf{A}_i \cdot \mathbf{U}_{\mathbf{L}_i}(\omega) \quad (11)$$

which minimizes the level of fuzziness

$$\mathbf{S} = \sum_{i=1}^K (\bar{a}_i - \underline{a}_i) \quad (12)$$

under the conditions that

$$h(\omega) = \bigvee_{y \in R} \left\{ \mu_{\mathbf{L}(\omega)}(y) \bigwedge \mu_{\mathbf{L}_0(\omega)}(y) \right\} \geq h^0 \quad (13)$$

$$\omega = 1, 2, \dots, n,$$

where $h(\omega)$ appropriate linguistic words are selected by matching some value to a word in the linguistic dictionary. $h(S)$ indicates the fitness of the estimated value with respect to a sample ω and h^0 denotes the fitness standard, while \bar{a}_i and \underline{a}_i are defined as

$$A_i^{h^0} = [\underline{a}_i, \bar{a}_i], \quad (14)$$

Note that for computing with fuzzy numbers, we employ here their triangular approximation. Note however that detailed calculations could be carried out as well and we refer here to the results presented by Tanaka *et al.* [19].

Given this, the membership function $\mu_{\mathbf{L}_0}(y)$ of the fuzzy grade of structural total assessment \mathbf{V}_0 can be obtained through the use of the extension principle in the form

$$\mu_{\mathbf{V}_0}(y) = \begin{cases} \bigwedge_{i=1}^K \{\mu_{\mathbf{A}_i}(t_i) \wedge \mu_{\mathbf{L}_i}(u_i)\} & \\ \left| \begin{array}{l} y = \sum_i t_i u_i \\ 0 \leq t_i \leq 1 \\ 0 \leq u_i \leq 1 \end{array} \right. \end{cases} \quad (15)$$

When the value $\mathbf{V}_{\mathbf{L}_0}(\omega)$ given in (9) has been obtained, this enables us to define its membership function using the parameter t_i for \mathbf{A}_i and parameter u_i for \mathbf{L}_i of equation (9).

2. The fuzzy grade.

Here we discuss a heuristic method to determine a fuzzy assessment function by using non-fuzzy grades of assessment attributes; i.e., consider \mathbf{U}_i being fuzzy numbers in $[0, 1]$.

According to the sign of \mathbf{A}_i , the product of fuzzy number \mathbf{A}_i and $\mathbf{U}_{\mathbf{L}_i}$ involves three cases:

- (i) In the case where $\bar{a}_i \geq \underline{a}_i \geq 0$,

$$(\mathbf{A}_i \mathbf{U}_{\mathbf{L}_i})^{h^0} = [\underline{a}_i \underline{u}_i, \bar{a}_i \bar{u}_i]. \quad (16)$$

- (ii) In the case where $\bar{a}_i \leq \underline{a}_i \leq 0$,

$$(\mathbf{A}_i \mathbf{U}_{\mathbf{L}_i})^{h^0} = [\underline{a}_i \bar{u}_i, \bar{a}_i \underline{u}_i]. \quad (17)$$

- (iii) In the case where $\underline{a}_i \leq \bar{a}_i \leq 0$,

$$(\mathbf{U}_{\mathbf{L}_i})^{h^0} = [\underline{a}_i \bar{u}_i, \bar{a}_i \bar{u}_i]. \quad (18)$$

It is difficult to derive analytical solutions to this problem. Therefore, some heuristic approach is being sought. The proposed procedure can be outlined as follows.

An α -level set of the fuzzy degree of a structural attribute $\mathbf{U}_{\mathbf{L}_i}$ ($i = 1, 2, \dots, K$) at h^0 is denoted by (14).

Step 1. Let the trial count $r = 1$ and let us consider

$$(\mathbf{A}_i^{(r)} \mathbf{U}_{\mathbf{L}_i})^{h^0} = [\underline{a}_i^{(r)} \underline{u}_i, \bar{a}_i^{(r)} \bar{u}_i].$$

Determine $\bar{a}_i^{(r)}$, $\underline{a}_i^{(r)}$ ($i = 1, 2, \dots, K$) by the liner programming to minimize the fuzziness S defined by (12).

Step 2. (i) If $\underline{a}_i^{(r)}, \bar{a}_i^{(r)} \geq 0$, let

$$(\mathbf{A}_i^{(r+1)} \mathbf{U}_{\mathbf{L}_i})^{h^0} = [\underline{a}_i^{(r+1)} \underline{u}_i, \bar{a}_i^{(r+1)} \bar{u}_i]. \quad (19)$$

(ii) If $\underline{a}_i^{(r)}, \bar{a}_i^{(r)} \leq 0$, let

$$(\mathbf{A}_i^{(r+1)} \mathbf{U}_{\mathbf{L}_i})^{h^0} = [\underline{a}_i^{(r+1)} \bar{u}_i, \bar{a}_i^{(r+1)} \underline{u}_i]. \quad (20)$$

(iii) If $\underline{a}_i^{(r)} \leq 0 \leq \bar{a}_i^{(r)}$, let

$$(\mathbf{A}_i^{(r+1)} \mathbf{U}_{\mathbf{L}_i})^{h^0} = [\underline{a}_i^{(r+1)} \bar{u}_i, \bar{a}_i^{(r+1)} \bar{u}_i]. \quad (21)$$

Step 3. Determine $\bar{a}_i^{(r+1)}, \underline{a}_i^{(r+1)}$ ($i = 1, 2, \dots, K$) by the liner programming to minimize the fuzziness S under the constraints (9) according to the judgment of $(\mathbf{A}_i^{(r+1)} \mathbf{U}_{L_i})^{h^0}$ in Step 2.

Step 4. If $\underline{a}_i^{(r)} \bar{a}_i^{(r+1)} \geq 0$ and $\bar{a}_i^{(r)} \bar{a}_i^{(r+1)} \geq 0$ ($i = 1, 2, \dots, K$), then go to Step 6. Otherwise, let $r = r + 1$ and go to Step 5.

Step 5. If the trial count r has not exceeded the given threshold, then go to Step 2. Otherwise, terminate the procedure.

32.4 An Illustrative Example

Let us illustrate a performance of the model by applying it to an example of expert's damage assessment of a structure. In real damage assessment of existing structures, we have to inspect many portions of the structure for the detection of various defects concerning both structural damage and non-structural damage. We also analyze a collection of records of readings collected by various instruments such as acceleration data. Sometimes historical records of the structure play an important role in damage assessment. Moreover, the expected loading of the structure and environmental condition anticipated in the future must be considered when forecasting possible failures of the structure. An overloaded structure is required to maintain greater strength than that being required when in normal usage. For instance, the relationship between the present state of a structure and the expected loading and environmental condition can be illustrated as shown in Figure 32.3.

Here, for sake of clarity of presentation, we consider a somewhat limited scenario by considering only a small portion of a structure, say, a beam. We are asked to model the expert's procedure of the total damage assessment of this portion of a structure. For this purpose, let us consider the leading factors such as (1) cracking state, (2) corrosion state, and (3) expecting loading and environmental conditions being critical as to its total damage. Table 32.3 shows a dictionary of linguistic values and their fuzzy numbers employed in expressing the conditions of their attributes and the total damage, while Figure 32.4 includes the corresponding fuzzy numbers.

Assume that we are provided with 16 training samples, in which experts have evaluated the total damage of the structure. Furthermore the conditions of their attributes in linguistic form are shown in Table 32.4. As illustrated in Table 32.5, these linguistic values are translated into fuzzy numbers through the vocabulary translation unit given in Figure 32.2 and using the dictionary shown in Table 32.3. By

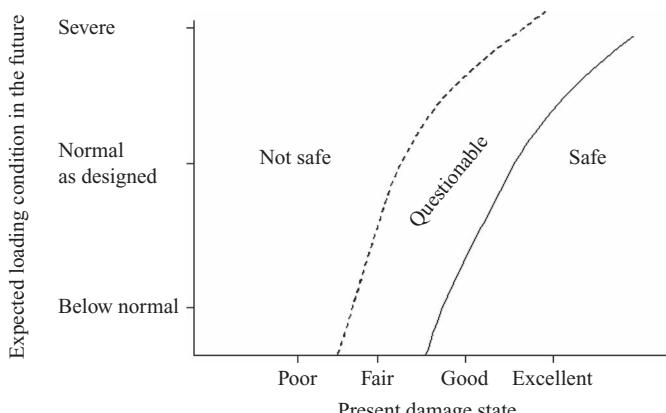


Figure 32.3 Relation between the expected loading condition and the present damage state

Table 32.3 Dictionary of attributes and their linguistic values

Name of attribute		Linguistic value	Fuzzy grade
Corrosion state	X_1	Extremely good	(0.00, 0.00, 0.15)
		Very good	(0.20, 0.15, 0.15)
		Good	(0.40, 0.15, 0.15)
		Bad	(0.60, 0.15, 0.15)
		Very bad	(0.80, 0.15, 0.15)
		Extremely bad	(1.00, 0.15, 0.15)
Cracking state	X_2	Extremely good	(0.00, 0.00, 0.15)
		Very good	(0.20, 0.15, 0.15)
		Good	(0.40, 0.15, 0.15)
		Bad	(0.60, 0.15, 0.15)
		Very bad	(0.80, 0.15, 0.15)
		Extremely bad	(1.00, 0.15, 0.15)
Expected loading and environmental condition	X_3	Below normal	(0.00, 0.00, 0.40)
		Normal	(0.50, 0.20, 0.20)
		Severe	(1.00, 0.40, 0.00)
Total assessment	Y	Safe	(0.00, 0.00, 0.40)
		Questionable	(0.50, 0.20, 0.20)
		Not safe	(1.00, 0.40, 0.00)

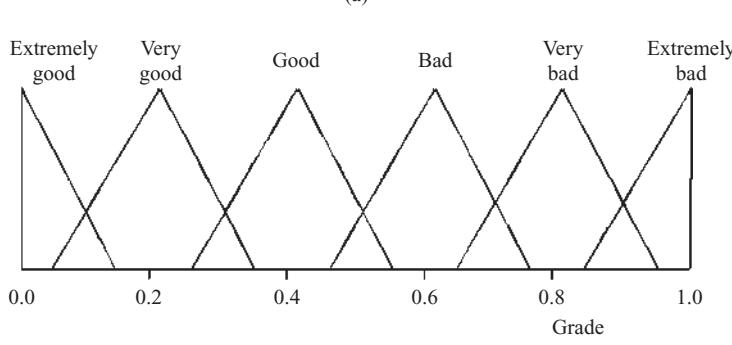
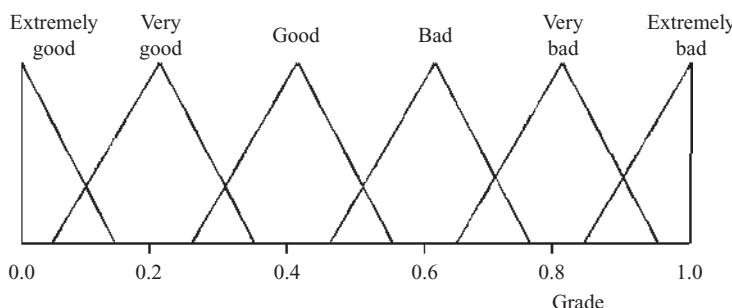
**Figure 32.4** Linguistic values of the dictionary: (a) corrosion state, and (b) cracking state

Table 32.4 Linguistic data of training samples

Training sample	Attribute			
	Corrosion state	Cracking state	Expected loading and environmental condition	Total assessment
	X ₁	X ₂	X ₃	Y
1	Extremely good	Extremely bad	Below normal	Not safe
2	Bad	Very bad	Below normal	Not safe
3	Very bad	Very bad	Below normal	Not safe
4	Very bad	Very good	Below normal	Questionable
5	Good	Extremely good	Below normal	Safe
6	Good	Bad	Normal	Not safe
7	Bad	Very bad	Normal	Not safe
8	Very bad	Good	Normal	Not safe
9	Bad	Very good	Normal	Questionable
10	Very bad	Very good	Normal	Questionable
11	Extremely good	Bad	Severe	Not safe
12	Very good	Bad	Severe	Not safe
13	Good	Very bad	Severe	Not safe
14	Very bad	Good	Severe	Not safe
15	Very good	Good	Severe	Not safe
16	Very good	Very good	Severe	Questionable

Table 32.5 Fuzzy grades translated from linguistic data in Table 32.4

Training sample	Attribute			
	Corrosion state	Cracking state	Expected loading and environmental condition	Total assessment
	X ₁	X ₂	X ₃	Y
1	(0.00, 0.00, 0.15)	(0.00, 0.00, 0.15)	(0.00, 0.00, 0.40)	(1.00, 0.40, 0.00)
2	(0.60, 0.15, 0.15)	(0.80, 0.15, 0.15)	(0.00, 0.00, 0.40)	(1.00, 0.40, 0.00)
3	(0.80, 0.15, 0.15)	(0.80, 0.15, 0.15)	(0.00, 0.00, 0.40)	(1.00, 0.40, 0.00)
4	(0.80, 0.15, 0.15)	(0.20, 0.15, 0.15)	(0.00, 0.00, 0.40)	(0.50, 0.20, 0.20)
5	(0.40, 0.15, 0.15)	(0.00, 0.00, 0.15)	(0.00, 0.00, 0.40)	(0.00, 0.00, 0.40)
6	(0.40, 0.15, 0.15)	(0.60, 0.15, 0.15)	(0.50, 0.20, 0.20)	(1.00, 0.40, 0.00)
7	(0.60, 0.15, 0.15)	(0.80, 0.15, 0.15)	(0.50, 0.20, 0.20)	(1.00, 0.40, 0.00)
8	(0.80, 0.15, 0.15)	(0.40, 0.15, 0.15)	(0.50, 0.20, 0.20)	(1.00, 0.40, 0.00)
9	(0.60, 0.15, 0.15)	(0.20, 0.15, 0.15)	(0.50, 0.20, 0.20)	(0.50, 0.20, 0.20)
10	(0.80, 0.15, 0.15)	(0.20, 0.15, 0.15)	(0.50, 0.20, 0.20)	(0.50, 0.20, 0.20)
11	(0.00, 0.00, 0.15)	(0.60, 0.15, 0.15)	(1.00, 0.40, 0.00)	(1.00, 0.40, 0.00)
12	(0.20, 0.15, 0.15)	(0.60, 0.15, 0.15)	(1.00, 0.40, 0.00)	(1.00, 0.40, 0.00)
13	(0.40, 0.15, 0.15)	(0.80, 0.15, 0.15)	(1.00, 0.40, 0.00)	(1.00, 0.40, 0.00)
14	(0.80, 0.15, 0.15)	(0.40, 0.15, 0.15)	(1.00, 0.40, 0.00)	(1.00, 0.40, 0.00)
15	(0.20, 0.15, 0.15)	(0.40, 0.15, 0.15)	(1.00, 0.40, 0.00)	(1.00, 0.40, 0.00)
16	(0.20, 0.15, 0.15)	(0.20, 0.15, 0.15)	(1.00, 0.40, 0.00)	(0.50, 0.20, 0.20)

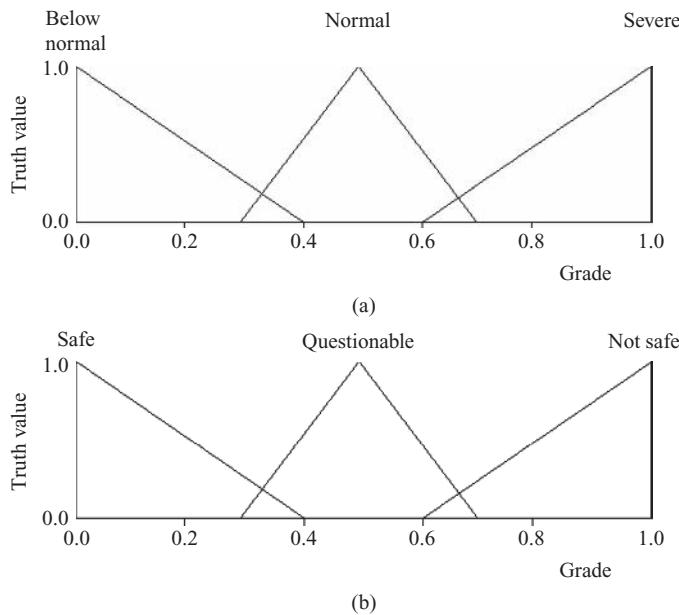


Figure 32.5 Linguistic values in dictionary(continued): (a) expected loading condition, and (b) Total assessment.

applying the heuristic method to these fuzzy numbers, we can obtain the total assessment model on the basis of condition of attributes. Because it is essential to obtain high fitness of the expert model to the real cases, in our investigations, we have employed a value 0.8 as this fitness threshold. In other words, when we obtained linguistic expressions experts gave, the translation from linguistic words to fuzzy numbers enables us to formulate a fuzzy regression model that is supposed under the relation among linguistic words. Finally, Table 32.6 includes the details of the resulting model.

Let us consider sample 1 in Table 32.4. In this case, the corrosion state is '*extremely good*' and the loading and environment condition in the future is expected to be '*below normal*', but its cracking state is '*extremely bad*'. On the basis of these linguistic evaluations, experts assessed its total damage state as '*not safe*' (see Table 32.4). Using the dictionary in our model as shown in Table 32.3, the words '*extremely good*' for corrosion state, '*extremely bad*' for cracking state, and '*below normal*' for expected loading and environment condition are translated into $\mu_{(\text{corrosion})} = (0.00, 0.00, 0.15)$, $\mu_{(\text{cracking})} = (1.00, 0.15, 0.00)$, and $\mu_{(\text{loading})} = (0.00, 0.00, 0.40)$, respectively. By using the fuzzy

Table 32.6 Coefficients in estimation unit

Attribute		
Corrosion state	Cracking state	Expected loading and environmental condition
\mathbf{X}_1	\mathbf{X}_2	\mathbf{X}_3
(0.322, 0.000, 0.000)	(0.813, 0.00, 0.00)	(0.373, 0.065, 0.065)

Fitting value = 0.70.

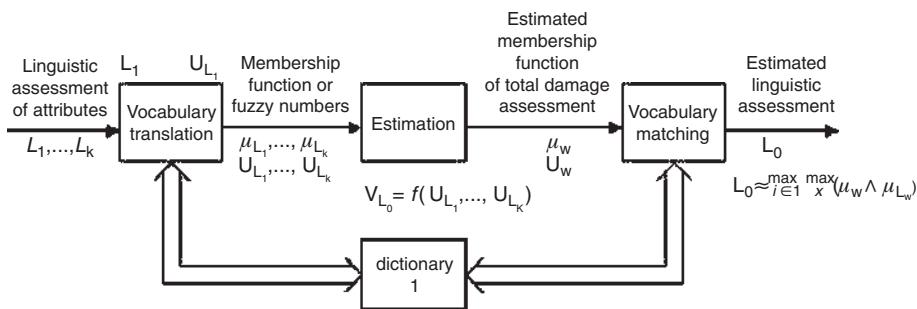


Figure 32.6 Process of linguistic regression model

assessment function

$$\begin{aligned} V &= f(U_{L_1}, U_{L_2}, U_{L_3}) \\ &= (0.322, 0.000, 0.000)\mu_{\text{(corrosion)}} \\ &\quad + (0.813, 0.000, 0.00)\mu_{\text{(cracking)}} \\ &\quad + (0.373, 0.065, 0.65)\mu_{\text{(loading)}}, \end{aligned}$$

we can estimate the associated total damage as

$$V = \mu_{\text{(total damage)}} = (0.81, 0.12, 0.29).$$

Through the matching process as shown in Figure 32.6, we obtain

$$\begin{aligned} \max_t \{\mu_{\text{(total damage)}} \wedge \mu_{\text{(safe)}}(t)\} &= 0.0, \\ \max_t \{\mu_{\text{(total damage)}} \wedge \mu_{\text{(questionable)}}(t)\} &= 0.03, \text{ and} \\ \max_t \{\mu_{\text{(total damage)}} \wedge \mu_{\text{(not safe)}}(t)\} &= 0.66. \end{aligned}$$

Table 32.7 Linguistic data of training samples

Training sample	Value provided by experts		Estimated value by the model	
	Linguistic value	Translated fuzzy grade	Estimated fuzzy grade	Matched word
1	Not safe	(1.00, 0.40, 0.00)	(0.81, 0.12, 0.29)	Not safe
2	Not safe	(1.00, 0.40, 0.00)	(0.84, 0.17, 0.56)	Not safe
3	Not safe	(1.00, 0.40, 0.00)	(0.91, 0.17, 0.40)	Not safe
4	Questionable	(0.50, 0.20, 0.20)	(0.42, 0.17, 0.41)	Questionable
5	Safe	(0.00, 0.00, 0.40)	(0.13, 0.05, 0.41)	Safe
6	Not safe	(1.00, 0.40, 0.00)	(0.80, 0.31, 0.40)	Not safe
7	Not safe	(1.00, 0.40, 0.00)	(1.03, 0.31, 0.40)	Not safe
8	Not safe	(1.00, 0.40, 0.00)	(0.77, 0.31, 0.35)	Not safe
9	Questionable	(0.50, 0.20, 0.20)	(0.54, 0.22, 0.40)	Questionable
10	Questionable	(0.50, 0.20, 0.20)	(0.61, 0.31, 0.39)	Questionable
11	Not safe	(1.00, 0.40, 0.00)	(0.86, 0.40, 0.39)	Not safe
12	Not safe	(1.00, 0.40, 0.00)	(0.93, 0.45, 0.38)	Not safe
13	Not safe	(1.00, 0.40, 0.00)	(1.15, 0.45, 0.39)	Not safe
14	Not safe	(1.00, 0.40, 0.00)	(0.96, 0.45, 0.39)	Not safe
15	Not safe	(1.00, 0.40, 0.00)	(0.76, 0.45, 0.40)	Not safe
16	Questionable	(0.50, 0.20, 0.20)	(0.60, 0.45, 0.39)	Questionable

Table 32.8 Linguistic data of new samples and their estimated total assessment

Training sample	Attribute			
	Corrosion state	Cracking state	Expected loading and environmental condition	Estimated Total assessment
	X ₁	X ₂	X ₃	Y
21	Good	Extremely bad	Below normal	Not safe
22	Very good	Very good	Normal	Not safe
23	Bad	Extremely good	Below normal	Questionable
24	Very good	Extremely good	Below normal	Safe
25	Bad	Very bad	Normal	Not safe

Therefore, the model has assigned the same expression ‘not safe’ to the total damage of this sample 1 as the experts did (see Table 32.8).

The estimated linguistic values can be obtained through the vocabulary matching unit when using the dictionary given in Table 32.7. Table 32.7 quantifies the performance of the model. Let us verify the performance of the model by considering some additional new samples. Table 32.8 includes linguistic data of these samples along with the estimated results of their total assessment.

32.5 Concluding Remarks

We have introduced a fuzzy regression model as an algorithmic vehicle realizing computing with words. We stressed the role of experts in accumulation of domain knowledge and experience. Experts frequently express their judgments in terms of linguistic expressions rather than pure numeric entities. In this sense, the linguistic treatment of assessments becomes essential when fully reflecting the subjectivity of the judgment process.

The process presented in this chapter involves four phases. The linguistic evaluation of the total assessment is produced through vocabulary matching. We have employed a fuzzy regression model to estimate the values of the total assessment developed in terms of the linguistic structural attributes.

References

- [1] L.A. Zadeh. What is computing with words. In: L.A. Zadeh and J. Kacprzyk (eds), *Computing with Words in Information/Intelligent Systems, Foundation*, Physica-Verlag, Heidelberg, 2006, pp. VIII–IX.
- [2] L.A. Zadeh. Fuzzy Logic: = computing with words. In: L.A. Zadeh and J. Kacprzyk (eds), *Computing with Words in Information/Intelligent Systems, Foundation*, Physica-Verlag, Heidelberg, 2006, pp. 3–23.
- [3] L.A. Zadeh. Fuzzy sets and information granularity. In: M.M. Gupta, R.K. Ragade, and R.R. Yager (eds), *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, 1979, pp. 3–18.
- [4] W. Pedrycz (ed.). *Granular Computing: An Emerging Paradigm*. Physica-Verlag Heidelberg, 2001.
- [5] W. Pedrycz. Computational intelligence as an emerging paradigm of software engineering. Keynote Speech. In: *Proceedings of SEKE2002*, July 15–19, 2002, Ischia, Italy, 2002, pp. 7–14.
- [6] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, 1980.
- [7] S. Imoto, Y. Yabuuchi, and J. Watada. Fuzzy regression model of R&D project evaluation. *Applied Soft Computing*, Special Issue: Forging new frontiers. Elsevier, New York, to appear.
- [8] H. Tanaka and H. Lee. Interval regression analysis by quadratic programming approach. *IEEE Trans. Fuzzy Syst.* 6 (1998) 473–481.
- [9] H. Tanaka, S. Uejima, and K. Asai. Linear regression analysis with fuzzy model. *IEEE Trans. Syst. Man Cybern. SMC-12* (1982) 903–907.
- [10] L.A. Zadeh. Fuzzy sets. *Inf. Control* 8 (1965) 338–353.

- [11] L.A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning, Part 1. *Inf. Sci.* 8 (1975) 199–249.
- [12] L.A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning, Part 2. *Inf. Sci.* 8 (1975) 301–357.
- [13] J. Watada, K.-S. Fu, and J.T.P. Yao. *Damage Assessment Using Fuzzy Multivariant Analysis*. Technical Report No. CE-STR-84-4. School of Civil Engineering, Purdue University, West Lafayette, 1984.
- [14] J. Watada, K.-S. Fu, and J.T.P. Yao. Fuzzy classification approach to damage assessment. In: *Proceedings of First International Conference on Fuzzy Information Professing*, Hawaii, 1984, pp. 235–240.
- [15] Y. Toyoura, J. Watada, M. Khalid, and R. Yusof. Formulation of linguistic regression model based on natural words. *Soft Comput. J.* 8 (2004) 681–688.
- [16] J. Watada, H. Tanaka, and K. Asai. Fuzzy quantification theory type I. *Japan. J. Behaviormetr.* 11 (1983) 66–73 [in Japanese].
- [17] J. Watada. Multiattribute decision-making. In: T. Terano, K. Asai, and M. Sugeno (eds), *Applied Fuzzy System*, Academic Press, New York, 1994, pp. 244–252.
- [18] J. Watada. The thought and model of linguistic regression. In: *Proceedings the 9th World Congress of International Fuzzy Systems Association*, Canada, 2001, pp. 340–346.
- [19] H. Tanaka, J. Watada, and K. Asai. Evaluation of alternatives with multiple attribute based on fuzzy sets, systems and control. *Japan Assoc. Autom. Control Eng.* 27 (1983) 403–409.

33

Fuzzy Associative Memories and Their Relationship to Mathematical Morphology

Peter Sussner and Marcos Eduardo Valle

33.1 Introduction

Fuzzy associative memories (FAMs) belong to the class of *fuzzy neural networks* (FNNs). A FNN is an *artificial neural network* (ANN) whose input patterns, output patterns, and/or connection weights are fuzzy valued [1, 2].

Research on FAM models originated in the early 1990s with the advent of Kosko's FAM [3, 4]. Like many other associative memory (AM) models, Kosko's FAM consists of a single-layer feedforward FNN that stores the fuzzy rule ‘If x is X_k then y is Y_k ’ using a fuzzy Hebbian learning rule in terms of max–min or max–product compositions for the synthesis of its weight matrix W .

Despite successful applications of Kosko's FAMs to problems such as backing up a truck and trailer [3], target tracking [4], and voice cell control in ATM networks [5], Kosko's FAM suffers from an extremely low storage capacity of one rule per FAM matrix. Therefore, Kosko's overall fuzzy system comprises several FAM matrices. Given a fuzzy input, the FAM matrices generate fuzzy outputs which are then combined to yield the final result. To overcome the original FAMs severe limitations in storage capacity, several researchers have developed improved FAM versions that are capable of storing multiple pairs of fuzzy patterns [6–10]. For example, Chung and Lee generalized Kosko's model by proposing a max– t composition for the synthesis of a FAM matrix. Chung and Lee showed that all fuzzy rules can be perfectly recalled by means of a single FAM matrix using max– t composition provided that the input patterns satisfy certain orthogonality conditions [8]. Junbo *et al.* had previously presented an improved learning algorithm for Kosko's max–min FAM model [6, 11]. Liu modified Junbo's FAM *et al.* by adding a threshold activation function to each node of the network [9].

We recently established *implicative fuzzy associative memories* (IFAMs) [12, 13], a class of AMs that grew out of *morphological associative memories* (MAMs) [14–16]. One particular IFAM model can be viewed as an improved version of Liu's FAM [13]. MAMs belong to the class of morphological neural networks (MNNs) [17, 18]. This class of ANNs is called *morphological* because each node performs a morphological operation [19–22]. Theory and applications of binary and grayscale MAMs have been developed since late 1990s [14–16, 23]. For example, one can store as many patterns as desired in an autoassociative MAM [14, 23, 24]. In particular, for binary patterns of length n , the binary autoassociative

MAM exhibits an absolute storage capacity of 2^n , which either equals or slightly exceeds the storage capacity of the *quantum associative memory* of Ventura and Martinez [25]. Applications of MAMs include face localization, robot vision, hyperspectral image analysis, and some general classification problems [15, 26–29].

This chapter demonstrates that the IFAM model as well as all other FAM models that we mentioned above can be embedded into the general class of *fuzzy morphological associative memories* (FMAMs). *Fuzzy logical bidirectional associative memories* (FLBAMs), which were introduced by Bělohlávek [30], can also be considered a subclass of FMAMs. Although a general framework for FMAMs has yet to appear in the literature, we believe that the class of FMAMs should be firmly rooted in fuzzy mathematical morphology and thus each node of an FMAM should execute a fuzzy morphological operation [31–33]. In general, the input, output, and synaptic weights of FMAMs are fuzzy valued. Recall that fuzzy sets represent special cases of information granules. Thus, FMAMs can be considered special cases of *granular associative memories*, a broad class of AMs which has yet to be investigated.

The chapter is organized as follows. First, we present some background information and motivation for our research. After providing some general concepts of neural associative memories, fuzzy set theory, and mathematical morphology, we discuss the types of artificial neurons that occur in FAM models. Section 33.5 provides an overview of Kosko's FAM and its generalizations, including the FAM model of Chung and Lee. In Section 33.6, we review variations of Kosko's max-min FAM, in particular the models of Junbo *et al.* and Liu in conjunction with their respective learning strategies. In Section 33.7, we present the most important results on IFAMs and FLBAMs. Section 33.8 compares the performances of different FAM models by means of an example concerning the storage capacity and noise tolerance. Furthermore, an application to a problem of prediction is presented. We conclude the chapter with some suggestions for further research concerning fuzzy and granular MAM models.

33.2 Some Background Information and Motivation

33.2.1 Associative Memories

AMs allow for the storage of pattern associations and the retrieval of the desired output pattern on presentation of a possibly noisy or incomplete version of an input pattern. Mathematically speaking, the AM design problem can be stated as follows: Given a finite set of desired associations $\{(x^\xi, y^\xi) : \xi = 1, \dots, k\}$, determine a mapping G such that $G(x^\xi) = y^\xi$ for all $\xi = 1, \dots, k$. Furthermore, the mapping G should be endowed with a certain tolerance with respect to noise; i.e., $G(\tilde{x}^\xi)$ should equal y^ξ for noisy or incomplete versions \tilde{x}^ξ of x^ξ . In the context of *granular computing* (GC), the input and the output patterns are information granules [34].

The set of associations $\{(x^\xi, y^\xi) : \xi = 1, \dots, k\}$ is called *fundamental memory set* and each association (x^ξ, y^ξ) in this set is called a *fundamental memory* [35]. We speak of an *autoassociative memory* when the fundamental memory set is of the form $\{(x^\xi, x^\xi) : \xi = 1, \dots, k\}$. The memory is said to be *heteroassociative* if the output y^ξ is different from the input x^ξ . One of the most common problem associated with the design of an AM is the creation of false or spurious memories. A *spurious memory* is a memory association that does not belong to the fundamental memory set; i.e., it was unintentionally stored in the memory.

The process of determining G is called *recording phase* and the mapping G is called *associative mapping*. We speak of a *neural associative memory* when the associative mapping G is described by an ANN. In particular, we have a *fuzzy (neural) associative memory* (FAM) if the associative mapping G is given by an FNN and the patterns x^ξ and y^ξ are fuzzy sets for every $\xi = 1, \dots, k$.

33.2.2 Morphological Neural Networks

In this chapter, we are mainly concerned with FAMs. As we shall point out during the course of this chapter, many models of FAMs can be classified as FMAMs, which in turn belong to the class of MNNs [17, 28].

The name ‘morphological neural network’ was coined because MNNs perform operations of mathematical morphology (MM) at every node.

Many models of MNNs are implicitly rooted in the mathematical structure $(\mathbb{R}_{\pm\infty}, \vee, \wedge, +, +')$, which represents a *bounded lattice ordered group* (blog) [14–17, 36–40]. The symbols ‘ \vee ’ and ‘ \wedge ’ represent the maximum and the minimum operation. The operations ‘ $+$ ’ and ‘ $+'$ act like the usual sum operation and are identical on $\mathbb{R}_{\pm\infty}$, with the following exceptions:

$$(-\infty) + (+\infty) = (+\infty) + (-\infty) = -\infty \quad \text{and} \quad (-\infty) +' (+\infty) = (+\infty) +' (-\infty) = +\infty. \quad (1)$$

In practice, the inputs, outputs, and synaptic weights of an MNN have values in \mathbb{R} where the operations ‘ $+$ ’ and ‘ $+'$ coincide.

In most cases, models of MNNs, including MAMs, are defined in terms of certain matrix products known as the max product and the min product. Specifically, for an $m \times p$ matrix A and a $p \times n$ matrix B with entries from $\mathbb{R}_{\pm\infty}$, the matrix $C = A \boxtimes B$, also called the *max product* of A and B , and the matrix $D = A \boxdot B$, also called the *min product* of A and B , are defined by

$$c_{ij} = \bigvee_{k=1}^p (a_{ik} + b_{kj}) \quad \text{and} \quad d_{ij} = \bigwedge_{k=1}^p (a_{ik} +' b_{kj}). \quad (2)$$

Let us consider an arbitrary neuron in an MNN defined on the blog $(\mathbb{R}_{\pm\infty}, \vee, \wedge, +, +')$. Suppose that the inputs are given by a vector $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ and let $\mathbf{w} = (w_1, \dots, w_n)^T \in \mathbb{R}^n$ denote the vector of corresponding synaptic strengths. The accumulative effect of the inputs and the synaptic weights in a simple morphological neuron is given by either one of the following equations:

$$\tau(\mathbf{x}) = \mathbf{w}^T \boxtimes \mathbf{x} = \bigvee_{i=1}^n (w_i + x_i) \quad \text{or} \quad \tau(\mathbf{x}) = \mathbf{w}^T \boxdot \mathbf{x} = \bigwedge_{i=1}^n (w_i +' x_i). \quad (3)$$

Since the equations in (3) are non-linear, researchers in the area of MNNs generally refrain from using a possibly non-linear activation function. It should be mentioned that Koch and Poggio make a strong case for multiplying with synapses [41], i.e., for $w_i \cdot x_i$ instead of $w_i + x_i$ or $w_i +' x_i$ as written in the equations in (3). However, multiplication could have been used just as well in these equations because the blog $(\mathbb{R}_{\pm\infty}, \vee, \wedge, +, +')$ is isomorphic to the blog $([0, \infty], \vee, \wedge, \cdot, \cdot')$ under the isomorphism $\phi(x) = e^x$. (We use the conventions $e^{-\infty} = 0$ and $e^\infty = \infty$.) Here, the multiplications ‘ \cdot ’ and ‘ \cdot' ’ generally behave as one would expect with the following exceptions:

$$0 \cdot \infty = \infty \cdot 0 = 0 \quad \text{and} \quad 0 \cdot' \infty = \infty \cdot' 0 = +\infty. \quad (4)$$

Note that in the multiplicative blog $([0, \infty], \vee, \wedge, \cdot, \cdot')$, the equations in (3) become, respectively,

$$\tau(\mathbf{x}) = \bigvee_{i=1}^n (w_i \cdot x_i) \quad \text{and} \quad \tau(\mathbf{x}) = \bigwedge_{i=1}^n (w_i \cdot' x_i). \quad (5)$$

Despite the facts that weights are generally considered to be positive quantities and that MNNs can also be developed in the multiplicative blog $([0, \infty], \vee, \wedge, \cdot, \cdot')$, computational reasons have generally led researchers to work in the additive blog $(\mathbb{R}_{\pm\infty}, \vee, \wedge, +, +')$ [13]. In fact, it is sufficient to consider the blog $(\mathbb{Z}_{\pm\infty}, \vee, \wedge, +, +')$. Moreover, the equations in (3) are closely linked to the operations of grayscale dilation and erosion in classical MM [42, 43]. These equations can also be interpreted as non-linear operations (image-template products) in the mathematical structure of image algebra [44, 45]. In fact, existing formulations of traditional neural network models in image algebra induced researchers such as Ritter, Davidson, Gader, and Sussner to formulate models of MNNs [14–17, 37, 38].

Thus, the motivation for establishing MNNs can be found in mathematics instead of biology. Nevertheless, recent research results by Yu, Giese, and Poggio have revealed that the maximum operation that lies at the core of morphological neurons is neurobiologically plausible [46]. In addition to its potential involvement in a variety of cortical processes [47–49], the maximum operation can be implemented by

simple, neurophysiologically plausible circuits. Previously, prominent neuroscientists such as Gurney, Segev, and Shepherd had already shown that simple logical functions can be modeled by local interactions in dendritic trees [50–52].

For fuzzy inputs $\mathbf{x} \in [0, 1]^n$ and fuzzy weights $\mathbf{w} \in [0, 1]^n$, the identity on the left-hand side of the first equation in (5) describes a fuzzy morphological neuron because it corresponds to an operation of dilation in fuzzy mathematical morphology [32, 53]. Note that the operation of multiplication represents a special case of fuzzy conjunction. At this point, we prefer not to go into the details of fuzzy morphological neural networks, in particular FMAMs. We would only like to point out that the lattice ordering of fuzzy sets has been paramount to the development of FMAMs. Thus, the lattice ordering of other information granules may turn out to be useful for the development of other granular associative memories.

33.2.3 Information Granules and Their Inherent Lattice Ordering

GC is based on the observation that we are only able to process the incoming flow of information by means of a process of abstraction which involves representing information in the form of aggregates or information granules [34, 54–57]. Thus, granulation of information occurs in everyday life whenever we form collections of entities that are arranged together due to their similarity, functional adjacency, indistinguishability, coherency, or alike.

These considerations indicate that set theory serves as a suitable conceptual and algorithmic framework for GC. Since a given class of sets is equipped with a partial ordering given by set inclusion, we believe that GC is closely related to lattice theory. More formally speaking, information granules include fuzzy sets, rough sets, intervals, shadowed sets, and probabilistic sets. Observe that all of these classes of constructs are endowed with an inherent lattice ordering.

In this chapter, we focus our attention on the class of fuzzy sets $[0, 1]^X$, i.e., the set of functions from a universe X to $[0, 1]$, because we are not aware of any significant research results concerning other classes of information granules in the context of AMs. However, we believe that their inherent lattice structure will provide for the means to establish AMs that store associations of other types of information granules.

Thus, this chapter is concerned with FAMs. More precisely, we describe a relationship between FAMs and MM that is ultimately due to the complete lattice structure of $[0, 1]^X$.

33.3 Relevant Concepts of Fuzzy Set Theory and Mathematical Morphology

33.3.1 The Complete Lattice Framework of Mathematical Morphology

In this chapter, we will establish a relationship between FAMs and MM that is due to the fact that the neurons of most FAM models perform morphological operations.

MM is a theory which is concerned with the processing and analysis of objects using operators and functions based on topological and geometrical concepts [22, 42]. This theory was introduced by Matheron and Serra in the early 1960s as a tool for the analysis of binary images [19, 58]. During the last decades, it has acquired a special status within the field of image processing, pattern recognition, and computer vision. Applications of MM include image segmentation and reconstruction [59], feature detection [60], and signal decomposition [61].

The most general mathematical framework in which MM can be conducted is given by complete lattices [21, 22]. A *complete lattice* is defined as a partially ordered set \mathbb{L} in which every (finite or infinite) subset has an infimum and a supremum in \mathbb{L} [62]. For any $Y \subseteq \mathbb{L}$, the infimum of Y is denoted by the symbol $\bigwedge Y$. Alternatively, we write $\bigwedge_{j \in J} y_j$ instead of $\bigwedge Y$ if $Y = \{y_j : j \in J\}$ for some index set J . Similar notations are used to denote the supremum of Y . The interval $[0, 1]$ represents an example of a complete lattice. The class of *fuzzy sets* $[0, 1]^X$, i.e., the set of functions from a universe X to $[0, 1]$, inherits the complete lattice structure of the unit interval $[0, 1]$.

The two basic operators of MM are *erosion* and *dilation* [20, 22]. An *erosion* is a mapping ε from a complete lattice \mathbb{L} to a complete lattice \mathbb{M} that commutes with the infimum operation. In other words, the operator ε represents an erosion if and only if the following equality holds for every subset $Y \subseteq \mathbb{L}$:

$$\varepsilon\left(\bigwedge Y\right) = \bigwedge_{y \in Y} \varepsilon(y). \quad (6)$$

Similarly, an operator $\delta : \mathbb{L} \rightarrow \mathbb{M}$ that commutes with the supremum operation is called a *dilation*. In other words, the operator δ represents a dilation if and only if the following equality holds for every subset $Y \subseteq \mathbb{L}$:

$$\delta\left(\bigvee Y\right) = \bigvee_{y \in Y} \delta(y). \quad (7)$$

Apart from erosions and dilations, we will also consider the elementary operators anti-erosion and anti-dilation that are defined as follows [22, 63]. An operator $\bar{\varepsilon}$ is called an anti-erosion if and only if the first equality in (8) holds for every $Y \subseteq \mathbb{L}$ and an operator $\bar{\delta}$ is called an anti-dilation if and only if the second equality in (8) holds for every subset $Y \subseteq \mathbb{L}$.

$$\bar{\varepsilon}\left(\bigwedge Y\right) = \bigvee_{y \in Y} \bar{\varepsilon}(y) \quad \text{and} \quad \bar{\delta}\left(\bigvee Y\right) = \bigwedge_{y \in Y} \bar{\delta}(y). \quad (8)$$

Erosions, dilations, anti-erosions, and anti-dilations exemplify the concept of morphological operator. Unfortunately, a rigorous mathematical definition of a morphological operator does not exist. According to Heijmans, any attempt to find a formal definition of a morphological operator would either be too restrictive or too general [22]. For the purposes of our chapter, it is sufficient to know that the four elementary operators erosion, dilation, anti-erosion, and anti-dilation are generally considered to be morphological ones [63].

If one of the four operators ε , δ , $\bar{\varepsilon}$, or $\bar{\delta}$ that we defined above is a mapping $[0, 1]^X \rightarrow [0, 1]^Y$ for some sets X and Y , then we speak of a *fuzzy erosion*, a *fuzzy dilation*, a *fuzzy anti-erosion*, or a *fuzzy anti-dilation* [31, 32, 53]. The operators of erosion and dilation are often linked in terms of either one of the following relationships of duality: adjunction or negation.

Let \mathbb{L} and \mathbb{M} be complete lattices. Consider two arbitrary operators $\delta : \mathbb{L} \rightarrow \mathbb{M}$ and $\varepsilon : \mathbb{M} \rightarrow \mathbb{L}$. We say that (ε, δ) is an *adjunction* from \mathbb{L} to \mathbb{M} if we have

$$\delta(x) \leq y \Leftrightarrow x \leq \varepsilon(y) \quad \forall x \in \mathbb{L}, y \in \mathbb{M}. \quad (9)$$

Adjunction constitutes a duality between erosions and dilations since they form a bijection which reverses the order relation in the complete lattice [22]. Moreover, if (ε, δ) is an adjunction, then δ is a dilation and ε is an erosion.

A second type of duality is based on *negation*. We define a *negation* on a complete lattice \mathbb{L} as an involutive bijection $v_{\mathbb{L}} : \mathbb{L} \rightarrow \mathbb{L}$, which reverses the partial ordering. In the special case where $\mathbb{L} = [0, 1]$, we speak of a *fuzzy negation*. Examples of fuzzy negations include the following unary operators.

$$N_S(x) = 1 - x \quad \text{and} \quad N_D(x) = \frac{1 - x}{1 + px} \quad \text{for } p > -1. \quad (10)$$

Suppose that N is an arbitrary fuzzy negation and that $\mathbf{x} \in [0, 1]^n$ and $W \in [0, 1]^{m \times n}$. For simplicity, $N(\mathbf{x})$ denotes the component-wise fuzzy negation of the vector \mathbf{x} and $N(W)$ denotes the entry-wise fuzzy negation of the matrix W .

Let Ψ be an operator mapping a complete lattice \mathbb{L} into a complete lattice \mathbb{M} and let $v_{\mathbb{L}}$ and $v_{\mathbb{M}}$ be negations on \mathbb{L} and \mathbb{M} , respectively. The operator Ψ^v given by

$$\Psi^v(x) = v_{\mathbb{M}}(\Psi(v_{\mathbb{L}}(x))) \quad \forall x \in \mathbb{L}, \quad (11)$$

is called the *negation* or the *dual* of Ψ with respect to $v_{\mathbb{L}}$ and $v_{\mathbb{M}}$. The negation of an erosion is a dilation and vice versa [22]. The preceding observations clarify that there is a unique erosion that can be associated with a certain dilation and vice versa in terms of either negation or adjunction.

An erosion, a dilation respectively, is usually associated with a *structuring element* (SE) which is used to probe a given image [19, 42]. In the fuzzy setting, the image \mathbf{a} and the SE \mathbf{s} are given by fuzzy sets [31, 32, 53]. For a fixed SE \mathbf{s} , a fuzzy dilation $\mathcal{D}(\cdot, \mathbf{s})$ is usually defined in terms of a supremum of fuzzy conjunctions C , where C commutes with the supremum operator in the second argument [31, 32, 64]. Similarly, a fuzzy erosion $\mathcal{E}(\cdot, \mathbf{s})$ can be defined in terms of an infimum of fuzzy disjunctions D or an infimum of fuzzy implications I , where D or I commutes with the infimum operator in the second argument.

If an ANN performs a (fuzzy) morphological operation at each node, we speak of a (*fuzzy*) *morphological neural network*. The neurons of an ANN of this type are called (*fuzzy*) *morphological neurons*. In particular, (fuzzy) neurons that perform dilations, erosions, anti-dilations, or anti-erosions are (fuzzy) morphological neurons. An AM that belongs to the class of FMNNs is called an FMAM.

33.3.2 Some Basic Operators of Fuzzy Logic

This chapter will show that – in their most general form – the neurons of FAMs are given in terms of a fuzzy conjunction, a fuzzy disjunction, or a fuzzy implication.

We define a *fuzzy conjunction* as an increasing mapping $C : [0, 1] \times [0, 1] \rightarrow [0, 1]$ that satisfies $C(0, 0) = C(0, 1) = C(1, 0) = 0$ and $C(1, 1) = 1$. The minimum operator and the product obviously yield simple examples. In particular, a commutative and associative fuzzy conjunction $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$ that satisfies $T(x, 1) = x$ for every $x \in [0, 1]$ is called *triangular norm* or simply *t-norm* [65]. The fuzzy conjunctions C_M , C_P , and C_L below are examples of *t*-norms.

$$C_M(x, y) = x \wedge y, \quad (12)$$

$$C_P(x, y) = x \cdot y, \quad (13)$$

$$C_L(x, y) = 0 \vee (x + y - 1). \quad (14)$$

A *fuzzy disjunction* is an increasing mapping $D : [0, 1] \times [0, 1] \rightarrow [0, 1]$ that satisfies $D(0, 0) = 0$ and $D(0, 1) = D(1, 0) = D(1, 1) = 1$. In particular, a commutative and associative fuzzy disjunction $S : [0, 1] \times [0, 1] \rightarrow [0, 1]$ that satisfies $S(0, x) = x$ for every $x \in [0, 1]$ is called *triangular conorm*, for short *s-norm*. The following operators represent *s*-norms:

$$D_M(x, y) = x \vee y, \quad (15)$$

$$D_P(x, y) = x + y - x \cdot y, \quad (16)$$

$$D_L(x, y) = 1 \wedge (x + y). \quad (17)$$

We would like to point out that in the literature of fuzzy logic, one often does not work with the overall class of fuzzy conjunctions and fuzzy disjunction but rather with the restricted class of *t*-norms and *s*-norms [65]. In particular, the FAM models presented in the next sections are based on *t*-norms and *s*-norms except for the FLBAM and the general FMAMs.

An operator $I : [0, 1] \times [0, 1] \rightarrow [0, 1]$ that is decreasing in the first argument and that is increasing in the second argument is called a *fuzzy implication* if I extends the usual crisp implication on $\{0, 1\} \times \{0, 1\}$; i.e., $I(0, 0) = I(0, 1) = I(1, 1) = 1$ and $I(1, 0) = 0$. Some particular fuzzy implications, which were

introduced by Gödel, Goguen, and Lukasiewicz, can be found below [31, 65].

$$I_M(x, y) = \begin{cases} 1 & x \leq y \\ y & x > y, \end{cases} \quad (18)$$

$$I_P(x, y) = \begin{cases} 1 & x \leq y \\ y/x & x > y, \end{cases} \quad (19)$$

$$I_L(x, y) = 1 \wedge (y - x + 1). \quad (20)$$

A fuzzy conjunction C can be associated with a fuzzy disjunction D or with a fuzzy implication I by means of a relationship of duality which can be either negation or adjunction. Specifically, we say that a fuzzy conjunction C and a fuzzy disjunction D are *dual operators with respect to a fuzzy negation N* if and only if the following equation holds for every $x, y \in [0, 1]$:

$$C(x, y) = N(D(N(x), N(y))). \quad (21)$$

In other words, we have that $C(x, \cdot) = D^N(N(x), \cdot)$ for all $x \in [0, 1]$ or, equivalently, $C(\cdot, y) = D^N(\cdot, N(y))$ for all $y \in [0, 1]$.

The following implication holds for fuzzy operators C and D that are dual with respect to N : If C is a dilation for every $x \in [0, 1]$, then D is an erosion for every $x \in [0, 1]$ and vice versa [22]. For example, note that the pairs (C_M, D_M) , (C_P, D_P) , and (C_L, D_L) are dual operators with respect to the standard fuzzy negation N_S . The dual operator of a (continuous) t -norm with respect to N_S is a (continuous) s -norm [65].

In this chapter, we will also consider the duality relationship of adjunction between a fuzzy conjunction C and a fuzzy implication I . We simply say that C and I form an adjunction if and only if $C(x, \cdot)$ and $I(x, \cdot)$ form an adjunction for every $x \in [0, 1]$. In this case, we also call C and I *adjoint operators* and we have that $C(z, \cdot)$ is a dilation and $I(z, \cdot)$ is an erosion for every $z \in [0, 1]$ [31]. Examples of adjunctions are given by the pairs (C_M, I_M) , (C_P, I_P) , and (C_L, I_L) .

The fuzzy operations C , D , and I can be combined with the maximum or the minimum operation to yield the following matrix products. For $A \in [0, 1]^{m \times p}$ and $B \in [0, 1]^{p \times n}$, we define the max- C product $C = A \circ B$ as follows:

$$c_{ij} = \bigvee_{k=1}^p C(a_{ik}, b_{kj}) \quad \forall i = 1, \dots, m, j = 1, \dots, n. \quad (22)$$

Similarly, the min- D product $D = A \bullet B$ and the min- I product $E = A \circledast B$ are given by the following equations:

$$d_{ij} = \bigwedge_{k=1}^p D(a_{ik}, b_{kj}) \quad \forall i = 1, \dots, m, j = 1, \dots, n, \quad (23)$$

$$e_{ij} = \bigwedge_{k=1}^p I(b_{kj}, a_{ik}) \quad \forall i = 1, \dots, m, j = 1, \dots, n. \quad (24)$$

Subscripts of the product symbols \circ , \bullet , or \circledast indicate the type of fuzzy operators used in equations (22)–(24). For example, the symbol \circ_M stands for the max- C product where the fuzzy conjunction C in equation (22) is given by C_M .

33.4 Types of Neurons Used in Fuzzy Associative Memory Models

This section describes the most important types of fuzzy neurons that occur in FAM models. These models of artificial neurons can be formulated in terms of the max- C , min- D , and min- I matrix products that we introduced in Section 33.3.2.

Let us consider an arbitrary model of an artificial neuron. The symbol $\mathbf{x} = [x_1, \dots, x_n]^T$ denotes the fuzzy input vector and y denotes the fuzzy output. The weights $w_i \in [0, 1]$ of the neuron form a vector $\mathbf{w} = [w_1, \dots, w_n]^T$. We use θ to denote the bias. A model without bias is obtained by setting $\theta = 0$ in equations (26) and (27) or by setting $\theta = 1$ in equations (28) and (29).

33.4.1 The Max-C and the Min-I Neuron

One of the most general classes of fuzzy neurons was introduced by Pedrycz [66] in the early 1990s. The neurons of this class are called *aggregative logic neurons* since they realize an aggregation of the inputs and synaptic weights. We are particularly interested in the *OR-neuron* described by the following equation, where S is a s -norm and T is a t -norm.

$$y = \bigvee_{j=1}^n T(w_j, x_j). \quad (25)$$

Let us adapt Pedrycz's original definition by introducing a bias term and by substituting the t -norm with a more general operation of fuzzy conjunction. We obtain a generalized OR-neuron or *S-C neuron*.

$$y = \left[\bigvee_{j=1}^n C(w_j, x_j) \right] s \theta. \quad (26)$$

We refrained from replacing the s -norm by a fuzzy disjunction since associativity and commutativity are required in a neural model. If S equals the maximum operation, we obtain the *max-C neuron* that is given by the following equation where \mathbf{w}^T represents the transpose of \mathbf{w} .

$$y = \left[\bigvee_{j=1}^n C(w_j, x_j) \right] \vee \theta = (\mathbf{w}^T \circ \mathbf{x}) \vee \theta. \quad (27)$$

Particular choices of fuzzy conjunctions yield particular max- C neurons. Given a particular max- C neuron, we will indicate the underlying type of fuzzy conjunction by means of a subscript. For example, $\text{max-}C_M$ will denote the neuron that is based on the minimum fuzzy conjunction. A similar notation will be applied to describe the min- I neuron and the min- D that will be introduced in Section 33.4.2. We define the *min-I neuron* by means of the following equation:

$$y = \left[\bigwedge_{j=1}^n I(x_j, w_j) \right] \wedge \theta = (\mathbf{w}^T \otimes \mathbf{x}) \wedge \theta. \quad (28)$$

We are particularly interested in $\text{min-}I_T$ neurons where I_T denotes a fuzzy implication that forms an adjunction together with a t -norm T . This type of neuron occurs in the FLBAM model [30].

To our knowledge, the max- C neuron represents the most widely used model of fuzzy neuron in FAMs. The FLBAM of Bělohlávek, which consists of min- I neurons, represents an exception to this rule. For example, Kosko's FAM employs max- C_M or max- C_P neurons. Junbo's FAM and the FAM model of Liu are also equipped with max- C_M neurons. The generalized FAM of Chung and Lee as well as the IFAM models employ max- T neurons, where T is a t -norm.

Note that we may speak of a *max-C morphological neuron* if and only if $C(x, \cdot)$ is a dilation for every $x \in [0, 1]$ [31]. Examples of max- C morphological neurons include max- C_M , max- C_P , and max- C_L neurons.

33.4.2 The Min-D Neuron: A Dual Model of the Max-C Neuron

Consider the neural model that is described in terms of the equation below:

$$y = \left[\bigwedge_{j=1}^n D(w_j, x_j) \right] \wedge \theta = (\mathbf{w}^T \bullet \mathbf{x}) \wedge \theta. \quad (29)$$

We refer to neurons of this type as *min-D neurons*. For example, dual IFAMs are equipped with min-*D* neurons [13].

Suppose that C and D are dual operators with respect to N . In this case, the max-*C* neuron and the min-*D* neuron are *dual with respect to N* in the following sense. Let \mathcal{W} denote the function computed by the max-*C* neuron; i.e., $\mathcal{W}(\mathbf{x}) = (\mathbf{w}^T \circ \mathbf{x}) \vee \theta$ for all $\mathbf{x} \in [0, 1]^n$. If m_j denotes $N(w_j)$ and ϑ denotes $N(\theta)$, then we obtain the negation of \mathcal{W} with respect to N as follows:

$$N(\mathcal{W}(N(\mathbf{x}))) = N\left(\left[\bigvee_{j=1}^n C(w_j, N(x_j))\right] \vee \theta\right) \quad (30)$$

$$= \left[\bigwedge_{j=1}^n N(C(w_j, N(x_j))) \right] \wedge N(\theta) = \left[\bigwedge_{j=1}^n D(m_j, x_j) \right] \wedge \vartheta. \quad (31)$$

Note that the dual of a max-*C* morphological neuron with respect to a fuzzy negation N is a min-*D* morphological neuron that performs an erosion.

33.5 Kosko's Fuzzy Associative Memory and Generalizations

Kosko's FAMs constitute one of the earliest attempts to develop neural AM models based on fuzzy set theory. These models were introduced in the early 1990s and are usually referred as *max-min FAM* and *max-product FAM* [4]. Later, Chung and Lee introduced generalizations of Kosko's models that are known as *generalized fuzzy associative memories* (GFAMs) [8]. The models of Kosko and Chung and Lee share the same network topology and Hebbian learning rules with the linear associative memory [67–69]. Thus, these FAM models exhibit a large amount of crosstalk if the inputs do not satisfy a certain orthonormality condition.

33.5.1 The Max-Min and the Max-Product Fuzzy Associative Memories

The max-min and the max-product FAM are both single-layer feedforward ANNs. The max-min FAM is equipped with max- C_M fuzzy neurons, while the max-product FAM is equipped with max- C_P fuzzy neurons. Thus, both models belong to the class of FMAMs. Kosko's original definitions do not include bias terms. Consequently, if $W \in [0, 1]^{m \times n}$ is the synaptic weight matrix of a max-min FAM and if $\mathbf{x} \in [0, 1]^n$ is the input pattern, then the output pattern $\mathbf{y} \in [0, 1]^m$ is computed as follows (cf. equation (22)).

$$\mathbf{y} = W \circ_M \mathbf{x}. \quad (32)$$

Similarly, the max-product FAM produces the output $\mathbf{y} = W \circ_P \mathbf{x}$. Note that both versions of Kosko's FAM perform dilations at each node (and overall). Thus, Kosko's models belong to the class FMAMs.

Consider a set of fundamental memories $\{(\mathbf{x}^\xi, \mathbf{y}^\xi) : \xi = 1, \dots, k\}$. The learning rule used to store the fundamental memory set in a max-min FAM is called *correlation-minimum encoding*. In this learning rule, the synaptic weight matrix is given by the following equation:

$$W = Y \circ_M X^T, \quad (33)$$

where $X = [\mathbf{x}^1, \dots, \mathbf{x}^k] \in [0, 1]^{n \times k}$ and $Y = [\mathbf{y}^1, \dots, \mathbf{y}^k] \in [0, 1]^{m \times k}$. In a similar fashion, the weight matrix of the max–product FAM is synthesized by setting $W = Y \circ_P X^T$. We speak of *correlation-product encoding* in this case.

Both the correlation-minimum and the correlation-product encoding are based on the Hebb's postulate, which states that the synaptic weight change depends on the input as well as the output activation [70]. Unfortunately, Hebbian learning entails an extremely low storage capacity of one input–output pair per FAM matrix in Kosko's models. More precisely, Kosko only succeeded in showing the following proposition concerning the recall of patterns by a max–min and a max–product FAM [4].

Proposition 1. Suppose that a single fundamental memory pair $(\mathbf{x}^1, \mathbf{y}^1)$ was stored in a max–min FAM by means of the correlation-minimum encoding scheme, then $W \circ_M \mathbf{x}^1 = \mathbf{y}^1$ if and only if $\bigvee_{j=1}^n x_j^1 \geq \bigvee_{i=1}^m y_i^1$. Moreover, we have $W \circ_M \mathbf{x} \leq \mathbf{y}^1$ for every $\mathbf{x} \in [0, 1]^n$.

Similarly, if a single fundamental memory pair $(\mathbf{x}^1, \mathbf{y}^1)$ was stored in a max–product FAM by means of the correlation-product encoding scheme, then $W \circ_P \mathbf{x}^1 = \mathbf{y}^1$ if and only if $\bigvee_{j=1}^n x_j^1 = 1$. Furthermore, we have $W \circ_P \mathbf{x} \leq \mathbf{y}^1$ for every $\mathbf{x} \in [0, 1]^n$.

In Section 33.5.2, we will provide conditions for perfect recall using a max–min or max–product FAM that stores several input–output pairs (cf. Proposition 2). Kosko himself proposed to utilize a *FAM system* in order to overcome the storage limitation of the max–min FAM and max–product FAM. Generally speaking, a FAM system consists of a bank of k FAM matrices W^ξ such that each FAM matrix stores a single fundamental memory $(\mathbf{x}^\xi, \mathbf{y}^\xi)$, where $\xi = 1, \dots, k$. Given an input pattern \mathbf{x} , a combination of the outputs of each FAM matrix in terms of a weighted sum yields the output of the system. Kosko argues that the separate storage of FAM associations consumes memory space but provides an ‘audit trail’ of the FAM inference procedure and avoids crosstalk [4]. According to Chung and Lee, the implementation of a FAM system is limited to applications with a small amount of associations [8]. As to computational effort, the FAM system requires the synthesis of at least k FAM matrices.

33.5.2 Generalized Fuzzy Associative Memories of Chung and Lee

Chung and Lee generalized Kosko's FAMs by substituting the max–min or the max–product by a more general max– t product in equations (32) and (33) [8]. The resulting model, called *generalized FAM* (GFAM), can be described in terms of the following relationship between an input pattern $\mathbf{x} \in [0, 1]^n$ and the corresponding output pattern $\mathbf{y} \in [0, 1]^m$. Here, the symbol \circ_T denotes the max– C product (cf. equation (22)) where C is a t -norm.

$$\mathbf{y} = W \circ_T \mathbf{x} \quad \text{where } W = Y \circ_T X^T. \quad (34)$$

We refer to the learning rule that is used to generate $W = Y \circ_T X^T$ as *correlation- t encoding*. Note that a GFAM performs a dilation at each node (and overall) if and only if the t -norm represents a dilation in $[0, 1]$.

We could generalize the GFAM even further by substituting the t -norm with a more general fuzzy conjunction. However, the resulting model does not satisfy Proposition 2 below since it requires the associativity and the boundary condition $T(x, 1) = x$ of a t -norm.

In the theory of linear associative memories trained by means of a learning rule based on Hebb's postulate, perfect recall of the stored patterns is possible if the patterns $\mathbf{x}^1, \dots, \mathbf{x}^k$ constitute an orthonormal set [35, 68]. Chung and Lee noted that a similar statement, which can be found below, is true for GFAM models.

A straightforward fuzzification of the orthogonality and orthonormality concepts leads to the following definitions. Fuzzy patterns $\mathbf{x}, \mathbf{y} \in [0, 1]^n$ are said *max– t orthogonal* if and only if $\mathbf{x}^T \circ_T \mathbf{y} = 0$; i.e., $T(x_j, y_j) = 0$ for all $j = 1, \dots, n$. Consequently, we speak of a *max– t orthonormal* set $\{\mathbf{x}^1, \dots, \mathbf{x}^k\}$ if and only if the patterns \mathbf{x}^ξ and \mathbf{x}^η are max– t orthogonal for every $\xi \neq \eta$ and \mathbf{x}^ξ is a normal fuzzy set

for every $\xi = 1, \dots, k$. Recall that a fuzzy set $\mathbf{x} \in [0, 1]^n$ is normal if and only if $\sum_{j=1}^n x_j = 1$; i.e., $\mathbf{x}^T \circ_T \mathbf{x} = 1$.

Based on the max- t orthonormality definition, Chung and Lee succeeded in showing the following proposition concerning the recall of patterns by a GFAM [8].

Proposition 2. *Suppose that the fundamental memories $(\mathbf{x}^\xi, \mathbf{y}^\xi)$, for $\xi = 1, \dots, k$, are stored in a GFAM by means of the correlation- t encoding scheme. If the set $\{\mathbf{x}^1, \dots, \mathbf{x}^k\}$ is max- t orthonormal, then $W \circ_T \mathbf{x}^\xi = \mathbf{y}^\xi$ for every $\xi = 1, \dots, k$.*

In particular, Chung and Lee noted that the Lukasiewicz GFAM, i.e., the GFAM based on the Lukasiewicz fuzzy conjunction, will perfectly recall the stored patterns if the \mathbf{x}^ξ 's are such that $0 \vee (\mathbf{x}_j^\xi + \mathbf{x}_j^\eta - 1) = 0$ for every $\xi \neq \eta$ and $j = 1, \dots, n$. In other words, we have $\mathbf{y}^\xi = W \circ_L \mathbf{x}^\xi$ for every $\xi = 1, \dots, k$ if $\mathbf{x}_j^\xi + \mathbf{x}_j^\eta \leq 1$ for every $\xi \neq \eta$ and $j = 1, \dots, n$. These inequalities hold true, in particular, for patterns \mathbf{x}^ξ 's that satisfy the usual condition $\sum_{j=1}^k x_j^\xi = 1$ for every $j = 1, \dots, n$.

33.6 Variations of the Max–Min Fuzzy Associative Memory

In this section, we will discuss two variations of Kosko's max–min FAM: the models of Junbo and Liu. Junbo *et al.* generate the weight matrix of their model according to the Gödel implicative learning scheme that we will introduce in equation (35). Liu modified Junbo's model by incorporating a threshold at the input and output layer.

33.6.1 Junbo's Fuzzy Associative Memory Model

Junbo's FAM and Kosko's max–min FAM share the same topology and the same type of morphological neurons, namely, max- C_M neurons [6]. Consequently, Junbo's FAM computes the output pattern $\mathbf{y} = W \circ_M \mathbf{x}$ on presentation of an input pattern $\mathbf{x} \in [0, 1]^n$.

The difference between the max–min FAM and Junbo's FAM lies in the learning rule. Junbo *et al.* chose to introduce a new learning rule for FAM which allows for the storage of multiple fuzzy fundamental memories. The synaptic weight matrix is computed as follows:

$$W = Y \circledast_M X^T. \quad (35)$$

Here, the symbol \circledast_M denotes the min- I_M product of equation (24). We will refer to this learning rule as *Gödel implicative learning* since it employs Gödel's fuzzy implication I_M [12, 13].

The following proposition shows the optimality (in terms of the perfect recall of the original patterns) of the Gödel implicative learning scheme for max–min FAMs [6, 71]. In particular, Proposition 3 reveals that Junbo's FAM can store at least as many patterns as the max–min FAM of Kosko.

Proposition 3. *Let $X = [\mathbf{x}^1, \dots, \mathbf{x}^k] \in [0, 1]^{n \times k}$ and $Y = [\mathbf{y}^1, \dots, \mathbf{y}^k] \in [0, 1]^{m \times k}$ be the matrices whose columns are the fundamental memories. If there exist $A \in [0, 1]^{m \times n}$ such that $A \circ_M \mathbf{x}^\xi = \mathbf{y}^\xi$ for all $\xi = 1, \dots, k$, then $W = Y \circledast_M X^T$ is such that $W \circ_M \mathbf{x}^\xi = \mathbf{y}^\xi$ for all $\xi = 1, \dots, k$.*

33.6.2 The Max–Min Fuzzy Associative Memory with Threshold of Liu

Proposition 3 shows that Gödel implicative learning guarantees the best possible storage capacity for a max–min FAM. Therefore, improvements in storage capacity can only be achieved by considering neural associative memories with a different architecture and/or different types of neurons. Since adding hidden layers to the max–min FAM also fails to increase the storage capacity, Liu proposes the following model

whose recall phase is described by the following equation [9]:

$$\mathbf{y} = (W \circ_M (\mathbf{x} \vee \mathbf{c})) \vee \mathbf{d}. \quad (36)$$

The weight matrix $W \in [0, 1]^{m \times n}$ is given in terms of Gödel implicative learning and the thresholds $\mathbf{d} \in [0, 1]^m$ and $\mathbf{c} = [c_1, \dots, c_n]^T \in [0, 1]^n$ are of the following form:

$$\mathbf{d} = \bigwedge_{\xi=1}^k \mathbf{y}^\xi \quad \text{and} \quad c_j = \begin{cases} \bigwedge_{i \in D_j} \bigwedge_{\xi \in LE_{ij}} y_i^\xi & \text{if } D_j \neq \emptyset \\ 0 & \text{if } D_j = \emptyset, \end{cases} \quad (37)$$

where $LE_{ij} = \{\xi : x_j^\xi \leq y_i^\xi\}$ and $D_j = \{i : LE_{ij} \neq \emptyset\}$.

Liu's model is also known as the *max-min FAM with threshold*. Note that equation (36) boils down to adding bias terms to the single-layer max–min FAM. The following proposition concerns the recall of patterns using the max–min FAM with threshold.

Proposition 4. Suppose the symbols W , \mathbf{c} , and \mathbf{d} denote the weight matrix and the thresholds of a max–min FAM with threshold that stores the fundamental memories $(\mathbf{x}^\xi, \mathbf{y}^\xi)$, where $\xi = 1, \dots, k$. If there exists $A \in [0, 1]^{m \times n}$ such that $A \circ_M \mathbf{x}^\xi = \mathbf{y}^\xi$ for all $\xi = 1, \dots, k$, then $\mathbf{y}^\xi = (W \circ (\mathbf{x}^\xi \vee \mathbf{c})) \vee \mathbf{d}$ for all $\xi = 1, \dots, k$.

Thus, the max–min FAM with threshold can store at least as many patterns as the FAM of Junbo and the max–min FAM of Kosko. In the next section, we will introduce a FAM model whose storage capacity is at least as high as that of Liu's model and which does not require the cumbersome computation of the threshold $\mathbf{c} \in [0, 1]^n$.

33.7 Other Subclasses of Fuzzy Morphological Associative Memories

In this section, we discuss the IFAM, the dual IFAM, and FLBAM models. The IFAMs and the dual IFAMs can be viewed as extensions of MAMs to the fuzzy domain [12, 13, 15]. Thus, these models maintain the features of the MAM models. In particular, an IFAM model computes a dilation whereas a dual IFAM performs an erosion and the FLBAM model computes an anti-dilation at every node [30].

33.7.1 Implicative Fuzzy Associative Memories

IFAMs bear some resemblance with the GFAM model of Chung and Lee. Specifically, an IFAM model is given by a single-layer feedforward ANN endowed with max- T neurons where T is a continuous t -norm. In contrast to the GFAM, the IFAM model includes a bias term $\theta = [0, 1]^n$ and employs a learning rule that we call *R-implicative fuzzy learning*. Note that a continuous t -norm represents a dilation in $[0, 1]$. Therefore, the neurons of an IFAM are dilative and thus IFAMs belong to the class of FMAMs.

Consider a fundamental memory set $\{(\mathbf{x}^\xi, \mathbf{y}^\xi) : \xi = 1, \dots, k\}$ and an IFAM model that is equipped with max- T neurons. Let I_T be the fuzzy implication such that I_T and the given continuous t -norm T are adjoint and let the symbol \circledast_T denote the min- I_T product (cf. equation (24)). Given an input pattern $\mathbf{x} \in [0, 1]^n$, the IFAM model produces the following output pattern $\mathbf{y} \in [0, 1]^m$:

$$\mathbf{y} = (W \circ_T \mathbf{x}) \vee \theta \quad \text{where } W = Y \circledast_T X^T \text{ and } \theta = \bigwedge_{\xi=1}^k \mathbf{y}^\xi. \quad (38)$$

The fuzzy implication I_T is uniquely determined by the following equation:

$$I_T(x, y) = \bigvee \{z \in [0, 1] : T(x, z) \leq y\} \quad \forall x, y \in [0, 1]. \quad (39)$$

We refer to I_T as the *R-implication associated with the t-norm T*, hence the name R-implicative fuzzy learning. Particular choices of T and I_T , respectively, lead to particular IFAM models. The name of a particular IFAM model indicates the choice of T and I_T . For example, the Gödel IFAM corresponds to the IFAM model given by the equation $\mathbf{y} = (W \circ_M \mathbf{x}) \vee \theta$, where $W = Y \otimes_M X^T$ and $\theta = \bigwedge_{\xi=1}^k \mathbf{y}^\xi$.

Note that the learning rule used in the Gödel IFAM model coincides with the Gödel implicative learning rule that is used in the FAM models of Junbo and Liu. Recall that Liu's max-min FAM with threshold can be viewed as an improved version of Junbo's FAM. Although the Gödel IFAM disposes of only one threshold term θ , its storage capacity is at least as high as the one of Liu's FAM [13]. In fact, the IFAM model can be considered a generalization of Liu's max-min FAM with threshold. The following proposition concerns the recall of patterns using an arbitrary IFAM model [13].

Proposition 5. *Consider the fundamental memory set $\{(\mathbf{x}^\xi, \mathbf{y}^\xi) : \xi = 1, \dots, k\}$. If there exist a synaptic weight matrix $A \in [0, 1]^{m \times n}$ and a bias vector $\beta \in [0, 1]^m$ such that $\mathbf{y}^\xi = (A \circ_T \mathbf{x}^\xi) \vee \beta$, for every $\xi = 1, \dots, k$, then $A \leq W = Y \otimes_T X^T$, $\beta \leq \theta = \bigwedge_{\xi=1}^k \mathbf{y}^\xi$, and $\mathbf{y}^\xi = (W \circ_T \mathbf{x}^\xi) \vee \theta$ for all $\xi = 1, \dots, k$.*

In the autoassociative case, we speak of the *autoassociative fuzzy implicative memory* (AFIM). The synaptic weight matrix and bias vector of an AFIM model are given by $W = X \otimes_T X^T$ and $\theta = \bigwedge_{\xi=1}^k \mathbf{x}^\xi$, respectively. We can convert the AFIM into a dynamic model by feeding the output $(W \circ_T \mathbf{x}) \vee \theta$ back into the memory. We refer to the patterns $\mathbf{x} \in [0, 1]^n$ that remain fixed under an application of $W = X \otimes_T X^T$ as the fixed points of W . In sharp contrast to the GFAM models, one can store as many patterns as desired in an AFIM [12]. In particular, the storage capacity of the AFIM is at least as high as the storage capacity of the quantum associative memory if the stored patterns are binary [25]. The following proposition characterizes the fixed points of an AFIM as well as the output patterns in terms of the fixed points [13].

Proposition 6. *Consider a fundamental memory set $\{\mathbf{x}^1, \dots, \mathbf{x}^k\}$. If $W = X \otimes_T X^T$ and $\theta = \bigwedge_{\xi=1}^k \mathbf{x}^\xi$, then for every input pattern $\mathbf{x} \in [0, 1]^n$, the output $(W \circ_T \mathbf{x}) \vee \theta$ of the AFIM is the supremum of \mathbf{x} in the set of fixed points of W greater than θ ; i.e., $(W \circ_T \mathbf{x}) \vee \theta$ is the smallest fixed point \mathbf{y} of W such that $\mathbf{y} \geq \mathbf{x}$ and $\mathbf{y} \geq \theta$.*

Moreover, a pattern $\mathbf{y} \in [0, 1]^n$ is a fixed point of W if $\mathbf{y} = \mathbf{c}$ for some constant vector $\mathbf{c} = [c, c, \dots, c]^T \in [0, 1]^n$ or if \mathbf{y} is of the following form for some $L_i \subseteq \{1, \dots, k\}$ and some $k \in \mathbb{N}$.

$$\mathbf{y} = \bigvee_{l=1}^k \bigwedge_{\xi \in L_l} \mathbf{x}^\xi. \quad (40)$$

This proposition reveals that AFIM models exhibit a very large number of fixed points, which include the original patterns \mathbf{x}^ξ , where $\xi = 1, \dots, k$, and many spurious states. Moreover, the basin of attraction of an original pattern \mathbf{x}^ξ only consists of patterns \mathbf{x} such that $\mathbf{x} \leq \mathbf{x}^\xi$. In the near future, we intend to generalize Proposition 6 to include the heteroassociative case.

IFAM models have been successfully applied to several problems in prediction where they have outperformed other models such as statistical models and the FAM models that we mentioned above [13, 28]. The Lukasiewicz IFAM which exhibited the best performance in these simulations is closely related to the gray-scale MAM [14, 15]. Both the MAM model and the general IFAM model are equipped with a dual model.

33.7.2 Dual Implicative Fuzzy Associative Memories

Recall that the IFAM model has dilative max-T neurons where T is a continuous t-norm. A dual IFAM model can be constructed by taking the dual neurons with respect to a certain fuzzy negation. We chose to consider only the standard fuzzy negation N_S .

Let us derive the dual model of a given IFAM. Suppose that we want to store the associations $(\mathbf{x}^\xi, \mathbf{y}^\xi)$, where $\xi = 1, \dots, k$, in a dual IFAM. Let us synthesize the weight matrix \bar{W} and the bias vector $\bar{\theta}$ of the IFAM using the fundamental memories $(N_S(\mathbf{x}^\xi), N_S(\mathbf{y}^\xi))$, where $\xi = 1, \dots, k$. If M denotes $N_S(\bar{W})$ and if ϑ denotes $N_S(\bar{\theta})$, then an application of equations (30) and (31) to equation (38) yields the recall phase of the dual IFAM model [12, 13]:

$$\mathbf{y} = N_S [(\bar{W} \circ_T N_S(\mathbf{x})) \vee \bar{\theta}] = (M \bullet_S \mathbf{x}) \wedge \vartheta. \quad (41)$$

Here, the symbol \bullet_S stands for the min- S product of equation (23) based on the continuous s -norm S that is the dual operator of T with respect to N_S . We conclude that the dual IFAM model performs an erosion at each node.

In view of equations (11) and (41), every statement concerning the IFAM model yields a corresponding dual statement concerning the dual IFAM model. Specifically, we obtain the corresponding dual statement from the statement about the IFAM model by replacing minimum with maximum, t -norm with s -norm, the product \circ_T with \bullet_S , and vice versa [13].

33.7.3 Fuzzy Logical Bidirectional Associative Memory

The FLBAM [30] constitutes a recurrent model whose network topology coincides with the one of Kosko's bidirectional associative memory (BAM) [72]. In contrast to Kosko's BAM, the neurons of the FLBAM calculate min- I_T products where the fuzzy implication I_T is adjoint to some continuous t -norm T . Using a Hebbian-style correlation-t encoding scheme, Bélohlávek constructs the weight matrix W for the forward direction of the FLBAM as follows:

$$W = Y \circ_T X^T. \quad (42)$$

The weight matrix for the backward direction simply corresponds to W^T . Thus, given an input pattern $\mathbf{x}_0 \in [0, 1]^n$, the FLBAM generates the following sequence $(\mathbf{x}_0, \mathbf{y}_0), (\mathbf{x}_1, \mathbf{y}_0), (\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_1), \dots$:

$$\mathbf{y}_k = W \circ_T \mathbf{x}_k \quad \text{and} \quad \mathbf{x}_{k+1} = W^T \circ_T \mathbf{y}_k \quad \text{for } k = 0, 1, 2, \dots. \quad (43)$$

The following proposition shows that an FLBAM reaches a stable state after one step in the forward direction and one step in the backward direction [30].

Proposition 7. *For an arbitrary input pattern $\mathbf{x}_0 \in [0, 1]^n$, the pair $(\mathbf{x}_1, \mathbf{y}_0)$ is a stable state of the FLBAM.*

The following observations demonstrate that the FLBAM models belong to the FMAM class [73]. Specifically, we will show that the neurons of an FLBAM compute anti-dilations. Recall that the FLBAM has min- I_T neurons, where I_T is adjoint to some continuous t -norm T . The fact that I_T and T form an adjunction implies that I_T can be expressed in the form given by equation (39). Therefore, the following equations hold true for every $X \subseteq [0, 1]$ and $y \in [0, 1]$:

$$I_T \left(\bigvee X, y \right) = \bigvee \left\{ z \in [0, 1] : T \left(\bigvee X, z \right) \leq y \right\} \quad (44)$$

$$= \bigvee \left\{ z \in [0, 1] : \bigvee_{x \in X} T(x, z) \leq y \right\} \quad (45)$$

$$= \bigwedge_{x \in X} \left[\bigvee \{z \in [0, 1] : T(x, z) \leq y\} \right] = \bigwedge_{x \in X} I_T(x, y). \quad (46)$$

Consequently, $I_T(\cdot, y)$ represents an anti-dilation for every $y \in [0, 1]$ and therefore the nodes of an FLBAM also calculate anti-dilations.



Figure 33.1 Fundamental memory set $\{x^1, \dots, x^{12}\}$ used in Section 33.8.1

33.8 Experimental Results

33.8.1 Storage Capacity and Noise Tolerance Example

Consider the 12 patterns shown in Figure 33.1. These are grayscale images $x^\xi \in [0, 1]^{56 \times 46}$, $\xi = 1, \dots, 12$, from the faces database of AT&T Laboratories Cambridge [74]. This database contains files in PGM format. The size of each image is 92×112 pixels, with 256 gray levels per pixel. We downsized the original images using neighbor interpolation. Then, we obtained fuzzy patterns (vectors) $x^1, \dots, x^{12} \in [0, 1]^{2576}$ using the standard row-scan method.

We stored the patterns x^1, \dots, x^k in the Lukasiewicz, Gödel, and Goguen AFIMs and we verified that they represent fixed points of these models; i.e., the AFIMs succeeded in storing the fundamental memory set. In order to verify the tolerance of the AFIM model with respect to corrupted or incomplete patterns, we used the images displayed in Figure 33.2 as input pattern. The first three patterns, r^1 , r^2 , and r^3 , of Figure 33.2 were generated by introducing pepper noise in x^1 with probabilities 25, 50, and 75%, respectively. The other three patterns, r^4 , r^5 , and r^6 , were obtained excluding, respectively, 25, 50, and 75% of the original image. The corresponding recalled patterns are shown in Figure 33.3. Note that the Lukasiewicz AFIM succeeded in recalling the original pattern almost perfectly.

We also conducted the same experiment using the FAM models presented previously. We observed that the max-min and max-product FAMs, as well as the Lukasiewicz GFAM and the FLBAMs based on the implication of Lukasiewicz and Gödel, failed to demonstrate an adequate performance on this task due to a high amount of crosstalk between the stored patterns. Moreover, we noted that the FAM of Junbo

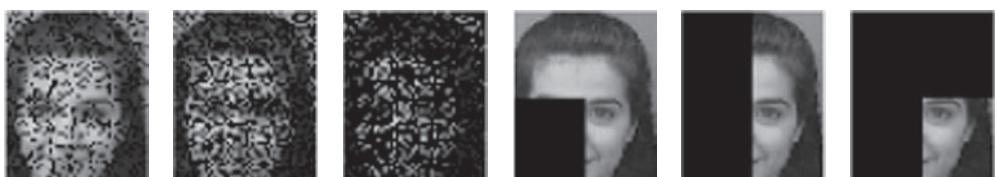


Figure 33.2 Patterns r^1, \dots, r^6 representing corrupted or incomplete versions of pattern x^1 used as input of the FMAM models



Figure 33.3 Patterns recalled by Lukasiewicz (first row), Gödel (second row), and Goguen (third row) when the patterns $\mathbf{r}^1, \dots, \mathbf{r}^6$ of Figure 33.2 are presented as input

and the max–min FAM with threshold produced the same outputs as the Gödel AFIM. The dual AFIMs succeeded in storing the fundamental memory set but failed to recall \mathbf{x}^1 when the patterns of Figure 33.2 were presented as input. In fact, concerning a dual AFIM model, we can show that the basin of attraction of an original pattern \mathbf{x}^ξ consists of patterns \mathbf{x} such that $\mathbf{x} \leq \mathbf{x}^\xi$; i.e., the dual AFIM exhibits tolerance with respect to corrupted patterns $\tilde{\mathbf{x}}^\xi$ only if $\tilde{\mathbf{x}}^\xi \geq \mathbf{x}^\xi$ [13]. Table 33.1 presents the normalized error produced by the FAM models when the incomplete or corrupted patterns of Figure 33.2 are presented as input. For instance, the normalized error $E(\mathbf{r}^\eta)$ of the AFIM models are computed as follows for $\eta = 1, \dots, 6$:

$$E(\mathbf{r}^\eta) = \frac{\|\mathbf{x}^1 - [(W \circ_T \mathbf{r}^\eta) \vee \theta]\|}{\|\mathbf{x}^1\|}. \quad (47)$$

Table 33.1 Normalized error produced by the FAM models when the patterns $\mathbf{r}^1, \dots, \mathbf{r}^6$ of Figure 33.2 are presented as input

Associative memory	$E(\mathbf{r}^1)$	$E(\mathbf{r}^2)$	$E(\mathbf{r}^3)$	$E(\mathbf{r}^4)$	$E(\mathbf{r}^5)$	$E(\mathbf{r}^6)$
Lukasiewicz IFAM	0.0248	0.0388	0.0587	0.0842	0.1051	0.1499
Gödel IFAM	0.1132	0.1690	0.2440	0.1878	0.2419	0.3355
Goguen IFAM	0.0459	0.0682	0.1079	0.1241	0.1498	0.2356
max–min FAM of Kosko	0.8331	0.8309	0.8332	0.8257	0.8066	0.7555
max–prod FAM of Kosko	0.4730	0.4723	0.4706	0.4729	0.4736	0.4565
Lukasiewicz GFAM	0.5298	0.4944	0.4932	0.5061	0.6200	0.7013
FAM of Junbo	0.1132	0.1690	0.2440	0.1878	0.2419	0.3355
max–min FAM with threshold	0.1132	0.1690	0.2440	0.1878	0.2419	0.3355
Lukasiewicz FLBAM	0.3512	0.3667	0.3827	0.3932	0.4254	0.4574
Gödel FLBAM	0.2954	0.3156	0.3277	0.2994	0.3111	0.4982

33.8.2 Application of the Lukasiewicz IFAM in Prediction

FAMs can be used to implement mappings of fuzzy rules. In this case, a set of rules in the form of human-like IF–THEN conditional statements are stored. In this section, we present an application of a certain FMAM model to a problem of forecasting time series. Specifically, we applied the Lukasiewicz IFAM to the problem of forecasting the average monthly streamflow of a large hydroelectric plant called Furnas, which is located in southeastern Brazil. This problem was previously discussed in [73, 75, 76].

First, the seasonality of the monthly streamflow suggests the use of 12 different predictor models, one for each month of the year. Let s_ξ , for $\xi = 1, \dots, q$, be samples of a seasonal streamflow time series. The goal is to estimate the value of s_γ from a subsequence of $(s_1, s_2, \dots, s_{\gamma-1})$. Here, we employ subsequences that correspond to a vector of the form

$$\mathbf{p}^\gamma = (s_{\gamma-h}, \dots, s_{\gamma-1})^T, \quad (48)$$

where $h \in \{1, 2, \dots, \gamma - 1\}$. In this experiment, our IFAM-based model uses only a fixed number of three antecedents. For example, the values of January, February, and March were taken into account to predict the streamflow of April.

The uncertainty that is inherent in hydrological data suggests the use of fuzzy sets to model the streamflow samples. For $\xi < \gamma$, a fuzzification of \mathbf{p}^ξ and s_ξ using Gaussian membership functions yields fuzzy sets $\mathbf{x}^\xi : \mathcal{U} \rightarrow [0, 1]$ and $\mathbf{y}^\xi : \mathcal{V} \rightarrow [0, 1]$, respectively, where \mathcal{U} and \mathcal{V} represent finite universes of discourse. A subset S of the resulting input–output pairs $\{(\mathbf{x}^\xi, \mathbf{y}^\xi), \xi < q\}$ is implicitly stored in the Lukasiewicz IFAM [73]. (We construct only those parts of the weight matrix that are actually used in the recall phase.) We employed the *subtractive clustering method* to determine the set S [77]. Feeding the pattern \mathbf{x}^γ into the IFAM model, we retrieve the corresponding output pattern \mathbf{y}^γ . For computational reasons, \mathbf{x}^γ is modeled as a singleton on \mathcal{U} . A defuzzification of \mathbf{y}^γ using the mean of maximum yields s_γ [73].

Figure 33.4 shows the forecasted streamflows estimated by the prediction model based on the Lukasiewicz IFAM for the Furnas reservoir from 1991 to 1998. Table 33.2 compares the errors that

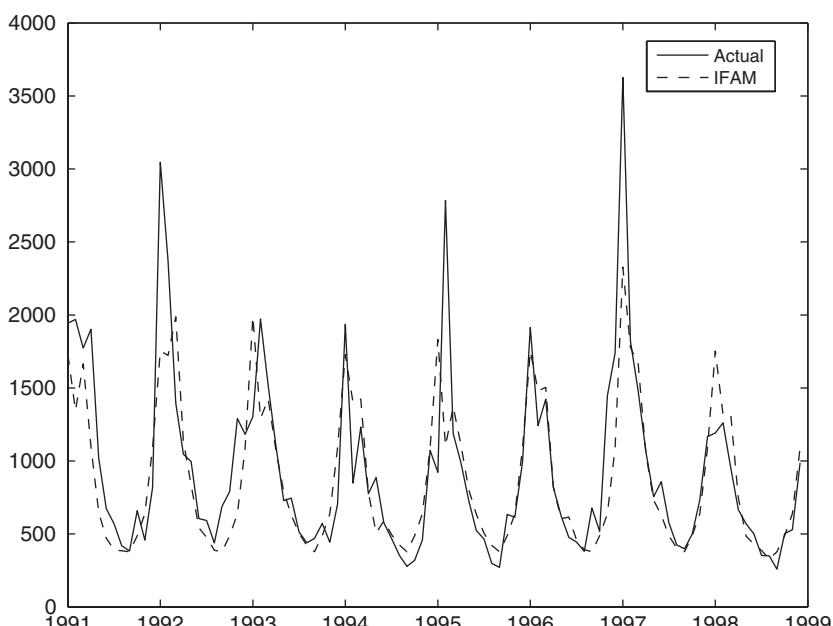


Figure 33.4 The streamflow prediction for the Furnas reservoir from 1991 to 1998. The continuous line corresponds to the actual values and the dashed line corresponds to the predicted values

Table 33.2 Mean square, mean absolute, and mean relative percentage errors produced by the prediction models

Methods	MSE ($\times 10^5$)	MAE (m^3/s)	MPE (%)
Lukasiewicz IFAM	1.42	226	22
PARMA	1.85	280	28
MLP	1.82	271	30
NFN	1.73	234	20
FPM-PRP	1.20	200	18

were generated by the IFAM model and several other models [75, 76]. In contrast to the IFAM-based model, the MLP, NFN, and FPM-PRP models were initialized by optimizing the number of the parameters for each monthly prediction. For example, the MLP considers four antecedents to predict the streamflow of January and three antecedents to predict the streamflow for February. Moreover, the FPM-PRP model also takes into account slope information, which requires some additional ‘fine-tuning.’ We experimentally determined a variable number of parameters (including slopes) for the IFAM model such that $MSE = 0.88 \times 10^5$, $MAE = 157$, and $MPE = 15$.

33.9 Conclusion and Suggestions for Further Research: Fuzzy and Granular Morphological Associative Memories

This chapter describes the most widely known models of FAM from the perspective of MM. We showed that most FAM models compute an elementary operation of MM at each node. Therefore, these models belong to the class of FMAMs. Although a general theory of FMAMs has yet to be developed, a number of useful theoretical results have already been proved for a large subclass of FMAMs called IFAMs [13]. In addition, certain FMAM models such as the Lukasiewicz IFAM have outperformed other FAM models in applications as fuzzy rule-based systems [28].

The mathematical basis for fuzzy (morphological) associative memories can be found in fuzzy mathematical morphology which relies on the fact that the set $[0, 1]^X$ represents a complete lattice for any universe X [21, 53]. Recall that a fuzzy set represents a special case of an information granule, a concept that also encompasses intervals, rough sets, probability distributions, and fuzzy interval numbers [55, 78]. Information granules have been used in a variety of applications [34] but – to our knowledge – granular associative memories have yet to be formulated and investigated. We believe that the complete lattice framework of mathematical morphology may prove to be useful for developing a general theory and applications of FMAMs and granular (morphological) associative memories.

Acknowledgments

This work was supported in part by CNPq under grants nos. 142196/03-7, 303362/03-0, and 306040/06-9, and by FAPESP under grant no. 2006/06818-1.

References

- [1] R. Fuller. *Introduction to Neuro-Fuzzy Systems*. Springer-Verlag New York, March 2000.
- [2] J.J. Buckley and Y. Hayashi. Fuzzy neural networks: A survey. *Fuzzy Sets Syst.* 66 (1994) 1–13.
- [3] S.-G Kong and B. Kosko. Adaptive fuzzy systems for backing up a truck-and-trailer. *IEEE Trans. Neural Netw.* 3 (1992) 211–223.

- [4] B. Kosko. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice Hall, Englewood Cliffs, NJ, 1992.
- [5] T.D. Ndousse. Fuzzy neural control of voice cells in ATM networks. *IEEE J. Sel. Areas Commun.* 12 (1994) 1488–1494.
- [6] F. Junbo, J. Fan, and S. Yan. A learning rule for fuzzy associative memories. In *Proc. IEEE Int. Joint Conf. Neural Netw.* 7 (1994) 4273–4277.
- [7] A. Blanco, M. Delgado, and I. Requena. Identification of fuzzy relational equations by fuzzy neural networks. *Fuzzy Sets Syst.* 71 (1995) 215–226.
- [8] F. Chung and T. Lee. On fuzzy associative memory with multiple-rule storage capacity. *IEEE Trans. Fuzzy Syst.* 4 (1996) 375–384.
- [9] P. Liu. The fuzzy associative memory of max–min fuzzy neural networks with threshold. *Fuzzy Sets Syst.* 107 (1999) 147–157.
- [10] Q. Cheng and Z.-T. Fan. The stability problem for fuzzy bidirectional associative memories. *Fuzzy Sets Syst.* 132 (2002) 83–90.
- [11] F. Junbo, J. Fan, and S. Yan. An encoding rule of fuzzy associative memories. In: *Proceedings of ICCS/ISITA 1992*, Singapore, Vol. 3, 1992, pp. 1415–1418.
- [12] M.E. Valle, P. Sussner, and F. Gomide. Introduction to implicative fuzzy associative memories. In: *Proceedings of the IEEE International Joint Conference on Neural Networks*, Budapest, Hungary, 2004, pp. 925–931.
- [13] P. Sussner and M.E. Valle. Implicative fuzzy associative memories. *IEEE Trans. Fuzzy Syst.* 14 (2006) 793–807.
- [14] G.X. Ritter, P. Sussner, and J.L. Diaz de Leon. Morphological associative memories. *IEEE Trans. Neural Netw.* 9 (1998) 281–293.
- [15] P. Sussner and M.E. Valle. Grayscale morphological associative memories. *IEEE Trans. Neural Netw.* 17 (2006) 559–570.
- [16] P. Sussner. Associative morphological memories based on variations of the kernel and dual kernel methods. *Neural Netw.* 16 (2003) 625–632.
- [17] G.X. Ritter and P. Sussner. An introduction to morphological neural networks. In: *Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, Austria, 1996, pp. 709–717.
- [18] G.X. Ritter and G. Urcid. Lattice algebra approach to single-neuron computation. *IEEE Trans. Neural Netw.* 14 (2003) 282–295.
- [19] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.
- [20] J. Serra. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. Academic Press, New York, 1988.
- [21] C. Ronse. Why mathematical morphology needs complete lattices. *Signal Process.* 21 (1990) 129–154.
- [22] H.J.A.M. Heijmans. *Morphological Image Operators*. Academic Press, New York, 1994.
- [23] P. Sussner. Generalizing operations of binary morphological autoassociative memories using fuzzy set theory. *J. Math. Imaging Vis.* 9 (2003) 81–93. Special issue on Morphological Neural Networks.
- [24] P. Sussner. Fixed points of autoassociative morphological memories. In: *Proceedings of the International Joint Conference on Neural Networks*, Como, Italy, 2000, pp. 611–616.
- [25] D. Ventura and T.R. Martinez. Quantum associative memory. *Inf. Sci.* 124 (2000) 273–296.
- [26] B. Raducanu, M. Graña, and X.F. Albizuri. Morphological scale spaces and associative morphological memories: Results on robustness and practical applications. *J. Math. Imaging Vis.* 19 (2003) 113–131.
- [27] M. Graña, J. Gallego, F.J. Torrealdea, and A. D’Anjou. On the application of associative morphological memories to hyperspectral image analysis. *Lect. Notes Comput. Sci.* 2687 (2003) 567–574.
- [28] P. Sussner and M.E. Valle. Recall of patterns using morphological and certain fuzzy morphological associative memories. In: *Proceedings of the IEEE World Conference on Computational Intelligence 2006*, Vancouver, Canada, 2006.
- [29] P. Sussner and M.E. Valle. Morphological and certain fuzzy morphological associative memories for classification and prediction. In: V.G. Kaburlasos and G.X. Ritter (eds), *Computational Intelligence Based on Lattice Theory*. Springer-Verlag, Heidelberg, Germany, 2007.
- [30] R. Belohlávek. Fuzzy logical bidirectional associative memory. *Inf. Sci.* 128 (2000) 91–103.
- [31] T.Q. Deng and H.J.A.M. Heijmans. Grey-scale morphology based on fuzzy logic. *J. Math. Imag. Vis.* 16 (2002) 155–171.
- [32] P. Sussner and M.E. Valle. A brief account of the relations between grayscale mathematical morphologies. In: *Proceedings of SIBGRAPI 2005*, Natal, RN, Brazil, 2005, pp. 79–86.
- [33] P. Sussner and M.E. Valle. Classification of fuzzy mathematical morphologies based on concepts of inclusion measure and duality. *J. Math. Imaging Vis.* 2008, accepted for publication.
- [34] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer Academic Publishers, Hingham, MA, 2003.

- [35] M.H. Hassoun. Dynamic associative neural memories. In: M.H. Hassoun (ed.), *Associative Neural Memories: Theory and Implementation*. Oxford University Press, Oxford, 1993.
- [36] R. Cuninghame-Green. *Minimax Algebra: Lecture Notes in Economics and Mathematical Systems 166*. Springer-Verlag, New York, 1979.
- [37] J.L. Davidson and F. Hummer. Morphology neural networks: An introduction with applications. *Circuits Syst. Signal Process.* 12 (1993) 177–210.
- [38] P.D. Gader, Y. Won, and M.A. Khabou. Image algebra network for pattern recognition. In: *Image Algebra and Morphological Image Processing V, Vol. 2300 of Proceedings of SPIE*, 1994, pp. 157–168.
- [39] R.A. Araújo, F. Madeiro, R.P. Sousa, L.F.C. Pessoa, and T.A.E. Ferreira. An evolutionary morphological approach for financial time series forecasting. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, Vancouver, Canada, 2006, pp. 2467–2474.
- [40] R.A. Araújo, F. Madeiro, R.P. Sousa, and L.F.C. Pessoa. Modular morphological neural network training via adaptive genetic algorithm for designing translation invariant operators. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, 2006, pp. 873–876.
- [41] C. Koch and T. Poggio. Multiplying with synapses and neurons. In: T. McKenna, J. Davis, and S.F. Zornetzer (eds), *Single Neuron Computation*. Academic Press Professional, Inc, San Diego, CA, 1992.
- [42] P. Soille. *Morphological Image Analysis*. Springer-Verlag, Berlin, 1999.
- [43] S.R. Sternberg. Grayscale morphology. *Comput. Vis. Graph. Image Process.* 35 (1986) 333–355.
- [44] G.X. Ritter and J.N. Wilson. *Handbook of Computer Vision Algorithms in Image Algebra*, 2nd ed. CRC Press, Boca Raton, FL, 2001.
- [45] G.X. Ritter, J. N. Wilson, and J.L. Davidson. Image algebra: An overview. *Comput. Vis. Graph. Image Process.* 49 (1990) 297–331.
- [46] A.J. Yu, M.A. Giese, and T. Poggio. Biophysically plausible implementations of the maximum operation. *Neural Comput.* 14 (2002) 2857–2881.
- [47] M.A. Giese. Neural field model for the recognition of biological motion. In: *Proceedings of the Second International ICSC Symposium on Neural Computation*, Berlin, Germany, 2000.
- [48] M.K. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neurosci.* 2 (1999) 1019–1025.
- [49] N.M. Grzywacz and A.L. Yuille. A model for the estimate of local image velocity by cells in the visual cortex. *Proc. R. Soc.* 239 (1990) 129–161.
- [50] K.N. Gurney. Information processing in dendrites I: Input pattern generalisation. *Neural Net.* 14 (2001) 991–1004.
- [51] I. Segev. Dendritic processing. In: M.A. Arbib (ed.), *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, MA, 1998.
- [52] G.M. Shepherd and R.K. Brayton. Logic operations are properties of computer-simulated interaction between excitable dendritic spines. *Neuroscience* 21 (1987) 151–165.
- [53] M. Nachtegael and E.E. Kerre. Connections between binary, grayscale and fuzzy mathematical morphologies. *Fuzzy Sets Syst.* 124 (2001) 73–85.
- [54] L.A. Zadeh. Fuzzy sets and information granularity. In: M.M. Gupta, R.K. Ragade, and R.R. Yager (eds), *Advances in Fuzzy Set Theory and Applications*. North-Holland, Amsterdam, 1979, pp. 3–18.
- [55] L.A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.
- [56] W. Pedrycz (ed.) *Granular Computing: An Emerging Paradigm*. Physica-Verlag, Heidelberg, Germany, 2001.
- [57] Y.Y. Yao. A partition model of granular computing. *Lect. Notes Comput. Sci.* 1 (January 2004) 232–253.
- [58] G. Matheron. *Random Sets and Integral Geometry*. Wiley, New York, 1975.
- [59] C. Kim. Segmenting a low-depth-of-field image using morphological filters and region merging. *IEEE Trans. Image Process.* 14 (2005) 1503–1511.
- [60] A. Sobania and J.P.O. Evans. Morphological corner detector using paired triangular structuring elements. *Pattern Recognit.* 38 (2005) 1087–1098.
- [61] U. Braga-Neto and J. Goutsias. Supremal multiscale signal analysis. *SIAM J. Math. Anal.* 36 (2004) 94–120.
- [62] G. Birkhoff. *Lattice Theory*, 3rd ed. American Mathematical Society, Providence, 1993.
- [63] G.J.F. Banon and J. Barrera. Decomposition of mappings between complete lattices by mathematical morphology, part 1. general lattices. *Signal Process.* 30 (1993) 299–327.
- [64] P. Maragos. Lattice image processing: A unification of morphological and fuzzy algebraic systems. *J. Math. Imaging Vis.* 22 (2005) 333–353.
- [65] W. Pedrycz and F. Gomide. *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press, Cambridge, MA, 1998.
- [66] W. Pedrycz. Fuzzy neural networks and neurocomputations. *Fuzzy Sets Syst.* 56 (1993) 1–28.

- [67] J.A. Anderson. *An Introduction to Neural Networks*. MIT Press, MA, 1995.
- [68] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ, 1999.
- [69] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, Germany, 1984.
- [70] D.O. Hebb. *The Organization of Behavior*. John Wiley & Sons, New York, 1949.
- [71] J.B. Fan, F. Jin, and X. Yuan. An efficient learning algorithm for fuzzy associative memories. *Acta Electr. Sin.* 24 (1996) 112–114.
- [72] B. Kosko. Bidirectional associative memories. *IEEE Trans. Syst. Man. Cybern.* 18 (1988) 49–60.
- [73] M.E. Valle. *Fundamentals and Applications of Fuzzy Morphological Associative Memories*. Ph.D. Thesis. State University of Campinas, Campinas, 2007.
- [74] The Database of Faces of AT&T Laboratories Cambridge. <http://www.uk.research.att.com/facedatabase.html>, accessed January, 2008.
- [75] M. Magalhães. *Redes neurais, metodologias de agrupamento e combinação de previsores aplicados a previsão de vazões naturais*. Master's Thesis. State University of Campinas, Campinas, 2004.
- [76] M. Magalhães, R. Ballini, R. Gonçalves, and F. Gomide. Predictive fuzzy clustering model for natural streamflow forecasting. In: *Proceedings of the IEEE International Conference on Fuzzy Systems*, Budapest, Hungary, 2004, pp. 390–394.
- [77] S. Chiu. Fuzzy model identification based on cluster estimation. *J. Intell. Fuzzy Syst.* (1994) 267–278.
- [78] V.G. Kaburlasos and S.E. Papadakis. Granular self-organizing map (grsom) for structure identification. *Neural Netw.* 19 (2006) 623–643.

34

Fuzzy Cognitive Maps

E.I. Papageorgiou and C.D. Stylios

34.1 Introduction

Fuzzy cognitive maps (FCMs) are a modeling methodology based on exploiting knowledge and experience. FCMs are originated from the theories of fuzzy logic, neural networks, and evolutionary computing and generally from soft computing and computational intelligence techniques. FCMs are consisted of concepts representing conceptual entities, which could be considered as information granules. Concepts of FCMs interact; one influences the others and exchanges information. Thus, FCMs belong to the granular computing area, which refers to the conceptualization and processing of information granules – concepts.

The FCM structure is consisted of concepts representing information granules and the causal relationships among the information granules. These information granules could be physical quantities, qualitative characteristics, abstract ideas, and generally conceptual entities. A group of experts design the FCM; they determine the most important entities which are essential to model a system. Then they define the causal relationships among these entities, indicating the relative strength of the relationships using a fuzzy linguistic variable. The directions of the causal relationships are indicated with arrowheads. FCMs conceptual models are constructed by experts, who have a sufficient and abstract understanding of the specific system and its operation.

FCMs is a qualitative modeling tool; they provide a simple and straightforward way to model the relationships among different factors. FCM can describe any system using a model with three distinct characteristics: (1) signed causality indicating positive or negative relationship, (2) the strengths of the causal relationships take fuzzy values, and (3) the causal links are dynamic where the effect of a change in one concept/node affects other nodes, which in turn may affect other nodes. The first characteristic implies both the direction and the nature of the causality. The second characteristic assigns a fuzzy number or linguistic value to reflect the strength of the causality or the degree of association between concepts. Finally, the third characteristic reflects a feedback mechanism that captures the dynamic relationship of all the nodes, which may have temporal implications.

This chapter is going to thoroughly present FCMs and the different contributions to the field. After a short literature review on FCMs, a general method for designing and developing FCMs will be presented along with the most important methodologies for training and upgrading FCMs.

34.1.1 History of Fuzzy Cognitive Mapping and Applications

Cognitive maps are originated from graph theory, which Euler introduced in 1736 [1] based on directed graphs. In directed graphs, there are links (connections) between variables (nodes) with a direction [2].

Anthropologists applied signed digraphs to represent the different social structures in human society [3]. Axelrod [4] was the first who used digraphs to describe causal relationships among variables and to mimic the way human do. He called these digraphs cognitive maps. Cognitive mapping has been applied in many areas for decision making as well as to describe expert's perceptions on complex social systems [4–11].

Kosko [12] modified Axelrod's cognitive maps, with binary values; he suggested the use of fuzzy causal functions taking numbers in $[-1, 1]$, so he introduced the FCM. Kosko examined the behavior of FCMs explaining the inference mechanism of FCM. He applied FCMs to model the effect of different policy options using a computational method [13]. There was a very important update of cognitive maps combined them with fuzzy logic, which gave them the temporal and causal nature [14, 15]. FCMs express causality over time and allow for causality effects to fluctuate as input values change. Non-linear feedback can only be modeled in a time-based system. FCMs are intended to model causality, not merely semantic relationships between concepts. The latter relationships are more appropriately represented with semantic networking tools like SemNet [16]. By modeling causality over time, FCMs facilitate the exploration of the implications of complex conceptual models, as well as represent them with greater flexibility.

In general, changes in the topology or in the weight parameters of the FCM model may result in totally different inference outcomes. The extended fuzzy cognitive map (eFCM) [17] introduced time delay in the FCM parameters and discussed non-linear properties. Actually, the eFCM was an extension of FCM that simply uses binary-valued functions for weights. As a consequence, it had little improvement over FCM in terms of inference performance.

FCMs have been successfully applied in many different scientific fields for modeling and decision making: political developments [18], electrical circuits [19], virtual sea world of dolphins, shark and fish [20], organizational behavior and job satisfaction [21], and the economic demographics of world nations [22]. FCMs were combined with data mining techniques to further utilize expert knowledge [23, 24].

Skov and Svenning [25] combined FCM with a geographic information system so that to apply expert knowledge to predict plant habitat suitability for a forest. Mendoza and Prabhu [26] used cognitive mapping to examine the linkages and interactions between indicators obtained from a multicriteria approach to forest management. FCMs were used to support the esthetical analysis of urban areas [27] and for the management of relationships among organizational members of airline services [28]. Furthermore, evaluation procedure for specifying and generating a consistent set of magnitudes for the causal relationships of an FCM, utilizing pairwise comparison techniques, has been presented [29].

Liu and Satur [30] investigated inference properties of FCMs and they proposed contextual FCMs introducing the object-oriented paradigm for decision support and they applied contextual FCMs to geographical information systems [31]. Miao and Liu proposed FCM as a robust and dynamic mechanism to represent the strength of the cause and the degree of the effect [32]. They stated that FCMs lack the temporal concept that is crucial in many applications, so that FCMs cannot effectively describe the dynamic nature of the cause. Miao *et al.* investigated the properties of the FCMs, in particular, *dynamics*, because causal inference systems by nature are dynamic systems with uncertainties and they introduced dynamic cognitive networks (DCNs), taking into account the three major causal inference factors, namely, the direction of the causal relationship, the strength of the cause, and the degrees of the effect, thus improving the capability of FCMs [33]. The DCN introduces a mechanism to quantify the description of concepts with the required precision. The arcs of DCNs are able to describe not only the causal relationship but also how it will make the effect and how long it takes for the effect to build up. Nevertheless, further research on learning paradigms and self-restructuring mechanisms in the DCN could have a significant impact on the development of intelligent systems and it could provide a flexible structure for effective causal inference in real-world applications [33].

Fuzzy cognitive networks (FCNs) [34] constitute an extension of the well-known FCMs [14] to be able to operate in continuous interaction with the physical system while at the same time to keep tracking the various operational equilibrium points met by the system. FCNs can model dynamical complex systems that change with time following non-linear laws. FCNs have been used to address the maximum power point trackers (MPPTs) in photovoltaic (PV) power systems by giving a good maximum power operation of any PV array under different conditions such as changing isolation and temperature [35]. Moreover, FCNs have been applied to control an anaerobic digestion process [36].

In bioengineering area, FCMs have been used for diagnosis and medical decision making. FCMs were used to model and analyze the radiotherapy process and they were successfully applied for decision making in radiation therapy planning systems [37]. FCMs have also been used to analyze the problem of specific language impairment diagnosis using several experts' opinions and introducing competitive learning methods [38]. Competitive learning methods for FCMs were successfully applied for medical diagnosis problems because they ensure that only one diagnosis will come up. FCMs have been used to model the supervisor for decision support actions during the labor [39]. Furthermore, an augmentation of FCMs based on learning methods was proposed to assist grade diagnosis of urinary bladder tumors [40]. An advanced medical decision support system was developed based on supplementing FCMs with case-based reasoning techniques taking advantages of the most essential characteristics of both methodologies [41].

34.2 Background of Fuzzy Cognitive Maps

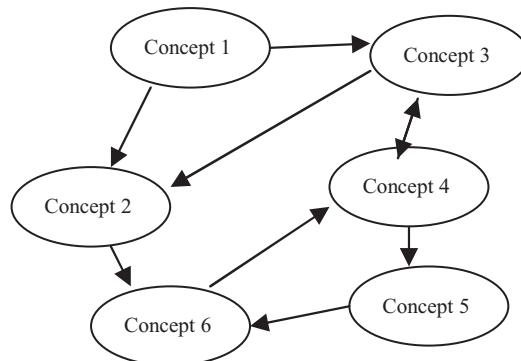
Kosko describes FCMs as fuzzy directional diagrams where feedback is permitted [42]. Like traditional causal concept maps, FCMs are consisted of nodes which represent variable concepts. The links between the concepts are signed with + or - to represent the positive or negative relationship between nodes and fuzzy-logic-based approach describes the degree of causality. FCMs can be used to create and model dynamic systems based on a causal explanation of interrelationships among concepts.

An FCM consisted of n concepts is represented in an $n \times n$ matrix. Generally, the causality between concepts is described by non-linear function $e(C_i, C_j)$, which describes the degree to which C_i influences C_j . The influence function takes values in the interval $[-1, 1]$.

See Table 34.1 for an example of a simple FCM matrix proposed by Kosko where weights take only bipolar values and Figure 34.1 illustrates a FCM with fuzzy weights.

Suppose that an FCM consists of n concepts. An $l \times n$ matrix A represents the values of the n concepts and an $n \times n$ matrix \mathbf{W} represents the causality of the relationships. In the weight matrix the row i represents the causality between concept i and all other concepts in the map. No one concept is assumed to cause itself; thus, the diagonal of the matrix is zeroed. Each element w_{ji} of the matrix \mathbf{W} indicates

Table 34.1 A simple FCM matrix



To/From	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5	Concept 6
Concept 1	0	1	1	0	0	0
Concept 2	0	0	0	0	0	1
Concept 3	0	1	0	1	0	0
Concept 4	0	0	1	0	1	0
Concept 5	0	0	0	0	0	1
Concept 6	0	0	0	1	0	0

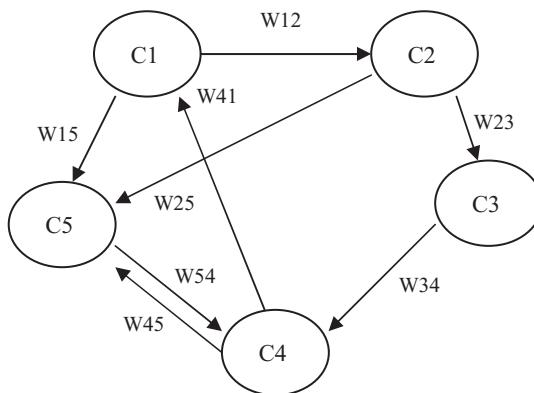


Figure 34.1 A simple FCM representation

the value of the weight w_{ji} between concept C_j and C_i . Equation (1) can be transformed as follows to describe the FCM operation using a compact mathematical equation:

$$\mathbf{A}^{(k)} = f(\mathbf{A}^{(k-1)} + \mathbf{A}^{(k-1)} \cdot \mathbf{W}) \quad (1)$$

where $\mathbf{A}^{(k)}$ is the column matrix with values of concepts at iteration step k , and f is the threshold function.

The FCM model of any system takes the initial values of concepts based on measurements from the real system and it is free to interact. The interaction is also caused by the change in the value of one or more concepts.

This interaction continues until the model [43]:

- Reaches an equilibrium fixed point, with the output values stabilizing at fixed numerical values.
- Exhibits limit cycle behavior, where the concept values falling in a loop of numerical values.
- Exhibits a chaotic behavior, where concept values reach a variety of numerical values in a non-deterministic, random way.

Compared to expert systems, FCMs are relatively quicker and easier to acquire knowledge and experience, especially exploiting the human approaches that do not usually take decisions using precise mathematics but using linguistic variables. The FCMs construction methods utilize different knowledge sources with diverse knowledge and different degrees of expertise. These knowledge sources can all be combined into one augmented FCM [44, 45]. The main advantage of the FCM development method is that there is no restriction on the number of experts or on the number of concepts.

34.2.1 General Methodology for Developing FCMs

The development and construction methodology of FCM has great importance for its success in modeling. FCM represents the human knowledge on the operation of the system and experts develop FCMs using their experience and knowledge on the system. Experts know what the main factors that influence the system are and what the essential elements of the system model may be; they determine the number and kind of concepts that the FCM is consisted of. Construction methodologies rely on the exploitation of experts' experience on system's modeling and behavior. Expert has observed the main factors that influence the behavior of the system; each one of these factors is represented by a concept at the FCM model. Experts, according to their experience, determine the concepts of FCM that may stand for events, actions, goals, values, and trends of the system. Moreover experts know which elements of the system influence other elements; for the corresponding concepts they determine the negative or positive effect of one concept on the others, with a fuzzy degree of causation. Causality is the key in representing human

cognition and the human way in reaching a decision. In this way, an expert decodes his own knowledge on the behavioral model of the system and transforms this knowledge in a weighted graph, the FCM [46].

Interconnections among concepts express the cause-and-effect relationship that exists between two concepts; this relationship can be direct or indirect. Interconnections describe the influence that has the variation on the value of one concept in the value of the interconnected concept. This causal relationship is characterized with vagueness, because of its nature, as it represents the influence of one qualitative factor on another one and is determined using linguistic variables. The following definitions describe the procedure of determining the cause-and-effect relationship between concepts.

Definition 1. Direction of correlation between two concepts.

The causal relationship between two concepts can have the following directions:

Concept C_i influence concept C_j and there is a connection between $i \rightarrow j$, so $\delta_{i,j} = 1$. Either there is a connection with the reverse direction $j \rightarrow i$ when concept C_j influences concept C_i and so $\delta_{j,i} = 1$, or there is no connection between these two concepts:

$$i, j = \begin{cases} \delta_{i,j} = 1 \\ \delta_{j,i} = 1 \\ 0 \end{cases}.$$

Definition 2. Sign of correlation between two concepts.

Correlation between two concepts can be positive or negative:

- (i) $W_{ij} > 0$, which means that when value of concept C_i increases, the value of concept C_j increases, and when value of concept C_i decreases, the value of concept C_j decreases.
- (ii) $W_{ij} < 0$, which means that when value of concept C_i increases, the value of concept C_j decreases, and when value of concept C_i decreases, the value of concept C_j increases.

Definition 3. Degree of correlation between two concepts.

The value of the weight for the interconnection w_{ij} between concept C_i and concept C_j expresses the degree of correlation of the value of one concept on the calculation of the value of the interconnected concept. Linguistic variables are used to describe the strength of influence from one concept to another and the crisp transformation of linguistic values of weights w_{ij} belongs to the interval $[-1, 1]$.

Different methodologies have been proposed to develop FCMs and extract knowledge from experts [47, 48]. Experts are asked to describe the causality among concepts and the influence of one concept to another, using linguistic notions. At first step experts use Definition 1 to describe the direction of causality between two concepts, and then they determine the kind of the relationship using Definition 2. Then they describe the degree of the causal relationship between two concepts according to Definition 3 using the linguistic variable *Influence* and the grade of influence is described with a linguistic variable such as ‘strong,’ ‘weak,’ and etc.

Influence of one concept on another is interpreted as a linguistic variable taking values in the universe $U = [-1, 1]$ and its term set $T(\text{influence})$ is proposed to be

$$T(\text{influence}) = \{\text{negatively very strong, negatively strong, negatively medium, negatively weak, zero, positively weak, positively medium, positively strong, positively very strong}\}.$$

The semantic rule M is defined as follows and these terms are characterized by the fuzzy sets whose membership functions are shown in Figure 34.2. $M(\text{negatively very strong})$ = the fuzzy set for ‘an influence below -75% ’ with membership function μ_{nvs} .

- $M(\text{negatively very strong})$ = the fuzzy set for ‘an influence more than -75% ’ with membership function μ_{nvs}

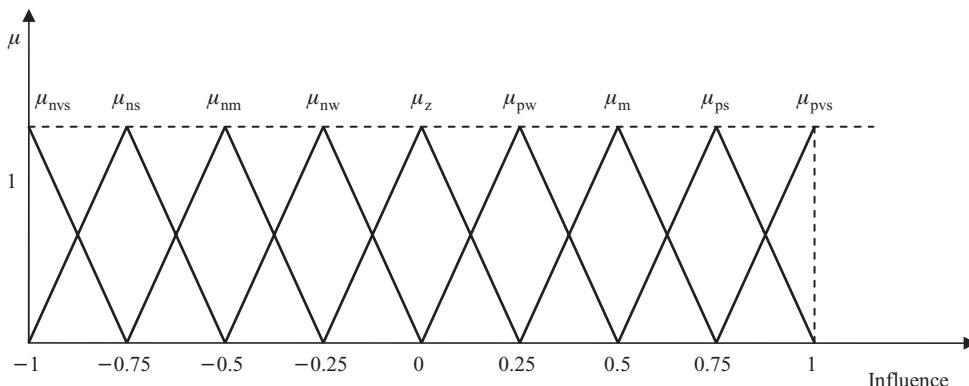


Figure 34.2 The nine membership functions corresponding to each one of the nine linguistic variables

- $M(\text{negatively strong})$ = the fuzzy set for ‘an influence close to -75% ’ with membership function μ_{ns}
- $M(\text{negatively medium})$ = the fuzzy set for ‘an influence close to -50% ’ with membership function μ_{nm}
- $M(\text{negatively weak})$ = the fuzzy set for ‘an influence close to -25% ’ with membership function μ_{nw}
- $M(\text{zero})$ = the fuzzy set for ‘an influence close to 0’ with membership function μ_z
- $M(\text{positively weak})$ = the fuzzy set for ‘an influence close to 25% ’ with membership function μ_{pw}
- $M(\text{positively medium})$ = the fuzzy set for ‘an influence close to 50% ’ with membership function μ_{pm}
- $M(\text{positively strong})$ = the fuzzy set for ‘an influence close to 75% ’ with membership function μ_{ps}
- $M(\text{positively very strong})$ = the fuzzy set for ‘an influence above 75% ’ with membership function μ_{pvs}

The values of variable *Influence* belong to a set of nine members that can describe sufficiently the relationship between two concepts. This nine-member set is in correspondence with the human matter describing the causal relationship among concepts. The set of values of the linguistic variable *Influence* could be consisted by a great number but in this case the description of relationships would be very detailed. And an expert could not describe the relationship as ‘very very much strong influence’ and discern this relationship to the ‘very much strong influence.’ On the other hand the definition of the grades of influence must be quite detailed and not to have a few members, e.g., three members describing the influence with only three statements as weak, medium, and strong.

A group of experts develop the FCM. For each interconnection of the FCM, every expert assigns a linguistic variable that describes the influence from one concept to the other. For each interconnection the M experts assign M linguistic variables and so a set of M linguistic variables describe each interconnection. The M linguistic variables are combined using the fuzzy logic method of minmax and so an overall linguistic weight is produced which represents the strength of this interconnection, which is then transformed in numeric value using the defuzzification method of center of area [15]. The overall linguistic variable is transformed in the interval $[-1, 1]$. The same procedure is applied to all the interconnections among the N concepts of the FCM [47].

The main advantage of this methodology is that experts are asked to describe the grade of causality among concepts using linguistic variables and they do not have to assign numerical causality weights [48]. Thus, an initial weight matrix

$$W^{\text{initial}} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1N} \\ w_{21} & w_{22} & \dots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \dots & w_{NN} \end{bmatrix}$$

with $w_{ii} = 0, i = 1, \dots, N$ is obtained. Using the initial concept values, A_i , and the matrix W^{initial} , the FCM interacts through the application of the rule of equation (1).

The potential convergence to undesired steady states is a major deficiency of FCMs. Thus, new techniques are proposed that could further refine the experts' knowledge and significantly enhance their performance. Learning algorithms are used to increase the efficiency and robustness of FCMs by updating the weight matrix so as to avoid convergence to undesired steady states.

34.3 Learning Methods for FCMs

The methodology of developing FCMs mainly relies on human expert experience and knowledge. The external intervention (typically from experts) for the determination of FCM parameters, the recalculation of the weights, and causal relationships every time a new strategy is adopted, as well as the potential convergence to undesired regions for concept values were significant FCM deficiencies. It is necessary to overcome these deficiencies so that to improve efficiency and robustness of FCM. Weight adaptation methods are very promising as they can alleviate these problems by allowing the creation of less error-prone FCMs where causal links are adjusted through a learning process.

Experts are involved in the construction of FCM by determining concepts and causality among them. This approach may yield to a distorted model, because experts may not consider the most appropriate factors and may assign inappropriate causality weights among FCM concepts. A better conductance of FCMs is obtained by combining them with approaches based on neural network characteristics and integrating their advantages. Specifically, neural learning techniques can be used to train the FCM and modify the most appropriate weights of interconnections among concepts. The result is a hybrid neurofuzzy system. Learning methods have been proposed for FCM training, where the gradient for each weight is calculated by the application of general rules:

$$w'_{ij} = g(w_{ij}, A_i, A_j, A'_i, A'_j) \quad (2)$$

Learning rules can fine-tune FCM cause-and-effect relationships, meaning adjusting the interconnections between concepts, as if they were synapses in a neural network.

Kosko [12] mentioned for first time that adaptation and learning methodologies based on unsupervised Hebbian-type rules can be used to adapt the FCM model and adjust its weights. He proposed the differential Hebbian learning (DHL), as a suitable unsupervised learning technique, to train FCM, but without any mathematical formulation or and implementation in any problem [14, 20]. DHL method is based on the law expressed in the equation (3), which correlates changes between causal concepts:

$$\dot{w}_{ij} = -w_{ij} + \Delta C_i \Delta C_j, \quad (3)$$

where \dot{w}_{ij} is the change of weight between concept i th and j th and where w_{ij} is the current value of this weight and $\Delta C_i \Delta C_j$ are changes in concepts i th and j th values, respectively.

The learning process iteratively updates the values of all weights at the FCM graph. Considering that value, ΔC_i is defined as the difference of i th concept values in two successive steps, which ranges between -1 and 1 . The values of C_i and C_j concepts increase or decrease at the same direction only when $\Delta C_i \Delta C_j > 0$. If $\Delta C_i \Delta C_j < 0$, then one of the concept values decreases when the other one increases. In general, the weights of outgoing edges for a given concept node are modified when the corresponding concept value changes. The weights are updated according to the following formula:

$$w_{ij}(t+1) = \begin{cases} w_{ij}(t) + c_t [\Delta C_i \Delta C_j - w_{ij}(t)], & \Delta C_i \neq 0 \\ w_{ij}(t), & \Delta C_i = 0 \end{cases}, \quad (4)$$

where w_{ij} denotes the weight of the edge between concepts C_i and C_j , ΔC_i represent the change in the C_i concept's value, t is the iteration number, and c_t is a decreasing learning coefficient; e.g.,

$$c_t = 0.1 \left[1 - \frac{t}{1.1N} \right], \quad (5)$$

where t is the current iteration number and the parameter N should be chosen to ensure that the learning coefficient c_t never becomes negative.

Huerga proposed an extension of DHL algorithm by introducing new rules to update edge values [49]. This algorithm was called balanced differential learning algorithm (BDLA). It eliminates the limitation of the initial DHL method where weight adaptation for an edge connecting two concepts (nodes) is dependent only on the values of these two concepts. But in BDLA, weights are updated taking into account all the concept values that change at the same time. This means that the formula for calculating $w_{ij}(t+1)$ takes into consideration not only changes ΔC_i and ΔC_j , but changes in all the other concepts if they occur at the same iteration and in the same direction. The BDLA algorithm was applied to adjust the structure of FCM models, which use bivalent transformation function, based on a historical data consisting of a sequence of state vectors. The goal of BDLA was to develop FCM, which would be able to generate identical sequence of state vectors given the same initial state vector. BDLA was improved in comparison to DHL method [49, 50], but both learning methods were applied only to FCMs with binary concept values, which significantly restricts their application areas. Another similar approach was the adaptive random FCMs based on the theoretical aspects of random neural networks [51].

The initial FCM model proposed by Kosko has two limitations. First, the model cannot describe many-to-one (or -many) causal relations. Second, the recall model gets trapped within limit cycles and is therefore not applicable in real-time decision-making problems. Konar and Chakraborty have proposed an extension of Kosko's model that can represent many-to-one (or -many) causal relations by using Petri nets. The proposed recall model was free from limit cyclic behavior. Other researchers selected fuzzy Petri nets (FPNs) [52, 53] to model FCMs due to the fact that FPNs support the necessary many-to-one (or -many) causal relations and have already proved themselves useful as an important reasoning [54, 55] and learning tool [56]. This unsupervised learning and reasoning process is realized with the adaptation of weights in an FPN using a different form of Hebbian learning. This proposed model converges to stable points in both encoding and recalls phases.

Moreover, a novel scheme of supervised learning on a fuzzy Petri net has been proposed in [57], providing semantic justification of the hidden layers and being capable of approximate reasoning and learning from noisy training instances. The algorithm for training a feedforward fuzzy Petri net and the analysis of its convergence have been successfully applied in object recognition from two-dimensional geometric views.

A different learning objective for FCM was presented by Khan and Chong in 2003 [58]. Instead of training the structure of FCM model, their approach was to find the initial state vector (initial conditions) that leads a given model to the desired fixed-point attractor or limit cycle.

Most of the existing learning approaches for weight updating do not take into account the feedback of the real system. A weight updating method for FCMs based on system feedback has been proposed by Boutalis *et al.* [34]. This method supposed that FCM reaches its equilibrium point using direct feedback from the node values of the real system and the learning limitations are imposed by the reference nodes. Moreover, the updating procedure of the method is enhanced using information on previous equilibrium points of the system operation. This is achieved by storing knowledge from already encountered operational situations into some fuzzy if–then rules.

Furthermore, two novel Hebbian-based approaches for FCM training, the active Hebbian learning (AHL) and the non-linear Hebbian learning (NHL) algorithms [59, 60], were proposed. The AHL algorithm takes into consideration the experts' knowledge and experience for the initial values of the weights, which are derived from the summation of experts' opinions. AHL algorithm supposes that there is a sequence of activation concepts, which is depending on the specific problem's configuration and characteristics. A seven-step AHL procedure is proposed to adjust the FCM weights. Mathematical formulation, implementation, and analysis of AHL supported by examples can be found in [59].

The core of the second approach, the NHL algorithm, is a non-linear extension to the fundamental Hebbian rule [61]. The main idea behind NHL is to update only those weights that experts determined, i.e., the non-zero weights. Weight values of FCM are updated synchronously, and yet they have fixed signs for the entire learning process. As a result, the NHL algorithm retains the structure of the obtaining model, which is enforced by the expert(s), but at the same time it requires human intervention before starting the learning process.

Moreover, evolutionary computation-based methods have been proposed for learning FCM causal weights by training the connection matrix of input data and thus eliminating expert involvement during development of the model. In Section 34.5, it is presented the evolutionary learning algorithms for FCMs proposed till today and referred in the corresponding literature.

34.4 Unsupervised Learning Algorithms for FCMs

34.4.1 The Active Hebbian Learning Algorithm

AHL algorithm has been introduced recently [59]. The novelty of this algorithm is based on supposing sequence of influence from one concept to another; in this way the interaction cycle is dividing in steps. During the FCM developing phase, experts are asked to determine the sequence of activation concepts, the activation steps, and the activation cycle. At every activation step, one (or more) concept(s) becomes activated concept(s), triggering the other interconnecting concepts, and in turn, at the next simulation step, may become activation concept. When all the concepts have become activated concepts, the simulation cycle has closed and a new one starts until the system converges in an equilibrium region.

In addition to the determination of sequence of activation concepts; experts select a limited number of concepts as outputs for each specific problem which are defined as the activation decision concepts (ADCs). These concepts are in the center of interest; they stand for the main factors and characteristics of the system, known as outputs and their values represent the final state of the system.

Suppose that there is the FCM shown on Figure 34.3, where experts determined the following activation sequence: $C_1 \rightarrow C_2, C_j \rightarrow C_i \rightarrow C_m \rightarrow C_n$. At second step of the cycle, according to the activation sequence concept C_j is the triggering concept that influences concept C_i , as shown in Figure 34.3. This concept C_j is declared the activation concept, with the value A_j^{act} that triggers the interconnected corresponding concept C_i , which is the activated concept. At the next iteration step, concept C_i influences the other interconnected concepts C_m and so forth. This learning algorithm has asynchronous stimulation mode which means that when concept C_j is the activation concept that triggers C_i , the corresponding weight w_{ji} of the causal interconnection is updated and the modified weight $w_{ji}^{(k)}$ is derived for each iteration step k .

Figure 34.3 is an instance of the FCM model during the activation sequence. The FCM model consists of n nodes and at the second activation step, the activation concept C_j influences the activated concept C_i . The following parameters are depicted in Figure 34.3:

C_i is the i th concept with value $A_i(k)$, $1 \leq i \leq n$.

w_{ji} is the weight describing the influence from C_j to C_i .

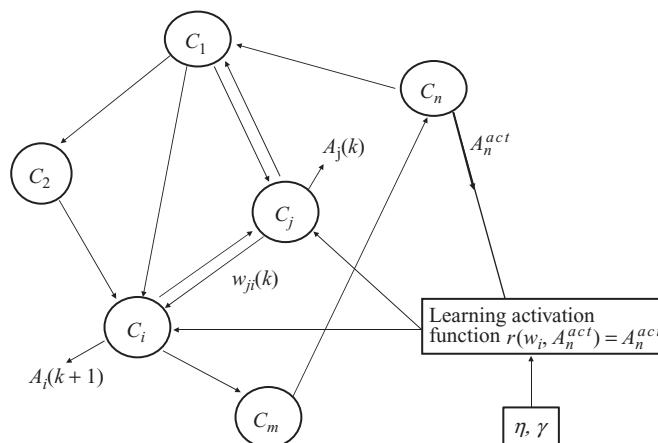


Figure 34.3 The activation weight-learning process for FCMs

$A_j^{act}(k)$ is the activation value of concept C_j , which is triggering the interconnected concept C_i .
 γ is the weight decay parameter.

η is the learning rate parameter, depending on simulation cycle c .
 $A_i(k)$ is the value of activated concept C_i , at iteration step k .

The value $A_i(k + 1)$ of the activated concept C_i , at iteration step $k + 1$, is calculated, computing the influence from the activation concepts with values A_l^{act} to the specific concept C_i due to modified weights $w_{li}(k)$ at iteration step k , through the equation

$$A_i(k + 1) = f(A_i(k) + \sum_{l \neq i} A_l^{act}(k) \cdot w_{li}(k)), \quad (6)$$

where A_l are the values of concepts C_l that influence the concept C_i , and $w_{li}(k)$ are the corresponding weights that describe the influence from C_l to C_i .

For example, in Figure 34.3, l takes values 1, 2, and j , and A_1 , A_2 , and A_j are the values of concepts C_1 , C_2 , and C_j that influence C_i . Thus, the value A_i of the concept, after triggering at step $k + 1$, is calculated:

$$A_i(k + 1) = f(A_i(k) + A_1^{act}(k) \cdot w_{1i}(k) + A_2^{act}(k) \cdot w_{2i}(k) + A_j^{act}(k) \cdot w_{ji}(k)). \quad (7)$$

The AHL algorithm relates the values of concepts and values of weights to the FCM model. We introduced a mathematical formalism for incorporating the learning rule, with the learning parameters and the introduction of the sequence of activation [59].

The proposed rule has the general mathematical form:

$$w_{ji}(k) = (1 - \gamma) \cdot w_{ji}(k - 1) + \eta \cdot A_j^{act}(k - 1) \cdot A_i(k - 1), \quad (8)$$

where the coefficients η and γ are positive learning factors called learning parameters.

In order to prevent indefinite growing of weight values, we suggest normalization of weight at value 1, $\|\mathbf{W}\| = 1$, at each step update:

$$w_{ji}(k) = \frac{(1 - \gamma) \cdot w_{ji}(k - 1) + \eta \cdot A_j^{act}(k - 1) \cdot A_i(k - 1)}{\left[\sum_{\substack{j=1 \\ j \neq i}} ((1 - \gamma) \cdot w_{ji}(k - 1) + \eta \cdot A_j^{act}(k - 1) \cdot A_i(k - 1))^2 \right]^{1/2}}, \quad (9)$$

where the addition in the denominator covers all of the interconnections from the activation concepts C_j to the activated concepts C_i .

For low learning rates of parameters η , γ , equation (8) can – without any loss of precision – be simplified to

$$w_{ji}(k) = (1 - \gamma) \cdot w_{ji}(k - 1) + \eta \cdot A_j^{act}(k - 1) \cdot [A_i(k - 1) - w_{ji}(k - 1) \cdot (A_j^{act}(k - 1))]. \quad (10)$$

The equation (1) that calculates the value of each concept of FCM takes the form of equation (6), where the value of weight $w_{ji}(k)$ is calculated using equation (10).

The learning parameters η and γ are positive scalar factors. The learning rate parameter η is exponentially attenuated with the number of activation-simulation cycles c so that the trained FCM converges fast. Thus, $\eta^{(c)}$ is selected to be decreased where the rate of decrease depends on the speed of convergence to the optimum solution and on the updating mode. Thus, the following equation is proposed:

$$\eta^{(c)} = b_1 \cdot \exp(-\lambda_1 \cdot c) \quad (11)$$

Depending on the problem's constraints and the characteristics of each specific case, the parameters b_1 and λ_1 may take values within the following bounds: $0.01 < b_1 < 0.09$ and $0.1 < \lambda_1 < 1$, which are determined using experimental trial and error method for fast convergence.

The parameter γ is the weight decay coefficient which is decreasing following the number of activation cycles c . The parameter γ is selected for each specific problem to ensure that the learning process converges in a desired steady state. When the parameter γ is selected as a decreasing function at each activation cycle c , the following form is proposed:

$$\gamma^{(c)} = b_2 \cdot \exp(-\lambda_2 \cdot c), \quad (12)$$

where b_2 and λ_2 are positive constants determined by a trial-and-error experimental process. These values influence the rate of convergence to the desired region and the termination of the algorithm.

In addition, for the AHL algorithm, two criteria functions have been proposed [59]. The first one is the criterion function J which examines the desired values of outputs concepts, which are the values of activation concepts we are interested in.

The criterion function J has been suggested as

$$J = \sqrt{\sum_{j=1}^m \left([ADC_j - A_j^{\min}]^2 + [ADC_j - A_j^{\max}]^2 \right)}, \quad (13)$$

where A_j^{\min} is the minimum target value of concept ADC_j and A_j^{\max} is the corresponding maximum target value of ADC_j . At the end of each cycle, the value of J calculates the Euclidean distance of ADC_j value from the minimum and maximum target values of the desired ADC_j respectively. The minimization of the criterion function J is the ultimate goal, according to which we update the weights and determine the learning process.

One more criterion for this learning algorithm of FCMs has been proposed. This second criterion is determined by the variation of the subsequent values of ADC_j concept, for simulation cycle c , yielding value e , which has to be minimum and takes the form

$$|ADC_j^{(c+1)} - ADC_j^{(c)}| < e, \quad (14)$$

where ADC_j is the value of j th concept.

The term e is a tolerance level keeping the variation of values of $ADC(s)$ as low as possible and it is proposed to be equal to $e = 0.001$, satisfying the termination of iterative process.

Thus, for training FCM using the asynchronous AHL algorithm two criteria functions have been proposed. The first one is the minimization of the criterion function J and the second one is minimization of the variation of the two subsequent values of ADC s, represented in equations (13) and (14), respectively, in order to determine and terminate the iterative process of the learning algorithm.

The proposed algorithm is based on defining a sequence of concepts that means distinction of FCM concepts as inputs, intermediates, and outputs; this distinction depends on the modeled system and the focus of experts. During the training phase a limited number of concepts are selecting as outputs (those we want to estimate their values). The expert's intervention is the only way to address this definition. This learning algorithm extracts the valuable knowledge and experience of experts and can increase the operation of FCMs and implementation in real case problems just by analyzing existing data, information, and experts' knowledge about the given systems.

The training process implementing the AHL into an n -concept FCM is described analytically in [59]. The schematic representation of this training process is given in Figure 34.4. This learning algorithm drives the system to converge in a desired region of concepts values within the accepted-desired bounds for ADC s concepts.

34.4.2 Non-Linear Hebbian Learning Algorithm for FCMs

The second proposed unsupervised algorithm for training FCMs is based on the non-linear Hebbian-type learning rule for ANNs learning [62–64]. This unsupervised learning rule has been modified and adapted for the FCM case, introducing the NHL algorithm for FCMs.

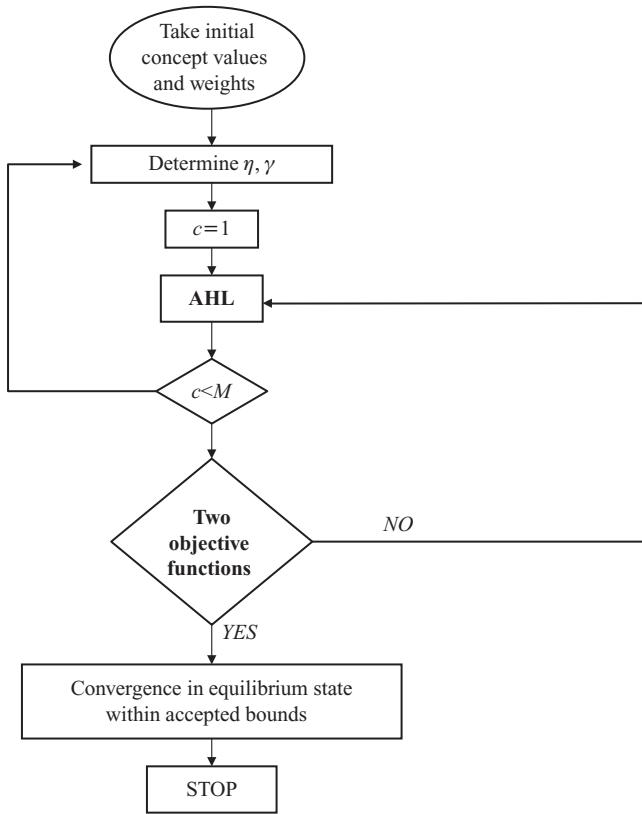


Figure 34.4 The flowchart of the training process using the AHL technique

The NHL algorithm is based on the premise that all the concepts in the FCM model are synchronously triggering at each iteration step and change their values synchronously. During this triggering process all weights w_{ji} of the causal interconnections of the concepts are updated and the modified weight $w_{ji}^{(k)}$ are derived for iteration step k .

The value $A_i^{(k+1)}$ of concept c_i at iteration step $k + 1$ is calculated, computing the influence of interconnected concepts with values A_j to the specific concept C_j due to modified weights $w_{ji}^{(k)}$ at iteration step k , through the following equation:

$$A_i^{(k+1)} = f \left(A_i^{(k)} + \sum_{\substack{j=1 \\ j \neq i}}^N A_j^{(k)} \cdot w_{ji}^{(k)} \right). \quad (15)$$

Taking the advantage of the general non-linear Hebbian-type learning rule for neural networks [64–66], we introduce the mathematical formalism incorporating this learning rule for FCMs. This algorithm relates the values of concepts and values of weights in the FCM model, and it takes the general mathematical form

$$\Delta w_{ji} = \eta A_i^{(k-1)} (A_j^{(k-1)} - w_{ji}^{(k-1)} A_i^{(k-1)}). \quad (16)$$

where the coefficient η is a very small positive scalar factor called learning parameter, which is determined using experimental trial-and-error method in order to optimize the final solution.

Equation (16) is modified and adjusted for FCMs and the following form of the non-linear weight-learning rule for FCMs is proposed:

$$w_{ji}^{(k)} = \gamma \cdot w_{ji}^{(k-1)} + \eta A_i^{(k-1)} \left(A_j^{(k-1)} - \text{sgn}(w_{ji}^{(k-1)}) w_{ji}^{(k-1)} A_i^{(k-1)} \right), \quad (17)$$

where the γ is the weight decay learning coefficient.

The value of each concept of FCM is updated through the equation (15), where the value of weight $w_{ji}^{(k)}$ is calculated using equation (17).

Indeed, when experts develop an FCM, they usually propose a quite spare weight matrix \mathbf{W} . Using the NHL algorithm the initially non-zero weights are updating synchronously at each iteration step through the equation (17), until the termination of the algorithm. The NHL algorithm does not assign new interconnections and all the zero weights do not change value. When the algorithm termination conditions are met, the final weight matrix \mathbf{W}_{NHL} is derived.

Implementation of NHL algorithm requires determination of upper and lower bounds for the learning parameter η ; using trial-and-error experiments the values of learning rate parameter η were determined to belong in $0 < \eta < 0.1$. For any specific case-study problem, a constant value for η is calculated [67].

34.4.2.1 Two Termination Conditions

During the FCM development stage, experts define the desired output concepts (DOCs). These concepts stand for the main characteristics and outputs of the system that we want to estimate their values, which reflect the overall state of the system. The distinction of FCM concepts as inputs and outputs is determined by the group of experts for each specific problem. Experts select the output concepts and they consider the rest as initial stimulators or interior concepts of the system. The proposed learning algorithm extracts hidden and valuable knowledge of experts and it can increase the effectiveness of FCMs and their implementation for real problems.

Two complementary termination conditions of the NHL process have been proposed: the first termination condition is the minimization of the following cost function F_1 :

$$F_1 = \sqrt{\|DOC_j^{(k)} - T_j\|^2}, \quad (18)$$

where T_j is the mean target value of the concept DOC_j . At each step, the value of F_1 calculates the square of the Euclidean distance of actual DOC_j value and mean target value T_j of the DOC_j values.

Let us assume that we want to calculate the cost function F_1 of concept C_j . It is required that DOC_j take values in the range $DOC_j = [T_j^{\min}, T_j^{\max}]$. Then the target value T_j of the concept C_j is determined as

$$T_j = \frac{T_j^{\min} + T_j^{\max}}{2}. \quad (19)$$

If we consider the case of an FCM model, where there are m DOCs, then for the calculation of F_1 , we take the sum of the square differences between the m DOC values and the m Ts mean values of DOCs, and the equation (19) takes the following form:

$$F_1 = \sqrt{\sum_{j=1}^m (DOC_j^{(k)} - T_j)^2}. \quad (20)$$

The objective of the training process is to find the set of weights that minimize function F_1 .

In addition to the previous statements, one more criterion for the NHL has been introduced so as to terminate the algorithm after a limited number of steps. This second criterion is based on the variation of the subsequent values of DOC_j concepts, for iteration step k , yielding a very small value e ,

Algorithm: “Nonlinear Hebbian Learning”
<i>Step 1:</i> Read input concept state \mathbf{A}^0 and initial weight matrix \mathbf{W}^0
<i>Step 2:</i> For iteration step k .
<i>Step 3:</i> Update the weights:
$w_{ij}^{(k)} = w_{ij}^{(k-1)} + \eta A_j^{(k-1)} (A_i^{(k-1)} - sgn(w_{ij}) w_{ij}^{(k-1)} A_j^{(k-1)})$
<i>Step 4:</i> Calculate $A_i^{(k)}$ according to the eq. (13)
<i>Step 5:</i> Calculate the two termination functions
<i>Step 6:</i> Until both the termination conditions are met, go to step 2
<i>Step 7:</i> Return the final weights \mathbf{W}_{NHL} .

Figure 34.5 NHL algorithm for FCMs

taking the form

$$F_2 = \left| DOC_j^{(k+1)} - DOC_j^{(k)} \right| < 0.002, \quad (21)$$

where $DOC_j^{(k)}$ is the value of j th concept at iteration step k .

The constant value $e = 0.002$ has been proposed after a large number of simulations for different FCM cases. When the variation of two subsequent values of DOC_j is less than this number, it is pointless for the system operation to continue the training process.

When both terminations functions F_1 and F_2 are satisfied, the learning algorithm terminates and the desired equilibrium region for the DOCs is reached.

A generic description of the proposed NHL algorithm for FCMs is given in Figure 34.5 and the flowchart in Figure 34.6 describes the NHL-based algorithmic procedure.

34.5 Enhancing FCMs Using Evolutionary Computation Techniques

Evolutionary computation (EC) has become a standard term to indicate problem-solving techniques, which use design principles inspired from models and the natural evolution of species. Historically, there are three main algorithmic developments within the field of EC: evolution strategies [68, 69], evolutionary programming [70], and genetic algorithms [71, 72]. Common on these approaches is that they are population-based algorithms, which use operators inspired by population genetics to explore the search space. (The most typical genetic operators are reproduction, mutation, and recombination.) Each individual in the algorithm represents directly or indirectly (through a decoding scheme) a solution to the problem under consideration. The reproduction operator refers to the process of selecting the individuals that will survive and be part of the next generation. This operator typically uses a bias toward good-quality individuals: the better the objective function value of an individual, the higher the probability that the individual will be selected and therefore it will be part of the next generation. The recombination operator (often also called crossover) combines parts of two or more individuals and generates new individuals, also called offspring. The mutation operator is a unary operator that introduces random modifications to one individual.

Differences among the different EC algorithms concern the particular representation chosen for the individuals and the way genetic operators are implemented [68, 69, 71, 73]. For example, genetic algorithms typically use binary or discrete-valued variables to represent information in individuals and

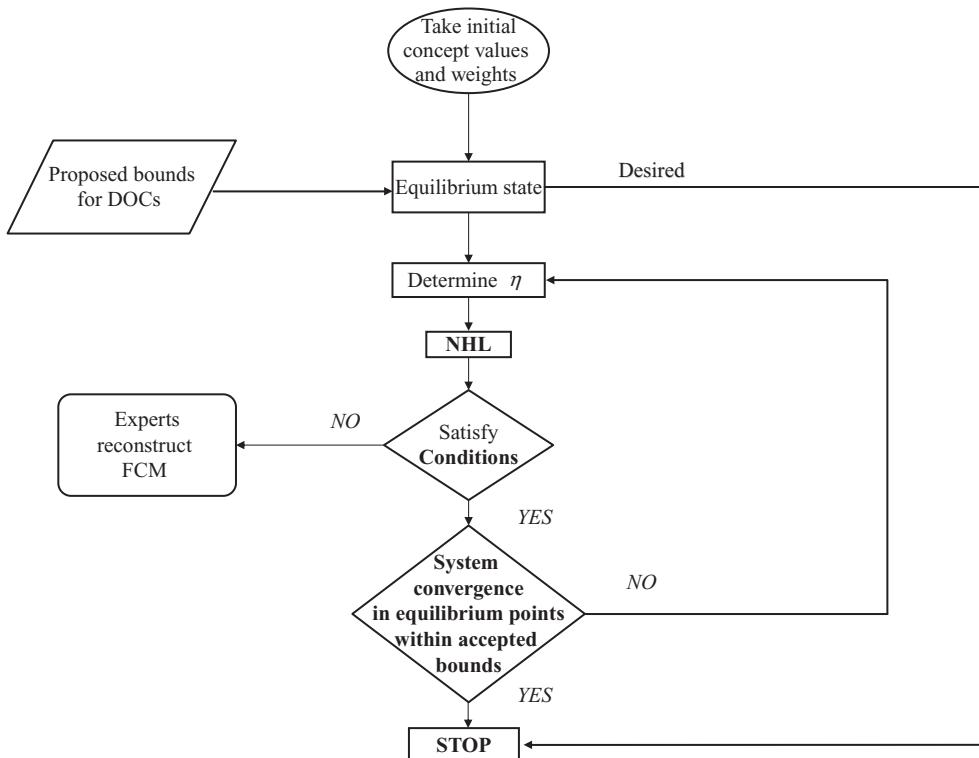


Figure 34.6 Flowchart of the training process using NHL technique

they favor the use of recombination, while evolution strategies and evolutionary programming often use continuous variables and put more emphasis on the mutation operator [72, 74, 75].

Koulouriotis *et al.* [76] were the first who applied evolution strategies for learning FCMs. Their technique was exactly the same used in neural networks training. In this case, the learning process is based on a collection of input/output pairs, which are called examples. Particular values of inputs and outputs depend on the designer's choice. Inputs are defined as the initial state vector values, whereas outputs are the final state vector values, i.e., values of state vector when the FCM simulation terminates. One of its main drawbacks is that it does not take into consideration the initial structure and experts' knowledge for the FCM model, but uses data sets determining input and output patterns in order to define the cause-and-effect relationships which satisfy the fitness function. Another main drawback is the need for multiple state vector sequences (input/output pairs), which might be difficult to obtain for many real-life problems. The calculated weights appear as large deviations from the actual FCM weights.

Recently, two different approaches based on the application of genetic algorithms for learning FCM connection matrix have been proposed. The first approach involving genetic algorithms performs a goal-oriented analysis of FCM [77]. This learning method did not aim to compute the weight matrix, but to find the initial state vector, which leads a predefined FCM (with a fixed weight matrix) to converge to a given fixed-point attractor or limit cycle solution. They viewed the problem of the FCM backward inference as one of optimization and applied a genetic algorithm-based strategy to search for the optimal stimulus state.

The second, more powerful, genetic algorithm-based approach has been proposed to develop FCM connection matrix which is based on historical data, consisting of a sequence of state vectors. It uses a real-coded genetic algorithm (RCGA) which allows eliminating expert involvement during development of the model and learns the connection matrix for an FCM that uses continues transformation function,

which is a more general problem than the one considered in [78, 79]. The RCGA learning method is fully automated. Based on historical data given as time series (called input data) it establishes the FCM model (called candidate FCM), which is able to mimic the data. This approach is very flexible in terms of input data: it can use either one time series or multiple sets of concepts values over successive iterations. The central part of this method is the real-coded genetic algorithm, which is a floating-point extension to genetic algorithm [72]. The RCGA learning approach was intensively tested and experiments proved its effectiveness and high quality [79]. The analysis of RCGA learning ability is depending on the available size of historical data. Increasing the size of input data improves the accuracy of learning and on the other hand insufficient size of input data may result in poor quality of learning. In the latter case multiple different models that mimic input data of small size can be generated, and most of them fail to provide accurate results for experiments with new initial conditions, which were unseen during learning process [80]. The main advantage of this method is the absence of human intervention but the RCGA method needs investigation in terms of its convergence and more investigation to associate the GA parameters with the characteristics of a given experimental data set.

Another novel FCM learning procedure has been suggested by Parsopoulos *et al.* [81, 82] based on particle swarm optimization (PSO), which is a stochastic, population-based optimization algorithm. PSO belongs to the class of swarm intelligence algorithms [83], exploiting a population, called a swarm, of individuals, called particles, to probe the search space. Each particle moves with an adaptable velocity within the search space and retains a memory of the best position it ever encountered. The best position ever attained by all individuals of the swarm is communicated to all the particles in the case of global variant of PSO [84–87]. The PSO-based learning algorithm utilizes historical data consisting of a sequence of state vectors that leads to a desired fixed-point attractor state. It provides a search procedure, which optimizes a problem-depended fitness function $\varphi()$, by maintaining and evolving a swarm of candidate solutions. The individual of the swarm yielding the best fitness value throughout all generations, so that to give the optimal solution. Using the PSO method, a number of appropriate weight matrices can be derived leading the system to desired convergence regions. This approach is very fast and efficient to calculate the optimum cause-and-effect relationships of the FCM model and to overcome the main drawback of the FCM, which is the recalculation of the weights every time a new real case is adopted. This training method was successfully applied to an industrial control problem [82].

The detection of an appropriate weight matrix that leads the FCM to a steady state at which the output concepts lie in their corresponding bounds, while the weights retain their physical meaning, is the main aspect of this learning process. This aspect is attained by imposing constraints on the potential values assumed by weights. To succeed that, an objective function F was considered, which is non-differentiable (described in [81]) to calculate the desired values of output concepts by producing the appropriate weight matrix of the system. The PSO approach is used for the minimization of the objective function, due to the non-differentiability of F , and the global minimizers of the objective function are weight matrices that lead the FCM to a desired steady state. Generally, a plethora of weight matrices are produced through the PSO algorithm that lead the FCM to convergence to the desired regions. It is quite natural to obtain such suboptimal matrices that differ in subsequent experiments because PSO is a stochastic algorithm. All these matrices are proper for the FCM design and follow the constraints of the problem, though each matrix may have different physical meaning for the system.

Statistical analysis of the obtained weight matrices may help in the better understanding of the system's dynamics as it is implied by the weights, as well as in the selection of the most appropriate suboptimal matrix. The aforementioned procedure uses only primitive information from the experts. Nevertheless, any information available *a priori* may be incorporated to enhance the procedure, either by modifying the objective function in order to exploit the available information or by imposing further constraints on the weights. The proposed approach has proved to be very efficient in practice and its operation is illustrated on two different application areas in industry and medicine [81, 82].

34.6 Conclusion

FCMs are a conceptual modeling technique, consisted of concepts, the information entities, which interact following the causal relationships among the granular entities. FCMs take advantage of theories

and approaches derived from discipline areas. FCMs synergistically utilize these theories of fuzzy sets, neurocomputing, evolutionary computing, and naturally inspired computing. FCMs representing good knowledge on a given system or process and accompanying them with computational intelligence techniques can contribute toward the establishment of FCMs as a robust technique.

One of the criticisms directed to FCMs is that they require human experts to come up with fuzzy causal relationship values, which may sometimes be inaccurate. Learning techniques can alleviate this problem by allowing the creation of less error-prone simple FCMs with causal links, which are then fuzzified through an adaptive learning process. The applicability of the proposed weight adaptation methodologies to real-world industrial problems has been successfully applied [88].

This chapter discussed the main aspects of FCMs representation, development, and then the learning approaches for FCMs. It presents their origin, the background, and the research efforts and contributions till today. Much research effort has been put on the utilization of neurocomputing methods, such as Hebbian learning algorithms for training FCMs and so developing augmented FCMs.

The most important characteristics of FCM model, which make it to differ from other knowledge-based models, is its interpretability, transparency, and efficiency in modeling and operation behavior of complex systems.

The main features of FCM model are summarized in the following:

- Representation of human knowledge on the modeling and operation behavior
- Exploitation of human's operator experience
- Selection of different factors of system
- Relations among different parts of system
- Symbolic representation of system's behavior
- Human-like reasoning
- Qualities for planning, decision making, and failure detection

All these qualities of FCMs make them suitable and applicable for modeling complex systems with many real applications.

Acknowledgments

The work of E.I. Papageorgiou was supported by a postdoctoral research grant from the Greek State Scholarship Foundation 'I.K.Y.'

References

- [1] N.L. Biggs, E.K. Lloyd, and R.J. Wilson. *Graph Theory: 1736-1936*. Clarendon Press, Oxford, 1976.
- [2] F. Harary, R.Z. Norman, and D. Cartwright. *Structural Models: An Introduction to the Theory of Directed Graphs*. John Wiley & Sons, New York, 1965.
- [3] P. Hage and F. Harary. Structural models. In: *Anthropology*. Cambridge University Press, Cambridge, 1983.
- [4] R. Axelrod. *Structure of Decision: The Cognitive Maps of Political Elites*. Princeton University Press, Princeton, NJ, 1976.
- [5] M.G. Bougon, K. Weick, and D. Binkhorst. Cognition in organizations: An analysis of the Utrecht Jazz Orchestra. *Adm. Sci. Q.* 22 (1977) 606–639.
- [6] S.M. Brown. Cognitive mapping and repertory grids for qualitative survey research: some comparative observations. *J. Manage. Stud.* 29 (1992) 287–307.
- [7] K. Carley and M. Palmquist. Extracting, representing and analyzing mental models. *Soc. Forces* 70 (1992) 601–636.
- [8] P. Cossette and M. Audet. Mapping of an idiosyncratic schema. *J. Manage. Stud.* 29 (1992) 325–347.
- [9] J.A. Hart. Cognitive maps of three Latin America policy makers. *World Polit.* 30 (1977) 115–140.
- [10] J.H. Klein and D.F. Cooper. Cognitive maps of decision makers in a complex game. *J. Oper. Res. Soc.* 33 (1982) 63–71.

- [11] K. Nakamura, S. Iwai, and T. Sawaragi. Decision support using causal knowledge base. *IEEE Trans. Syst. Man Cybern.* SMC12 (1982) 765–777.
- [12] B. Kosko. Fuzzy cognitive maps. *Int. J. Man-Mach. Stud.* 24 (1986) 65–75.
- [13] Kosko. Hidden patterns in combined and adaptive knowledge networks. *Int. J. Approx. Reason.* 2 (1988) 377–393.
- [14] B. Kosko. Fuzzy associative memory systems. In: *Fuzzy Expert Systems..* CRC Press, Boca Raton, FL, 1992, pp. 135–162.
- [15] C.T. Lin and C.S.G. Lee. *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems.* Prentice Hall, Upper Saddle River NJ, 1996.
- [16] J.F. Sowa. *Principles of Semantic Networks: Explorations in the Representation of Knowledge.* Morgan Kaufmann Publishers, San Mateo, CA, 1991.
- [17] M. Hagiwara. Extended fuzzy cognitive maps. In *Proceedings of the 1st IEEE International Conference on Fuzzy Systems*, New York, March 1992, pp.795–801.
- [18] R. Taber. Knowledge processing with fuzzy cognitive maps. *Expert Syst. Appl.* 2 (1991) 83–87.
- [19] M.A. Styblinski and B.D. Meyer. Fuzzy cognitive maps, signal flow graphs, and qualitative circuit analysis. In: *Proceedings of the 2nd IEEE International Conference on Neural Nets (ICNN-87)*, San Diego, CA, 1988, pp. 549–556.
- [20] J. Dickerson and B. Kosko. *Virtual Worlds as Fuzzy Cognitive Maps. Presence*, Vol. 3, no. 2. MIT Press, MA, 1994, pp. 173–189.
- [21] J.P. Craiger, D.F. Goodman, R.J. Weiss, and A. Butler. Modeling organizational behavior with fuzzy cognitive maps. *J. Comput. Intell. Organ.* 1 (1996) 120–123.
- [22] M. Schneider, E. Shnaider, A. Kandel, and G. Chew. Automatic construction of FCMs. *Fuzzy Sets Syst.* 93 (1998) 161–172.
- [23] T. Hong and I. Han. Knowledge-based data mining of news information on the Internet using cognitive maps and neural networks. *Expert Syst. Appl.* 23 (2002) 1–8.
- [24] K.C. Lee, J.S. Kin, N.H. Chung, and S.J. Kwon. Fuzzy cognitive map approach to web-mining inference amplification. *J. Experts Syst. Appl.* 22 (2002) 197–211.
- [25] F. Skov and J.-C. Svenning. Predicting plant species richness in a managed forest. *Forest Ecol. Manage.* 620 (2003) 1–11.
- [26] G.A. Mendoza and R. Prabhu. Qualitative multi-criteria approaches to assessing indicators of sustainable forest resource management. *Forest Ecolog. Manage.* 174 (2003) 329–343.
- [27] G. Xirogiannis, J. Stefanou, and M. Glykas. A fuzzy cognitive map approach to support urban design. *Expert Syst. Appl.* 26 (2) (2004) 257–268.
- [28] I.I. Kang and S. Lee. Using fuzzy cognitive map for the relationship management in airline service. *Expert Syst. Appl.* 26 (4) (2004) 545–555.
- [29] K. Muata and O. Bryson. Generating consistent subjective estimates of the magnitudes of causal relationships in fuzzy cognitive maps. *Comput. Oper. Res.* 31 (8) (2004) 1165–1175.
- [30] Z.Q. Liu and R. Satur. Contextual fuzzy cognitive map for decision support in geographical information systems. *J. IEEE Trans. Fuzzy Syst.* 7 (1999) 495–507.
- [31] Z.Q. Liu. *Fuzzy Cognitive Maps: Analysis and Extension.* Springer, Tokyo, 2000.
- [32] Y. Miao and Z.Q. Liu. On causal inference in fuzzy cognitive map. *IEEE Trans. Fuzzy Syst.* 8 (1) (2000) 107–120.
- [33] Y. Miao, Z. Liu, C. Siew, and C. Miao. Dynamical cognitive network—an extension of fuzzy cognitive map. *IEEE Trans. Fuzzy Syst.* 9 (5) (2001) 760–770.
- [34] Y.S. Boutalis, T.L. Kottas, B. Mertzios, and M.A. Christodoulou. A fuzzy rule based approach for storing the knowledge acquired from dynamical FCMs. In: *Proceedings of the 5th International Conference on Technology and Automation (ICTA'05)*, Thessaloniki, Greece, October 2005, pp. 119–124.
- [35] Y. Boutalis, A. Karlis, and T. Kottas. Fuzzy cognitive networks + fuzzy controller as a self adapting control system for tracking maximum power point of a PV-Array. In: *Proceedings of 32nd Annual Conference of the IEEE Industrial Electronics Society, IECON 2006*, Paris, France, November 7–10, 2006, pp. 4355–4360.
- [36] T. Kottas, Y. Boutalis, V. Diamantis, O. Kosmidou, and A. Aivasidis. A fuzzy cognitive network based control scheme for an anaerobic digestion process. In: *Proceedings of the 14th Mediterranean Conference on Control and Applications, Session TMS – Process Control*, Ancona, Italy, June 28–30, 2006, pp. 1–7.
- [37] E.I. Papageorgiou, C.D. Stylios, and P.P. Groumpas. An integrated two-level hierarchical decision making system based on fuzzy cognitive maps (FCMs). *IEEE Trans. Biomed. Eng.* 50 (12) (2003) 1326–1339.
- [38] V.C. Georgopoulos, G.A. Malandraki, and C.D. Stylios. A fuzzy cognitive map approach to differential diagnosis of specific language impairment. *Artif. Intell. Med.* 679 (2003) 1–18.
- [39] C.D. Stylios, G. Georgoulas, and P.P. Groumpas. The challenge of using soft computing for decision support during labour. In: *Proceedings of 23rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, Istanbul, Turkey, CD-ROM, October 25–28, 2001.

- [40] E.I. Papageorgiou, C.D. Stylios, and P.P. Groumpas. Unsupervised learning techniques for fine-tuning fuzzy cognitive map causal links. *Int. J. Hum-Comput. Stud.* 64 (2006) 727–743.
- [41] V.C. Georgopoulos and C.D. Stylios. Augmented fuzzy cognitive maps supplemented with case base reasoning for advanced medical decision support. In: M. Nikravesh, L.A. Zadeh, and J. Kacprzyk (eds), *Soft Computing for Information Processing and Analysis Enhancing the Power of the Information Technology, Studies in Fuzziness and Soft Computing*. Springer-Verlag, Berlin/Heidelberg, 2005, pp. 391–405.
- [42] B. Kosko. Adaptive bi-directional associative memories. *IEEE Trans. Syst. Man Cybern.* 18 (1) (1988) 49–60.
- [43] B. Kosko. *Fuzzy Engineering*. Prentice Hall, NJ, 1997.
- [44] C.D. Stylios and P.P. Groumpas. A soft computing approach for modeling the supervisor of manufacturing systems. *J. Intell. Robot. Syst.* 26 (3–4) (1999) 389–403.
- [45] C.D. Stylios, V. Georgopoulos, and P.P. Groumpas. introducing the theory of fuzzy cognitive maps in distributed systems. In: *Proceedings of the 12th IEEE International Symposium on Intelligent Control*, Istanbul, Turkey, July 16–18, 1997, pp. 55–60.
- [46] C.D. Stylios and P.P. Groumpas. Modeling complex systems using fuzzy cognitive maps. *IEEE Trans. Syst., Man Cybern.* 34 (1) (2004) 155–162.
- [47] C.D. Stylios, P.P. Groumpas, and V.C. Georgopoulos. A fuzzy cognitive maps approach to process control systems. *J. Adv. Comput. Intell.* 3 (5) (1999) 409–417.
- [48] C.D. Stylios and P.P. Groumpas. Fuzzy cognitive maps in modeling supervisory control systems. *J. Intell. Fuzzy Syst.* 8 (2000) 83–98.
- [49] A.V. Huerga. A balanced differential learning algorithm in fuzzy cognitive maps. In: *Proceedings of 16th International Workshop on Qualitative Reasoning2002*, Sitges, Spain, Poster, June 2, 2002.
- [50] A. Vazquez. *A Balanced Differential Learning Algorithm in Fuzzy Cognitive Maps*. Technical Report. Departament deLlenguatges I Sistemes Informatics, Universitat Politecnica deCatalunya (UPC) 2002.
- [51] J. Aguilar. Adaptive random fuzzy cognitive maps. In: F.J. Garjio, J.C. Riquelme, and M. Toro (eds), *Proceedings of IBERAMIA, Lecture Notes in Artificial Intelligence 2527*. Springer-Verlag, Berlin/Heidelberg, 2002, pp. 402–410.
- [52] A.J. Bugarin and S. Barro. Fuzzy reasoning supported by Petri nets. *IEEE Trans. Fuzzy Syst.* 2 (2) (1994) 135–150.
- [53] A. Daltrini and F. Gomide. An extension of fuzzy Petri nets and its applications. *IEEE Trans. Syst. Man Cybern.* 23 (5) (1993) 1255–1264.
- [54] A. Konar and A.K. Mandal. Uncertainty management in expert systems using fuzzy Petri nets. *IEEE Trans. Knowl. Data Eng.* 8 (1) (1996) 96–105.
- [55] A. Konar. *Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain*. CRC Press, Boca Raton, FL, 1999.
- [56] W. Pedrycz and F. Gomide. A generalized fuzzy Petri net model. *IEEE Trans. Fuzzy Syst.* 2 (4) (1994) 295–301.
- [57] A. Konar and U.K. Chakraborty. Reasoning and unsupervised learning in a fuzzy cognitive map. *Inf. Sci.* 170 (2–4) (2005) 419–441.
- [58] M.S. Khan and A. Chong. Fuzzy cognitive map analysis with genetic algorithm. In: *Proceedings of the 1st Indian International Conference on Artificial Intelligence*, IICAI 2003, Hyderabad, India, December 18–20, 2003, IICAI 2003, pp. 1196–1205.
- [59] E.I. Papageorgiou, C.D. Stylios, and P.P. Groumpas. Active Hebbian learning algorithm to train fuzzy cognitive maps. *Int. J. Approx. Reason.* 37 (3) (2004) 219–247.
- [60] E.I. Papageorgiou and P.P. Groumpas. A weight adaptation method for fuzzy cognitive maps to a process control problem. In: M. Bubak, G.D.V. Albada, P.M.A. Sloot, and J.J. Dongarra (eds), *Proceedings of International Conference on Computational Science, ICCS 2004, Krakow, Poland, Lecture Notes in Computer Science 3037 Vol. II*. Springer-Verlag, Berlin/Heidelberg, 2004, pp. 515–522.
- [61] E. Oja, H. Ogawa, and J. Wangviattana. Learning in nonlinear constrained Hebbian networks. In: T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas (eds), *Artificial Neural Networks*. North-Holland, Amsterdam, 1991, pp. 385–390.
- [62] D.O. Hebb. *The Organization of Behaviour: A Neuropsychological Theory*. John Wiley, New York, 1949.
- [63] E. Oja. Neural networks, principal components and subspaces. *Int. J. Neural Syst.* 1 (1989) 61–68.
- [64] M. Hassoun. *Fundamentals of Artificial Neural Networks*. MIT Press, Bradford Book, MA, 1995.
- [65] E. Oja. Data compression, Feature extraction, and autoassociation in feed- forward neural networks. In: T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas (eds), *Artificial Neural Networks*. Elsevier Science Publishers B.V., North-Holland, Amsterdam 1991.
- [66] L. Xu. Theories for unsupervised learning: PCA and its nonlinear extensions. In: *Proceedings of IEEE International Conference on Neural Networks*, Vol. II, New York, 1994, pp. 1252–1257.

- [67] E.I. Papageorgiou, K.E. Parsopoulos, P.P. Groumpas, and M.N. Vrahatis. A first study of fuzzy cognitive maps learning using particle swarm optimization. In: *Proceedings of IEEE 2003 Congress on Evolutionary Computation*. IEEE Press, Canberra, Australia, December 8–12, 2003, pp. 1440–1447.
- [68] I. Rechenberg. Evolution strategy. In: J.M. Zurada, R.J. Marks II, and C. Robinson (eds), *Computational Intelligence: Imitating Life*. IEEE Press, Piscataway, NJ, 1994, pp. 147–159.
- [69] H.P. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.
- [70] D. Fogel. *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ, 1995.
- [71] J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, MI, 1975.
- [72] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley/Prentice Hall, MA, 1989.
- [73] A.P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, IN, 2005.
- [74] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin, 1994.
- [75] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 1996.
- [76] D.E. Koulouriotis, I.E. Diakoulakis, and D.M. Emiris. Learning fuzzy cognitive maps using evolution strategies: A novel schema for modeling a simulating high-level behavior. *Proc. IEEE Congr. Evol. Comput.* 1 (2001) 364–371.
- [77] M.S. Khan, S. Khor, and A. Chong. Fuzzy cognitive maps with genetic algorithm for goal-oriented decision support. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 12 (2004) 31–42.
- [78] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat. Evolutionary development of fuzzy cognitive maps. In: *Proceedings of 2005 IEEE International Conference on Fuzzy Systems (FUZZ IEEE 2005)*, Reno, NV, 2005a, pp. 619–624.
- [79] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat. Genetic learning of fuzzy cognitive maps. *Fuzzy Sets Syst.* 153 (3) (2005b) 371–401.
- [80] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat. Learning fuzzy cognitive maps with required precision using genetic algorithm approach. *Electron. Lett.* 40 (24) (2004) 1519–1520.
- [81] K.E. Parsopoulos, E.I. Papageorgiou, P.P. Groumpas, and M.N. Vrahatis. Evolutionary computation techniques for optimizing fuzzy cognitive maps in radiation therapy systems. In: K. Deb, R. Poli, W. Banzhaf, H.-G. Beyer, E. Burke, P. Darwen, D. Dasgupta, D. Floreano, J. Foster, M. Harman, O. Holland, and P.L. Lanzi (eds), *Proceedings of Genetic and Evolutionary Computation Conference (GECCO), Seattle, Washington, USA, June 26–30, 2004, Lecture Notes in Computer Science 3102*. Springer-Verlag Publications, Berlin/Heidelberg, 2004b, pp. 402–413.
- [82] E.I. Papageorgiou, K.E. Parsopoulos, C.D. Stylios, P.P. Groumpas, and M.N. Vrahatis. Fuzzy cognitive maps learning using particle swarm optimization. *Int. J. Intell. Inf. Syst.* 25 (1) (2005) 95–121.
- [83] J. Kennedy and R.C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, 2001.
- [84] R.C. Eberhart, P. Simpson, and R. Dobbins. *Computational Intelligence PC Tools*. Academic Press, Professional (APP), New York, 1996.
- [85] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, Vol. IV. IEEE Service Center, Piscataway, NJ, 1995, pp. 1942–1948.
- [86] K.E. Parsopoulos and M.N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing* 1 (2–3) (2002) 235–306.
- [87] K.E. Parsopoulos and M.N. Vrahatis. On the computation of all global minimizers through particle swarm optimization. *IEEE Trans. Evol. Comput.* 8 (3) (2004a) 211–224.
- [88] E.I. Papageorgiou, C.D. Stylios, and P.P. Groumpas. Fuzzy cognitive map learning based on nonlinear hebbian rule. In: T.D. Gedeon and L.C.C. Fung (eds), *Proceedings on Artificial Intelligence. Lecture Notes in Artificial Intelligence 2903*. Springer-Verlag, Berlin/Heidelberg, 2003b, pp. 254–266.

Part Three

Applications and Case Studies

35

Rough Sets and Granular Computing in Behavioral Pattern Identification and Planning

Jan G. Bazan

35.1 Introduction

Solving problems in complex dynamical systems (see Section 35.2) requires techniques for combining information arriving from many sources of different quality. Usually the information is inaccurate and incomplete. One of paradigms for dealing with such complex problems is *granular computing* (see, e.g., [1–4]). The basic idea of the granular computing paradigm is a notion of *granule*. From our perspective a granule is a conceptual unit (a piece of information or data), which can be integrated into a larger infrastructure consisting of other granules and dependencies between them. In other words, granules are building blocks for other, more complex granules and, finally, for the whole intelligent system, which is able to solve considered problems. The structure of complex granules is usually described hierarchically and each granule matches the context in which it is used at a particular level of abstraction in the total system infrastructure (see, e.g., [1]).

Behaviors of objects from complex dynamical systems (see Section 35.2) are often described in a natural language and they can be treated as spatiotemporal concepts (granules). The exact description of such concepts is often not possible using purely analytical methods because the description itself contains many vague concepts. They are expressed on a much higher level of abstraction than data obtained by sensor measurement, so far mostly applied to the approximation of concepts.

In this chapter, we discuss methods for modeling of granules in behavioral pattern identification and planning tasks. These granules are constructed from data and domain knowledge and are used for approximation of many vague concepts used in solving the above-mentioned tasks.

In the first part of this chapter (from Section 35.2 to Section 35.9), we introduce rough set tools (see [5]) for approximation of such complex spatiotemporal concepts by granules. In our approach, we use the notion of *temporal pattern* (see Section 35.3) that expresses simple temporal features of objects or groups of objects in a given complex dynamical system. Temporal patterns can be used to approximate *temporal concepts* (see Section 35.4), which represent more complex features of complex objects. More complex behavior of complex objects or groups of complex objects can be presented in the form of *behavioral graphs* (see Sections 35.5 and 35.6). Any behavioral graph can be interpreted as a *behavioral pattern* and can be used as a complex classifier for identification of complex behaviors (see Section 35.7). The

identification of complex behaviors can be speeded up using special decision rules, which we present in Section 35.8. To illustrate presented methods and to verify the effectiveness of classifiers based on behavioral patterns, we have performed several experiments with the data sets recorded in the road simulator (see [6–8]) and medical data (see [8, 9]). In Section 35.9, we present results of our experiments with the data sets recorded in the road simulator.

The aim of the second part of this chapter (from Section 35.10 to Section 35.13) is to present an automatic planning method for complex objects or groups of complex objects in the complex dynamical system on the basis of data sets and domain knowledge. The problem of automated planning can be understood as a standard problem of searching automatic for classifier from a decision table with a complex decision attribute. Values of such decision attribute are plans that should be realized for complex objects or groups of complex objects described by condition attributes. Hence, the discussed method of automated planning can be treated in some sense as an application of methods described in the first part of this chapter (see Section 35.10 for more details). Behavior of single complex objects is modeled by means of the so-called *planning rules* (see Section 35.10), which are defined on the basis of data sets and domain knowledge. All planning rules may be represented in a form of the so-called *planning graphs* (see Section 35.10), whose nodes are state descriptions and directed edges correspond to actions occurring in planning rules. Solving problems of automatic planning consists in finding a path in the planning graph from the initial state to an expected final state. However, because of the rule planning indeterminism many variant plans should be considered and the possibility of plan reconstruction should be taken into account. It is worth noticing that the conditions for performing an action are described by vague complex concepts which are expressed in natural language and require approximation. In order to check the effectiveness of suggested automatic planning methods, there were performed experiments concerning planning of treatment of infants suffering from respiratory failure (see [10, 11]). We present results of experiments in Section 35.13.

35.2 Complex Dynamical Systems and Their Monitoring

Many real-life problems can be modeled by systems of objects and their parts changing and interacting over time. The objects are usually linked by some dependencies, can cooperate between themselves, and are able to perform flexible autonomous complex actions (operations). Such systems are identified as *complex dynamical systems* (see, e.g., [12, 13]), also called as an *autonomous multiagent system* (see, e.g., [13–16]) or *swarm systems* (see, e.g., [15]). For example, one can consider *road traffic* as a dynamical system represented by road simulator (see, e.g., [17, 18]). Driving simulation takes place on a board (see Figure 35.1) which presents a crossroads together with access roads. During the simulation the vehicles may enter the board from all four directions that is east, west, north, and south. The vehicles coming to the crossroads from the south and north have the right of way in relation to the vehicles coming from the west and east. Each of the vehicles entering the board has only one aim – to drive though the crossroads safely and leave the board. Both the entering and exiting roads of a given vehicle are determined at the beginning; that is, at the moment the vehicle enters the board. While driving on a road each vehicle can be treated as an *intelligent autonomous agent* (see, e.g., [14]). Each agent is ‘observing’ the surrounding situation on the road, keeping in mind his destination and his own parameters, and makes an independent decision about further steps by performing some maneuvers such as passing, overtaking, changing lane, or stopping.

Another example can be taken from the medical practice. It concerns with the treatment of infants with respiratory failure, where a given patient is treated as an investigated complex dynamical system, while diseases of the patient are treated as complex objects changing and interacting over time (see [8, 9]).

Behaviors of objects from complex dynamical systems are often described in a natural language and they can be treated as spatiotemporal concepts (i.e., some kinds of granules). The description of such concepts is often not possible using purely analytical methods and the description itself contains many vague concepts. They are expressed on a much higher level of abstraction than data obtained by sensor measurements (further called *sensor data*), so far mostly applied as the only data source to the approximation of concepts. Much attention has been devoted to spatiotemporal exploration methods

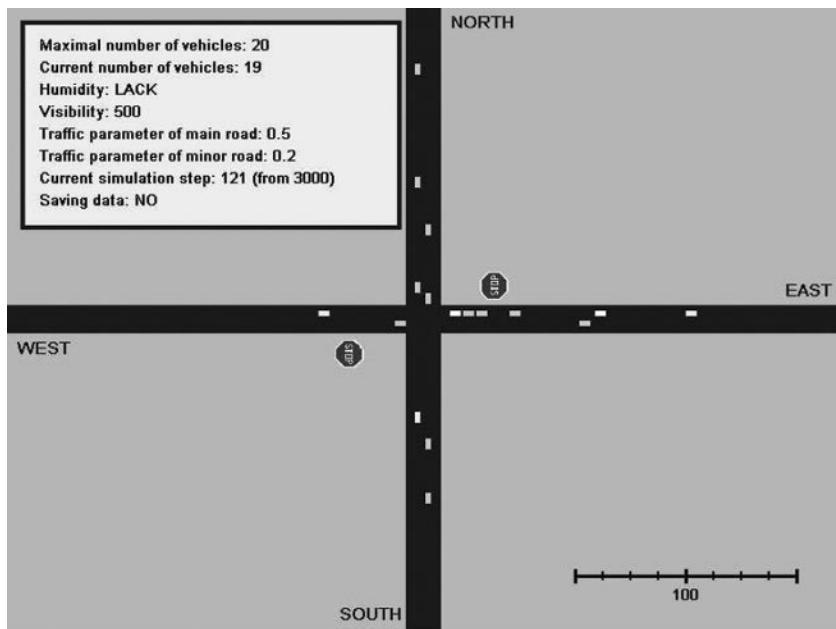


Figure 35.1 The board of simulation

in literature (see e.g., [19, 20]). Current experience indicates that the approximation of such concepts requires the support of knowledge of the domain to which the approximated concepts are applied, i.e., *domain knowledge*. This is the knowledge regarding concepts occurring in a particular domain and various relationships between these concepts. This knowledge exceeds considerably the knowledge gathered in data sets. It is often represented in natural language and acquiring it usually takes place during user-expert dialogs.

Difficulties which appear while approximating spatiotemporal concepts with the help of sensor data result from the fact that sensor data and complex spatiotemporal concepts are separated by a considerable *semantic distance*, for they are defined and interpreted on extremely different levels of abstraction. Hence the approximation of such spatiotemporal concepts with the help of elementary granules obtained from sensory data does not lead to classifiers of satisfactory quality (see, e.g., [21, 22]). Therefore such concepts we call in this chapter as *complex concepts*.

One of the methods of representing domain knowledge is recording it in a form of so-called conceptual *ontology* (see e.g., [23]). The ontology is to be understood as a finite set of concepts creating a hierarchy and relationships between these concepts, which connect concepts from different hierarchical levels. At the same time we assume that ontology specification includes incomplete information about concepts and relationships occurring in ontology. In particular, for every concept there is a set of objects (granules) consisting of positive and negative examples-objects for the concepts.

Our approach to hierarchical modeling of granules is based on conceptual ontologies and decision systems. These systems are used for approximation of concepts on different levels of ontology hierarchy (see e.g., [6, 9, 24]). In the modeling process, several kinds of granules are involved. Assume that on a given level of hierarchy objects are represented by some already-constructed granules, e.g., by relational structures over some subsets of objects from the lower level of hierarchy or families of such structures. Composition relative to some constraints of such granules is making it possible to define granules representing objects (or their groups) on the next level of hierarchy. These granules become objects of information systems or decision systems on the next level of hierarchy. Condition attributes in these systems define new granules describing properties of objects. In particular, new granules can

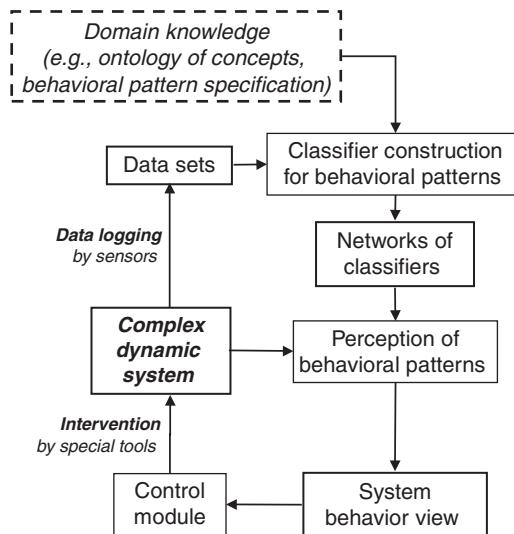


Figure 35.2 Monitoring of complex dynamical systems using behavioral patterns

be defined by the indiscernibility classes of information systems constructed on the higher level of the hierarchy. Granules defined by condition attributes on each level of the hierarchy can be fused into new granules called as classifiers, i.e., granules approximating concepts defined by decision attributes.

An efficient monitoring of complex dynamical systems can be made using so-called behavioral patterns (see Figure 35.2). Roughly speaking, any behavioral pattern can be understood as a way to represent some behavior of complex objects and their parts changing over time (see Section 35.7 for more details). Behavioral patterns are often described in a natural language and they can be treated as complex spatiotemporal concepts. Hence, their identification is possible using network of classifiers constructed on the basis domain knowledge and data sets accumulated for the given complex dynamical system (see Figure 35.2). Such identification can be very important for identification or prediction of behavior of a dynamical system; e.g., some behavioral patterns correspond to undesirable behaviors of complex objects. If in the current situation some patterns are identified, then the control module can use this information to tune selected parameters to obtain the desirable behavior of the system (see Figure 35.2). This can make it possible to overcome dangerous or uncomfortable situations. For example, if some behavior of a vehicle that causes a danger on the road is identified, the control module can try to change its behavior by using some suitable means such as road traffic signaling, radio message, or police patrol intervention (see [6, 7]).

35.3 Temporal Patterns

In this section, we present a method of granules construction that can be used to represent of simple behaviors of complex objects in complex dynamical systems. This method is based on so-called *elementary actions* and *temporal patterns*.

In many complex dynamical systems, there are some *elementary actions* (performed by complex objects) that are easily expressed by a local change of object parameters, measured in a very short, but a registerable, period. So, an elementary action should be understood as a very small but meaningful change of some sensor values such as location, distance, and speed. In case of the road traffic example, we distinguish the following elementary actions such as increase in speed, decrease in speed, and lane change.

However, a perception of composite actions requires analysis of elementary actions performed over a longer period called a *time window*. Therefore, if we want to predict composite actions or identify a behavioral pattern, we have to investigate all elementary actions that have been performed in the current

time window. Hence, one can consider, e.g., the frequency of elementary actions within a given time window and temporal dependencies between them. These properties can be expressed using so-called *temporal patterns*. We define a temporal pattern as a function of parameters of an object observed over a time window. In this chapter, we consider temporal patterns of the following types:

- *Sensory pattern*: a numerical characterization of values of selected sensor from a time window (e.g., the minimal, maximal or mean value of a selected sensor, initial and final values of selected sensor, deviation of selected sensor values);
- *Local pattern*: a crisp (binary) characterization of occurrences of elementary actions in a given time window (e.g., action A occurs within a time window, the action B occurs at the beginning of a time window, action C does not occur within a time window);
- *Sequential pattern*: a binary characterization of temporal dependencies between elementary actions inside a time window (e.g., action A persists throughout a time window or action A begins before action B, action C occurs after action D).

Sensory patterns are determined directly by values of some sensors. For example, in case of the road traffic one can consider sensory patters such as

- the minimal speed of a vehicle in a given time window,
- the average speed of a vehicle in a given time window,
- the speed of a vehicle in the first time point of a given time window.

The value of a local or sequential pattern is determined by elementary actions registered in a time window. Local or sequential patterns are often used in queries with binary answers such as *Yes* or *No*. For example, in case of road traffic we have exemplary local patterns such as

- ‘Did the vehicle speed increase in the time window?’;
- ‘Was the speed stable in the time window?’;

and sequential patterns such as

- ‘Did the speed increase before a move to the left lane occurred?’;
- ‘Did the speed increase before a speed decrease occurred?’;

We assume that any temporal pattern ought to be defined by a human expert using domain knowledge accumulated for the given complex dynamical system.

One can see that any temporal pattern represents a family of granules associated with its values. For instance the temporal pattern defined as the following question: ‘Did vehicle speed increase in the time window?’ represents two granules: first granule, to which the vehicles whose speed increased in the investigated time window belong, and second granule, to which the vehicles whose speed did not increase in the investigated time window belong.

35.4 Temporal Concepts

The temporal patterns discussed in Section 35.3 can be treated as new features for approximation of complex concepts. In this chapter, we call them *temporal concepts*. We assume that temporal concepts are specified by a human expert.

Such complex concepts (often interpreted as complex granules) are usually used in queries about the status of some objects in a particular time window. Answers to such queries can be of the form *yes*, *no*, or *does not concern*.

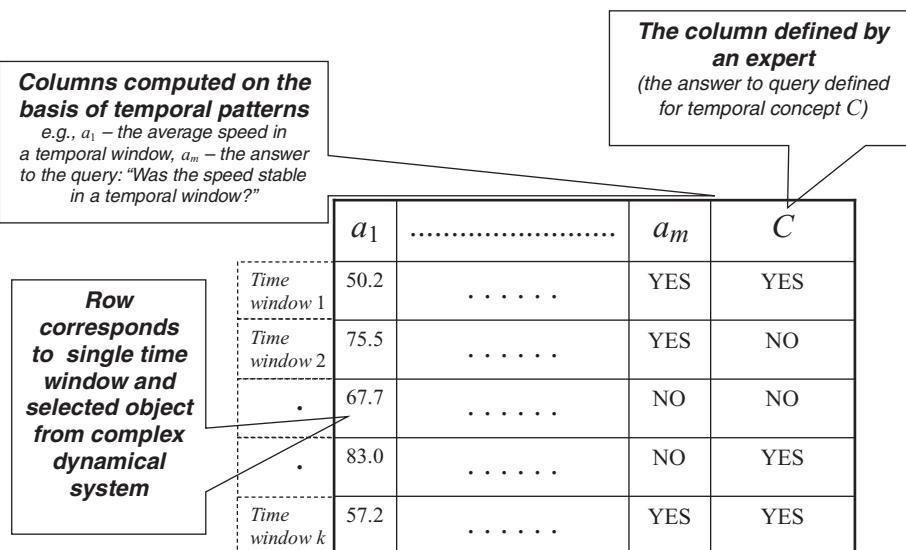


Figure 35.3 The scheme of the temporal pattern table (PT)

For example, in case of road traffic one can define complex concepts such as ‘Is a vehicle accelerating in the right lane?’, ‘Is a vehicle speed stable while changing lanes?’, or ‘Is the speed of a vehicle in the left lane stable?’.

The approximation of temporal concepts is defined by classifiers, which are usually constructed on the basis of decision rules. However, if we want to apply classifiers for approximation of temporal concepts, we have to construct a relevant decision table called a *temporal pattern table* (PT – see Figure 35.3).

A temporal PT is constructed from a table T in which it is stored information about objects occurring in a complex dynamical system. Any row of table T represents information about parameters of a single object registered in a time window. Such a table can be treated as a data set accumulated from observations of the behavior of a complex dynamical system. Assume, for example, that we want to approximate a temporal concept C using table T. Initially, we construct a temporal pattern table PT as follows.

- Construct table PT with the same objects as contained in table T.
- Any condition attribute of table PT is computed using temporal patterns defined by a human expert for the approximation of concept C.
- Values of the decision attribute (the characteristic function of concept C) are proposed by the human expert.

Finally, we construct a classifier for table PT that can approximate temporal concept C.

35.5 Behavioral Graphs for Objects

Temporal concepts defined for objects from a complex dynamical system and approximated by classifiers can be used to the construction of more complex granules, which we call as *behavioral graphs*. Nodes of this graph represent temporal concepts, while links between nodes represent temporal dependencies. Such links between nodes can be defined by an expert or extract from a data set that has been accumulated for a given complex dynamical system. Figure 35.4 presents a behavioral graph for a single object-vehicle exhibiting a behavior of vehicle while driving on a road.

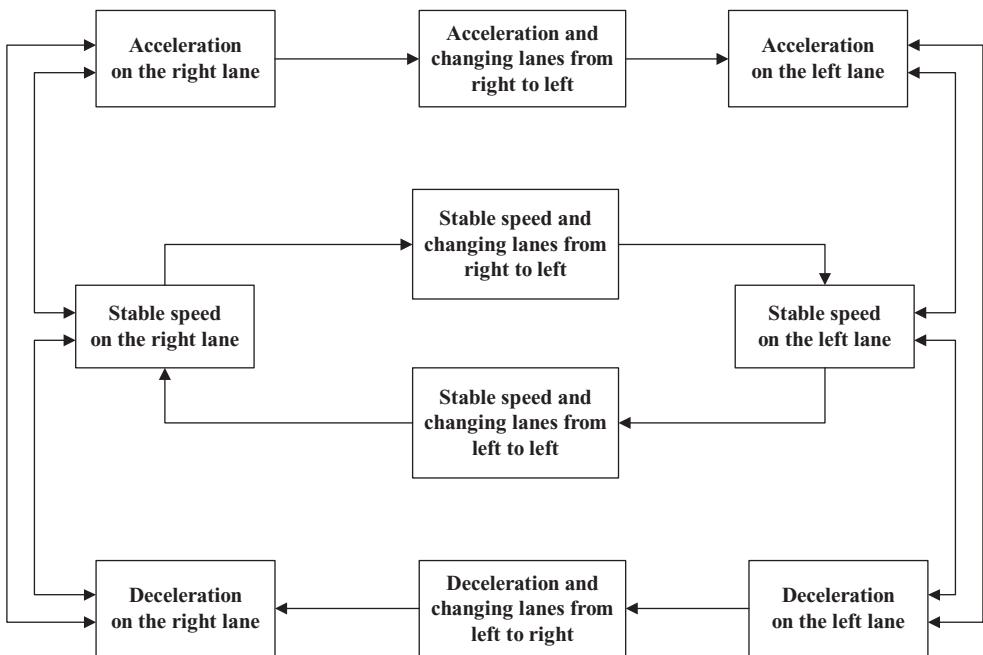


Figure 35.4 A behavioral graph for a single object-vehicle

In this behavioral graph, for example, links between node ‘Acceleration on the right lane’ and node ‘Acceleration and changing lanes from right to left’ indicate that after an acceleration in the right lane, a vehicle can change to the left lane (maintaining its acceleration during both time windows). In addition, a behavioral graph can be constructed for different kinds of objects such as single vehicles or groups of vehicles and defined for behaviors such as driving on the strength road, driving through crossroads, overtaking, and passing.

One can say that a behavior graph represents information about possible changes or possible behaviors of the investigated complex object (e.g., a vehicle, a patient). Therefore we consider behavioral graphs as models for behavioral patterns (see Section 35.7).

35.6 Behavioral Graphs for Groups of Objects

Temporal concepts can be defined and approximated not only for single object (see Section 35.4) but also for groups of objects. We introduce a method for approximation of temporal concepts for a group of objects based on what is known as a *group temporal pattern table* (GT – see Figure 35.5), which is constructed using the methodology presented in Section 35.4 for construction of the temporal pattern table, but with some important differences. To construct such a table, assume that behavioral graphs for all objects belonging to a group of objects have been constructed. For example, we construct behavioral graphs for all vehicles belonging to the investigated group of vehicles.

Except behavioral graphs created for every object belonging to the group, usually we consider still different behavioral graphs. These behavioral graphs describe temporal relations between objects in the group. For example, one can define a behavioral graph describing changes of distances between vehicles from the investigated group of vehicles.

For each behavioral graph, we define new temporal patterns using only two kinds of temporal patterns, namely, local patterns and sequential patterns. Sensory patterns are not used since information about sensors is not directly accessible on this abstraction level.

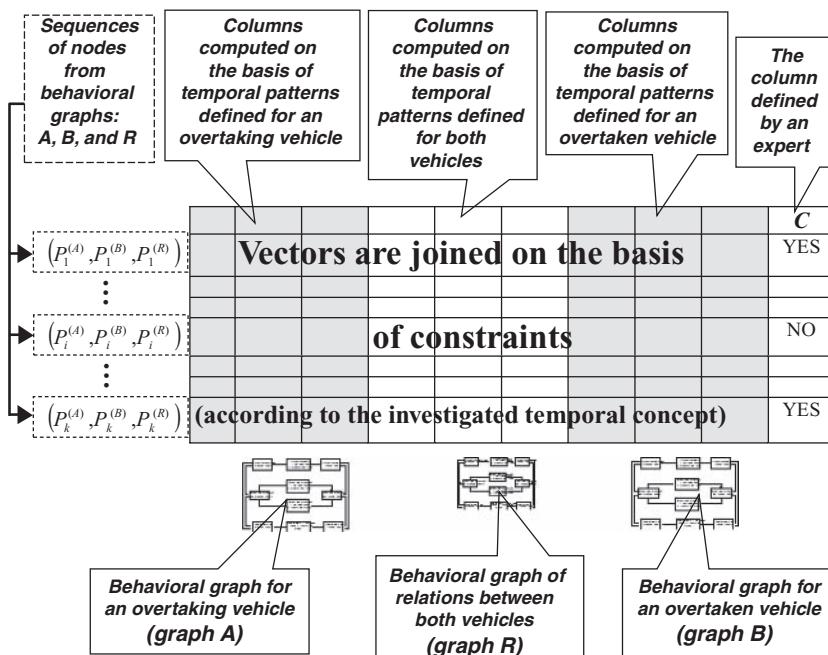


Figure 35.5 The scheme of the group temporal pattern table (GT) constructed for group of two vehicles

The value of a local or sequential pattern is determined by temporal concepts registered in a sequence of time windows. Local or sequential patterns are used in queries with binary answers such as *Yes* or *No*. For example, in case of the behavioral graph from the Figure 35.4, we have exemplary local patterns such as

- Was one of the time window from the investigated sequence classified to the concept ‘Acceleration on the right lane?’;
- Were all time windows from the investigated sequence classified to the concept ‘Stable speed on the left lane?’;

and sequential patterns such as

- Is there a time window in the investigated sequence which was classified to the notion ‘Acceleration on the right lane’ and is there other window (later in the investigated sequence) which was classified to the notion ‘Acceleration and changing lanes from right to left’?
- Is there a time window in the investigated sequence which was classified to the notion ‘Deceleration on the left lane’ and is there other window (later in the investigated sequence) which was classified to the notion ‘Deceleration and changing lanes from left to right’?

Any row in GT represents information about all complex objects (from the investigated group). This information is given as paths of nodes from behavioral graphs of complex objects belonging to this group. Any path in a behavioral graph is understood as a sequence of graph nodes (temporal concepts) registered over some period (over a sequence of time windows) for a complex object.

More precisely, GT is constructed in the following way (see also Figure 35.5):

- Table GT is created from training data for approximation of a given temporal concept C.
- Any object of GT represents information for all objects from the considered group. This information is based on behavioral graphs and for any such object is given a sequence of graph nodes from behavioral graphs registered over some period.

- Any condition attribute of GT is computed using temporal patterns provided by the expert for the approximation of a concept C. There are two types of condition attributes:
 - Condition attributes of the first type are defined for any pair selected by expert consisting of a temporal pattern and a sequence of graph nodes from behavioral graph constructed for a considered complex object belonging to the investigated group. The condition attribute value is computed using the selected temporal pattern (local or sequential) on the selected path.
 - Condition attributes of the second type are defined for selected by expert pairs consisting of a temporal pattern and all paths in the row. These attributes describe temporal relations between objects in the investigated group of complex objects.
- Values of the decision attribute (the characteristic function of concept C) for any object in GT are given by the human expert.

It is very important that during construction of GT, we insert into this table objects representing information about groups of objects which are relevant for the investigated temporal concept (see Figure 35.5). For example, if we are interested in the overtaking maneuver, we compose only vehicle pairs that are close to each other. It means that during construction of GT vehicles are joined on the basis of constraints according to the investigated temporal concept.

Figure 35.5 presents a scheme used for construction of GT for group of two objects (vehicles). Any object is represented by a triple $(P_i^{(A)}, P_i^{(B)}, P_i^{(R)})$, where $P_i^{(j)}$ denotes the j th path in the i th row ($j \in \{A, B, R\}$ and $i \in \{1, \dots, k\}$).

The temporal concepts, defined for group of objects and approximated by classifiers, are nodes of a new graph, which we call as *a behavioral graph for a group of objects*. One can say that the behavioral graph for a group of objects expresses temporal dependencies on a higher level of generalization. On lower level behavioral graphs are expressing temporal dependencies between single objects (or simpler groups of objects).

In Figure 35.6, we present exemplary behavioral graph for two vehicles: vehicle A and vehicle B, related to the standard overtaking pattern. There are six nodes in this graph representing the following temporal concepts: vehicle A is driving behind B on the right lane, vehicle A is changing lanes from right to left, vehicle A is moving back to the right lane, vehicle A is passing B (when A is on the left lane and B is on the right lane), vehicle A is changing lanes from left to right, and vehicle A is before B on the right lane. There are seven connections representing spatiotemporal dependencies between behavioral patterns from nodes. For example, after the node ‘Vehicle A is driving behind B on the right lane’ the behavior of these two vehicles matches to the pattern ‘Vehicle A is changing lanes from right to left and B is driving on the right lane.’

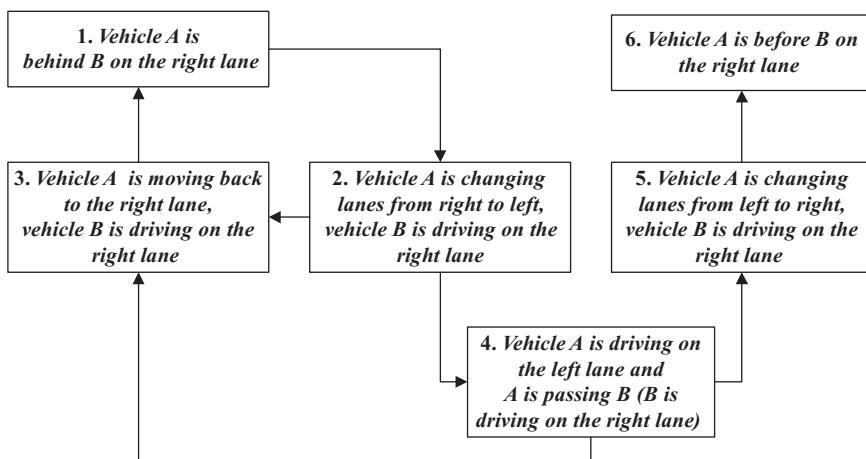


Figure 35.6 A behavioral graph for the maneuver of overtaking

35.7 Behavioral Patterns

For modeling of perception of complex behavior by individual objects or by a group of objects over a long period of time, we construct behavioral graphs to codify our observations (see Sections 35.5 and 35.6). Such graphs facilitate observations about transitions between nodes of behavioral graph and registering a sequence of nodes that form paths of temporal concepts. If the path of temporal concepts matches a path in a behavioral graph, we conclude that the observed behavior is compatible with the behavioral graph. In effect, we can use a behavioral graph as a complex classifier for perception of the complex behavior of individual objects or groups of objects. For this reason, a behavioral graph constructed for some complex behavior can be called as a *behavioral pattern*.

Let us notice that there is the large semantic difference exists among the notion of the temporal pattern and the behavioral pattern. The temporal pattern describes behavior of a complex object in a certain period of time, e.g., in a time window or in a sequence of time windows. Whereas the behavioral pattern describes many variants of object's behaviors or object group's behaviors, which can be observed on various sequences of time windows.

As an example, let us study the behavioral graph for a group of two objects-vehicles (vehicle A and vehicle B) related to the standard overtaking pattern (see Sections 35.5 and Figure 35.6). We can see that the path of temporal patterns with indexes '1, 2, 3, 1, 2, 4' matches a path from this behavioral graph, while the path with indexes '6, 5, 4' does not match any path from this behavioral graph. (This path can match some other behavioral patterns.)

A path of temporal concepts (which makes it possible to identify behavioral patterns) should have a suitable length. In the case where the length is too short, it may be impossible to discern one behavioral pattern from another pattern. For example, we can make a mistake between an overtaking and a passing by a vehicle in traffic.

35.8 Discovering Rules for Fast Identification of Behavioral Patterns

In this section, we present how the method of behavioral patterns identification presented in Section 35.7 can be speeded up using special decision rules. Let us assume that we have a family of behavioral patterns $BP = \{b_1, \dots, b_n\}$ defined for groups of objects (or parts of a given object). For any pattern b_i from the family BP one can construct a complex classifier based on a suitable behavioral graph (see Section 35.7) that makes it possible to answer the question: 'Does the behavior of the investigated group (or the part of a complex object) match the pattern b_i ?' The identification of behavioral patterns of any group is performed by investigation of a time window sequence registered for this group during some period (sometimes quite long). This registration of time windows is necessary if we want to avoid mistakes in identification of the investigated object group. However, in many applications, we are forced to make a faster (often in real-time) decision if some group of objects matches the given behavioral pattern. In other words, we would like to check the investigated group of objects at once, that is, using the first or second time window of our observation only. This is very important from the computational complexity point of view, because if we investigate complex dynamical systems, we usually have to investigate very many groups of objects. Hence, the faster verification of groups can help us optimize the process of searching among groups matching the given behavioral pattern.

The verification of complex objects consisting of some groups of objects can be speeded up by using some special decision rules, which are computed by an algorithm presented below (see also Figure 35.7).

Step 1. Define a family of temporal concepts $TC = \{t_1, \dots, t_m\}$ that have influence on matching of investigated groups to behavioral patterns from family BP (defined on the basis of information from time windows).

Step 2. Construct classifiers for all temporal concepts from TC using the method from Section 35.4.

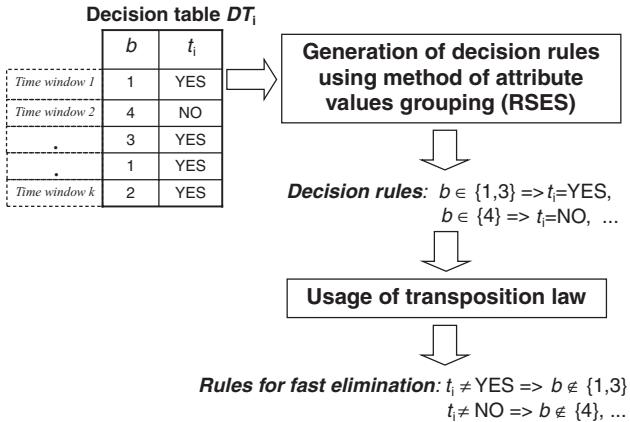


Figure 35.7 The scheme of rules extraction for the fast elimination of behavioral patterns from data tables

Step 3. For any temporal concept t_i from the family TC create a decision table DT_i , which has the following structure:

- Any row of the table DT_i is constructed on the basis of information registered during a period that is typical for the given temporal concept t_i .
- The condition attribute b of table DT_i registers the index of behavioral pattern from the family BP (The index computation is based on observation that any complex classifier from BP can check for the investigated group of objects if there is a sequence of time windows matching the given behavioral pattern and starting from a given time window.)
- The decision attribute of the table DT_i is computed on the basis of values returned by classifier constructed for t_i in previous step.

Step 4. Compute decision rules for DT_i using methods of attribute values grouping that have been developed in the RSES system (see [25]).

Any decision rule computed by the above algorithm expresses a dependency between a temporal concept and the set of behavioral patterns that are not matching this temporal concept. Such rules can make it possible to exclude very fast many parts (groups of objects) of a given complex object as irrelevant for identification of a given behavioral pattern. This is possible because these rules can often be applied at once, that is, after only one time window of our observation. Let us consider a very simple illustrative example. Assume we are interested in the recognition of overtaking that can be understood as a behavioral pattern, defined for the group of two vehicles. Using the methodology presented above, we can obtain the following decision rule:

- If the vehicle A is overtaking B then the vehicle B is driving on the right lane.

After applying the transposition law, we obtain the following rule:

- If the vehicle B is not driving on the right lane then the vehicle A is not overtaking B.

The last rule (see also Figure 35.8) allows us for fast verification whether the investigated group of objects (two vehicles: A and B) is matching the behavioral pattern of overtaking.

Of course, in case of the considered complex dynamical system, there are many other rules that can help us in the fast verification of groups of objects related to the overtaking behavioral pattern. Besides,

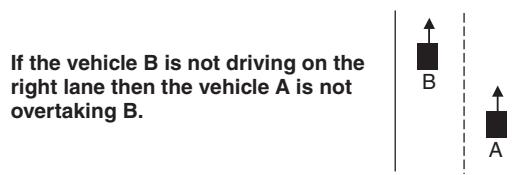


Figure 35.8 The illustration of the rule for fast elimination of behavioral pattern

there are many other behavioral patterns in this complex dynamical system and we have to calculate rules for them using the methodology presented above.

The presented method, which we call as *the method for online elimination of non-relevant for a given behavioral pattern parts of complex object* (ENP method), is not a method for behavioral pattern identification. However, this method allows us to eliminate some paths of a given complex object behavior that are not relevant for checking if this object matches a given behavioral pattern. After such elimination the complex classifiers based on a suitable behavioral graphs should be applied to the remaining parts of the complex object.

35.9 Experimental Results with Behavioral Pattern Identification

To verify the effectiveness of classifiers based on behavioral patterns, we have implemented the algorithms in a *behavioral patterns* library (BP-lib), which is an extension of the RSES-lib 2.1 library forming the computational kernel of the RSES system (see [25, 26]). The experiments have been performed on the data sets obtained from the road simulator (see [17, 18]). We have applied the ‘train-and-test’ method for estimating accuracy. A training set consists of 17,553 objects generated by the road simulator during one thousand of simulation steps. Whereas, a testing set consists of 17,765 objects collected during another (completely different) session with the road simulator.

In our experiments, we compared the quality of three classifiers: *rough set classifier with decomposition* (RS-D), *behavioral pattern classifier* (BP), and *behavioral pattern classifier with the fast elimination of behavioral patterns* (BP-E). For induction of RS-D, we employed RSES system generating the set of minimal decision rules that are used next for classification of situations from the testing data. However, we had to use the method of generating decision rules joined with a standard decomposition algorithm from the RSES system. This was necessary because the size of the training table was too large for the directly generating decision rules (see [25, 26]). The classifiers BP is based on behavioral patterns (see Section 35.7), whilst BP-E are based on behavioral patterns too but with application of fast elimination of behavioral patterns related to the investigated group of objects (see Section 35.8). We compared RS-D, BP, and BP-E using accuracy of classification. Table 35.1 shows the results of applying these classification algorithms for the concept related to the *overtaking* behavioral pattern.

Table 35.1 Results of experiments for the overtaking pattern

Decision class	Method	Accuracy	Coverage	Real accuracy
Yes (overtaking)	RS-D	0.800	0.757	0.606
	BP	0.923	1.0	0.923
	BP-E	0.883	1.0	0.883
No (no overtaking)	RS-D	0.998	0.977	0.975
	BP	0.993	1.0	0.993
	BP-E	0.998	1.0	0.998
All classes (Yes + No)	RS-D	0.990	0.966	0.956
	BP	0.989	1.0	0.989
	BP-E	0.992	1.0	0.992

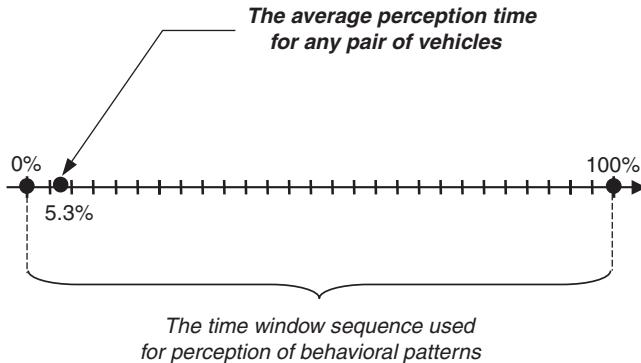


Figure 35.9 The illustration of the average perception time by ENP method of the *no overtaking* behavioral pattern

One can see that in case of perception of the overtaking maneuver (decision class *Yes*) the accuracy and the real accuracy (*real accuracy = accuracy × coverage*) of algorithm BP are higher than the accuracy and the real accuracy of algorithm RS-D for the analyzed data set. Besides, we see that the accuracy of algorithm BP-E is only 4% lower than the accuracy of algorithm BP. Whereas the algorithm BP-E allows us to reduce the time of perception, because during perception we can usually identify the lack of overtaking earlier than in the algorithm BP. This means that it is not necessary to collect and investigate the whole sequence of time windows (that is required in the BP method) but only some first part of this sequence (see Figure 35.9).

In our experiments with the classifier BP-E, it was at an average 5.3% of the whole time sequence window for objects from the decision class *No* (the lack of overtaking in the time window sequence).

35.10 The Automatic Planning for Complex Objects

The prediction of behaviors of a complex object (granules) evaluated over time is usually based on some historical knowledge representation used to store information about changes in relevant features or parameters. This information is usually represented as a data set and has to be collected during long-term observation of a complex dynamical system. For example, in case of the treatment of the infants with respiratory failure (see [8–11]), we associate the object parameters mainly with values of arterial blood gases measurements and the X-ray lung examination. A single action is often not sufficient for changing the complex object in the expected direction. Therefore a sequence of actions needs to be used instead of a single action during medical treatment. Hence, methods of automated planning are necessary during monitoring of a given complex dynamical system (see [10, 11, 27, 28]).

The problem of automated planning can be understood as a standard problem of searching for classifier from a decision table with a complex decision attribute. Values of such decision attribute are plans that should be realized for complex objects or groups of complex objects described by condition attributes.

In this section, we discuss some rough set (see [5]) tools for automated planning as part of a system for modeling networks of classifiers. Such networks are constructed using an ontology of concepts delivered by experts (see, e.g., [23]).

The basic concept we use is a *planning rule*. Let $s_l, s_{r_1} \dots s_{r_k}$ denote states of a complex object and a – an action that causes a transition to some another state. A planning rule proposed by a human expert such as a medical doctor has the following simplified form: $(s_l, a) \rightarrow s_{r_1}|s_{r_2} \dots |s_{r_k}$. Such rule can be used to change the state s_l of a complex object, using the action a to some state from the right-hand side of a rule. But the result of applying such a rule is non-deterministic, because there are usually many states on the right-hand side of a planning rule (see Figure 35.10).

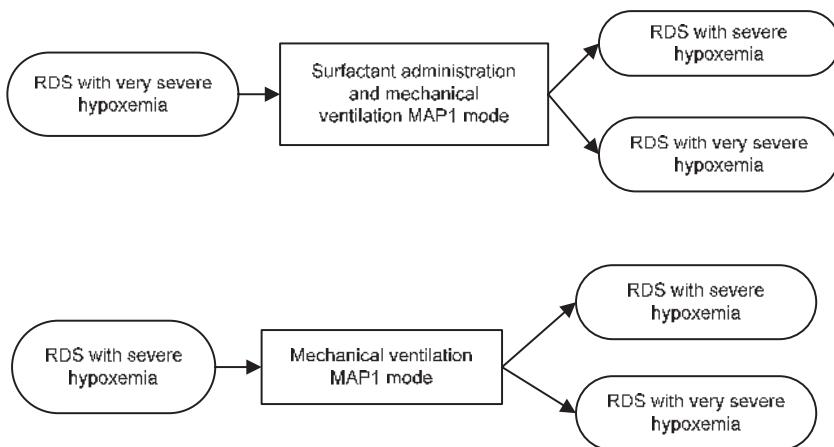


Figure 35.10 Two medical planning rules

A set of planning rules can be represented by a *planning graph*, which can also be interpreted as a complex granule. There are two kinds of nodes in planning graphs: *state nodes* represented by ovals and *action nodes* represented by rectangles (see Figure 35.11).

The links between nodes represent temporal dependencies, e.g., the link between the state node s_1 and the action node a_1 says that in state s_1 of a complex object (e.g., patient), action a_1 can be performed, while the link between a_1 and state node s_3 means that after performing action a_1 in s_1 the status of the complex object (e.g., patient) can be changed from s_1 to s_3 . Figure 35.12 shows how planning rules can be joined to obtain a planning graph.

Let us notice that the essential difference exists between the behavioral graph and the planning graph. There is one kind of nodes in the behavioral graph only, representing properties of behavior of complex objects during certain period (e.g., time window). Whereas there are the following two kinds of nodes in the planning graph, namely, states of complex objects (registered in a time point) and actions, which can be performed on complex objects during some period (e.g., time window). Hence, the main application of behavioral graphs is to represent observed properties of complex objects, while the main application of planning graphs is to represent changes of object's parameters in the expected direction.

Similarly, there is the semantic difference among the elementary action used to define temporal patterns (see Section 35.3) and the action in the planning graph. The elementary action is an elementary change of the parameters of the complex object which was observed in the given moment (time point) of the observation of this object in relation to the previous moment of the observation. Whereas the action in

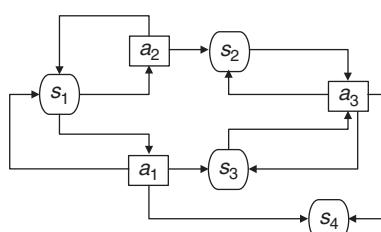


Figure 35.11 A planning graph

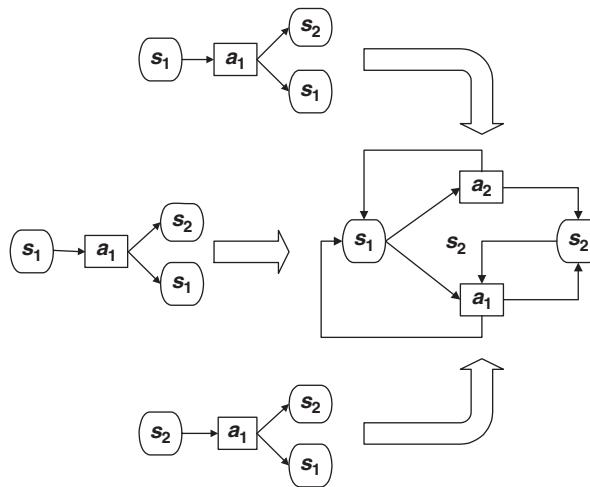


Figure 35.12 From planning rules to a planning graph

the planning graph is a concrete operation executed on the complex object in certain period of the time (during some time window) aiming to change a given state of the complex object.

Any state from the planning graph can be treated as a complex spatiotemporal concept specified by a human expert in natural language and can be approximated by classifiers using data sets and domain knowledge accumulated for a given complex dynamical system. We suggest that the method of temporal concepts approximation presented in Section 35.4 can be used to approximation of states from planning graphs. However, at first sight, the essential difference exists among temporal concepts from the behavioral graph and states from the planning graph. Any temporal concept describes changes of properties of observed complex objects, while any state represents a current status of the observed complex object. However, methods for temporal concept approximation can also be used to approximation of states under a specific interpretation of temporal concepts. We, namely, can assume that the state of the complex object in the given time point is the result of observed changes and behaviors of this object, registered in the time window before the studied time point. We also assume that actions known from the planning graph and executed for the complex object are important causes of behaviors of the complex object. Therefore, every action can be treated as a kind of elementary action performed on objects specified by temporal patterns (see Section 35.3). According to such interpretation, any state of the complex object can be identified with a temporal concept describing parameters of the complex object in the last time point of the investigated time window and can be approximated by classifier constructed in Section 35.4. Such classifiers should be constructed for all states separately.

The output for the planing problem for a single complex object is a path in the planning graph from the initial node state to the *expected (target) node state*. Such a path can be treated as a plan of actions that should be performed beginning from the given complex object in order to change its state to the expected status (see Figure 35.13).

In practice, it is often the case that a generated plan must be compatible with the plan proposed by a human expert (e.g., the treatment plan should be compatible with the plan suggested by human experts from a medical clinic). It is strongly recommended that the method of the verification and evaluation of generated plans should be based on the similarity between the generated plan and the plan proposed by human experts (see Section 35.13). Hence, the usage of special tools that make it possible to resolve conflicts (nondeterminism) of actions in planning rules is needed. (All conflicts should be resolved according to the medical knowledge.) Therefore, in this paper we propose a family of classifiers constructed for all state nodes from a planning graph. These classifiers are constructed on the basis of decision rules computed for a special decision table called a *resolving table* (see Figure 35.14).

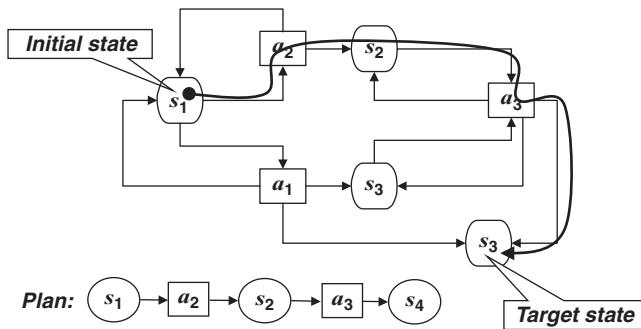


Figure 35.13 The output for the planning problem

The resolving table is constructed for any state node from the planning graph and stores information about objects of a given complex dynamical system satisfying the concept from the current state node. Any row of this table represents information about some history of a single complex object (e.g., patient) satisfying the current state. Condition attributes (features) from this table are defined by human experts and have to be computed on the basis of information included in the description of the current state of a complex object as well on some previous states or actions obtained from the near or far history of an object. It should be emphasized that the definition of such condition attributes should facilitate easy update of attribute values during the construction of a given plan according to performed actions and new states of a complex object. The proposed approach should be accompanied by some kind of simulation during plan construction. The decision attribute of the resolving table is defined as the action that has been performed for a given training object combined with the real effect of this action for an object. Next, we construct rule-based classifiers for all states, i.e., for all corresponding to them resolving tables. In addition, these classifiers make it possible to obtain a list of actions and states after application of actions with their weights in descending order. This is very important in generating plans for groups of objects (see Section 35.11).

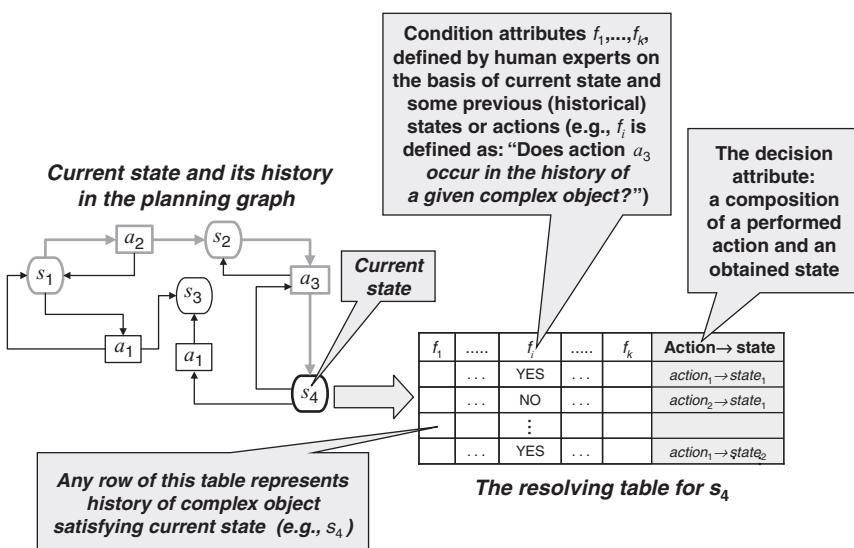


Figure 35.14 The scheme of construction of the resolving table for a given state

35.11 Automatic Planning for Groups of Complex Objects

In this section, we present a generalization of the method for automated planning described in Section 35.10. For a group of objects, we define a graph that we call a *planning graph for a group of objects*. This new graph is similar to a planning graph for a single object (see Section 35.10). There are two kinds of nodes in this graph, namely, *states nodes* (denoted by ovals) that represent the current state of a group of objects specified as complex concepts by a human expert in natural language and *action nodes* (denoted by rectangles) that represent so-called *meta actions* defined for groups of objects by a human expert. Meta actions are performed over a longer period called a *time window* (see e.g., [6]).

In Figure 35.15, we present an exemplary planning graph for a group of four diseases: sepsis, ure-aplasma, RDS and PDA, related to the planning of the treatment of the infant during the respiratory failure (see [9–11] for more details). This graph was created on the basis of analysis of medical data sets (see Section 35.13) supported by human experts.

Notice that any state node from a planning graph for groups of objects can be defined as a complex spatiotemporal concept that is specified by a human expert in natural language. Such concepts can be approximated by classifiers using data sets and domain knowledge accumulated for a given complex dynamical system. Analogously to states from the planning graph for single complex object, any state from the planning graph for group of objects can be approximated as a temporal concept (see Section 35.10). However, the state from the planning graph for group of objects is not treated as the temporal concept for single object, but as the temporal concept for group of objects. Therefore in this case we use the method of approximation from Section 35.6 instead of the method from Section 35.4. As a result, it is possible to recognize the initial state at the beginning of planning for a particular group of complex objects.

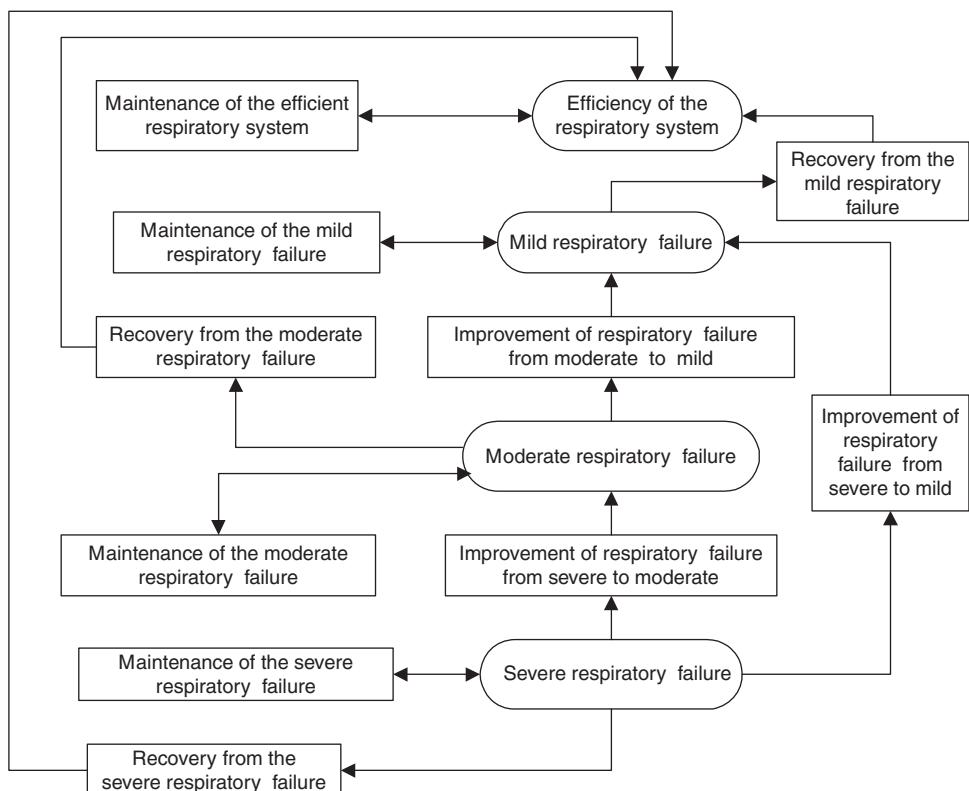


Figure 35.15 A planning graph for the treatment of infants during the respiratory failure

At the beginning of planning for a group of objects, we identify the current state of a group of objects. As mentioned earlier, this can be done by classifiers that have been constructed for all states from the planning graph. Next, we plan a sequence of actions for transforming a group of objects from the current state to the target state (more expected, safer, or more comfortable).

So, our system can propose many plans on the basis of links in a planning graph for groups of objects starting from the current state. Next, the proposed system chooses a plan that seems to be the most effective. However, it is necessary to guarantee that the proposed plan can be realized on the level of any object belonging to a group. In other words, for every object from the group a specific plan should be constructed that leads to a given meta action realization from the level of the group. Besides, all constructed plans for objects belonging to a group should be compatible.

Therefore, during planning a meta action for a group of objects, we use a special tool for verifying the compatibility of plans generated for all members of a group. This verification can be performed by using some special decision rules that we call *elimination rules* (see [10, 11]). Such rules make it possible to eliminate combination of plans that are not compatible relative to domain knowledge. This is possible because elimination rules describe all important dependencies between plans that are joined together. If any combination of plans is not consistent with any elimination rule, then it is eliminated. A set of elimination rules can be specified by human experts or can be computed from data sets. In both of these cases, we need a set of attributes (features) defined for a single plan that are used for the explaining elimination rules. Such attributes are specified by human experts on the basis of domain knowledge and they describe some important features of the plan (generated for single complex object) with respect to proper joining a plan with plans generated for other members of a group.

These features are used as a set of attributes in the special table that we call *an elimination table* (see Figure 35.16). Any row of an elimination table represents information about features of plans assigned for complex objects belonging to an exemplary group of objects from the training data. In other words, any row of this table is a combination of plans assigned for complex objects belonging to an exemplary group of objects. We propose the following method of calculation the set of elimination rules on the basis of the elimination table (see Figure 35.16).

For any attribute from an elimination table, we compute the set of rules treating this attribute as a decision attribute. In this way, we obtain a set of dependencies in the elimination table explained by decision rules. In practice, it is necessary to filter elimination rules to remove the rules with low support

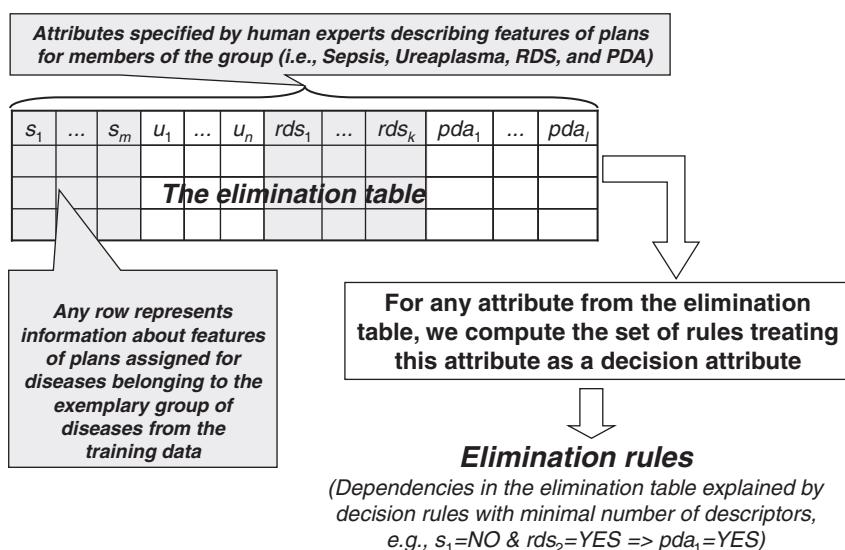


Figure 35.16 The scheme of elimination rules construction

because such rules are not enough general. The resulting set of elimination rules can be used as a filter of inconsistent combinations of plans generated for members of groups. Any combination of plans is eliminated when there exists an elimination rule that is not supported by features of a combination while the combination matches a predecessor of this rule. In other words, a combination of plans is eliminated when the combination matches to the predecessor of some elimination rule and does not match the successor of a rule.

If the combination of plans for members of the group is consistent (it was not eliminated by elimination rules), we should check if the execution of this combination allows us to achieve the expected meta action result from the level of group of objects. This can be done by a special classifier constructed for a table called as an *action table* (see [10, 11]). The structure of an action table is similar to the structure of an elimination table; i.e., attributes are defined by human experts, where rows represent information about features of plans assigned for complex objects belonging to exemplary groups of objects from the training data. In addition, we add to this table a decision attribute. Values of decision attributes represent names of meta actions which will be realized as an effect of the execution of plans described in the current row of a training table. The classifier computed for an action table makes it possible to predict the name of a meta action for a given combination of plans from the level of members of a group. Such classifier we call as *meta-action classifier*. The last step is the selection of combinations of plans that makes it possible to obtain a target meta action with respect to a group of objects. The general view of the meta action planning has been presented on Figure 35.17.

It was mentioned in Section 35.10 that the resolving classifier used for the generation of a next action during the planning for a single object gives us the list of actions (and states after usage of action) with their weights in descending order. This makes it possible to generate many alternative plans for any single object and many alternative combinations of plans for a group of objects. Therefore, the chance of finding an expected combination of plans from a lower level to realize a given meta action (from the higher level) is relatively high.

After planning the selected meta action from the path of action from the planning graph (for a group of objects), the system begins the planning of the next meta action from this path. The planning is stopped when the planning of the last meta action from this path is finished.

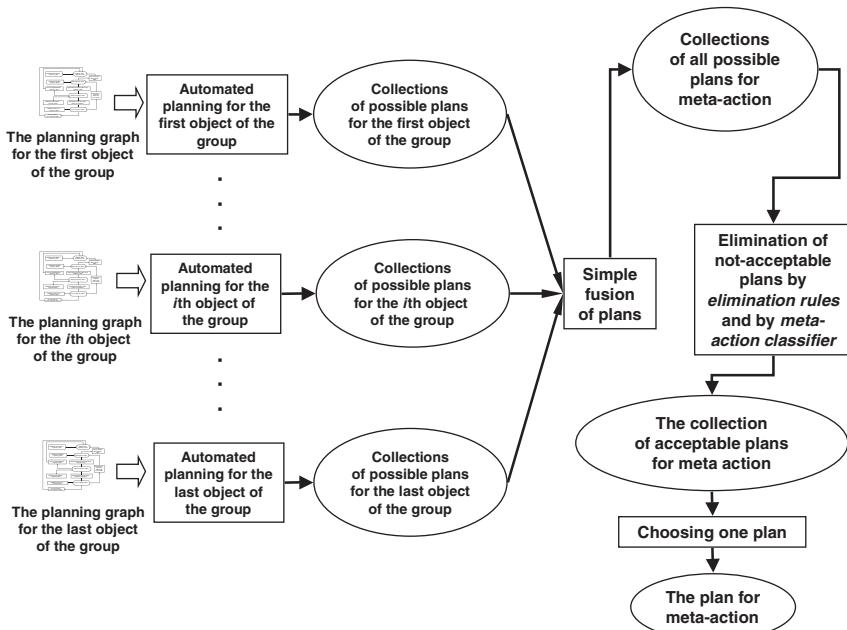


Figure 35.17 The meta action planning

35.12 Estimation of Similarity between Plans

If we want to compare two plans, for example, the plan generated automatically by our computer system and the plan proposed by human expert, we need a tool to estimate the similarity. The problem of inducing classifiers for similarity relations is one of the challenging problem in data mining and knowledge discovery. The existing methods are based on building models for similarity relations based on simple strategies for fusion of local similarities. The optimization of the assumed parameterized similarity formula is performed by tuning parameters relative to local similarities and their fusion. For example, in case of our medical data (see [10, 11]), such formula can compute a similarity between two plans as the arithmetic mean of similarity between all corresponding pairs of actions (nodes) from both plans, where the similarity for the single corresponding pair of actions is defined as a consistency measure of medicines and medical procedures expressed by these actions. For instance, let $M = \{m_1, \dots, m_k\}$ be a set of medicines and P_1 and P_2 are medical plans, where $\mathcal{A} = \{A_1, \dots, A_n\}$ is the set of medical actions from P_1 and $\mathcal{B} = \{B_1, \dots, B_n\}$ is the set of medical actions from P_2 . We define the formula F_s of the similarity among plans in the following way:

$$F_s(P_1, P_2) = \frac{1}{n} \sum_{i=1}^n \frac{\sum_{j=0}^k M_s(A_i, B_i, m_j)}{k},$$

where $M_s : 2^M \times 2^M \times M \longrightarrow \{0, 1\}$ is the function of the similarity of two medical actions in the aspect of the given medicine and it is defined in the following way:

$$M_s(A, B, m) = \begin{cases} 1 & \text{if } m \in A \cap B \vee m \in M \setminus (A \cup B) \\ 0 & \text{otherwise.} \end{cases}$$

For example, if $M = \{m_1, m_2, m_3\}$, $\mathcal{A} = \{\{m_1\}, \{m_1, m_2\}, \{m_1, m_2, m_3\}, \{m_1, m_3\}\}$ and $\mathcal{B} = \{\{m_1\}, \{m_1, m_3\}, \{m_1, m_3\}, \{m_1, m_3\}\}$, then

$$F_s(P_1, P_2) = \frac{1}{4} \cdot \left(\frac{3}{3} + \frac{1}{3} + \frac{2}{3} + \frac{3}{3} \right) = \frac{3}{4}.$$

However, such an approach seems to be very abstract and arbitrary, because it does not take into account the domain knowledge on similarity of plans. According to the domain knowledge, it is quite common that there are many aspects of similarity between plans. Any aspect of the similarity between plans should be taken into account in the specific way and the domain knowledge is necessary for joining all these similarities (obtained for all aspects – see [10, 11]). Therefore, the similarity between plans should be assigned on the basis of a special ontology specified in a dialog with human experts (see [10, 11]). Such ontology we call as *similarity ontology*. Using such similarity ontology we developed methods for inducing hierarchical classifiers predicting the similarity between two plans (generated automatically and proposed by human experts). The method for construction of such classifier can be based on a *table of similarity between plans* (see Figure 35.18). Condition columns of this table represent all concepts from the similarity ontology besides the concept representing the general similarity between plans (the top of similarity ontology). Any row corresponds to a pair of plans: generated automatically and proposed by experts. Values of all attributes have been provided by experts from the set $\{0.0, 0.1, \dots, 0.9, 1.0\}$. Finally, the decision column represents the general similarity between plans. The classifier computed for a table of similarity between plans can be used to determine the similarity between plans generated by our methods of automated planning and plans proposed by human experts during the realistic clinical treatment (see Section 35.13).

Pairs of plans	C_1	C_k	GS (general similarity)
(pa1, pe1)	0.2	0.3	0.1
(pa2, pe2)	0.4	0.5	0.5
(pa3, pe3)	0.2	0.8	0.8
(pa4, pe4)	0.8	0.1	0.2
(pa5, pe5)	0.3	0.2	0.6

Figure 35.18 The scheme of the table of similarity between plans

35.13 Experimental Results with Planning

To verify the effectiveness of our automated planning methods, we have implemented the algorithms in AP-lib, which is an extension of the RSES-lib 2.1 library forming the computational kernel of the RSES system (see [25, 26]).

The experiments have been performed on the medical data sets obtained from Neonatal Intensive Care Unit in Department of Pediatrics, Collegium Medicum, Jagiellonian University, Cracow. The data were collected between 2002 and 2004 using computer database NIS (Neonatal Information System). The detailed information about treatment of 340 newborns is available in the data set, such as perinatal history, birth weight, gestational age, laboratory tests results, imagine techniques results, detailed diagnoses during hospitalization, procedures, and medication were recorded for the each patient. The study group included prematurely born infants with the birth weight ≤ 1500 g, admitted to the hospital before end of the second day of life.

In our experiments, we used one data table extracted from the NIS system, that consists of 11,099 objects. Each object of this table describes parameters of one patient in single time point. There were prepared 7022 situations on the basis of this data table, when the plan of treatment has been proposed by human experts during the realistic clinical treatment.

As a measure of planning success (or failure) in our experiments, we use the special hierarchical classifier that can predict the similarity between two plans as a number between 0.0 (very low similarity between two plans) and 1.0 (very high similarity between two plans). This classifier has been constructed on the basis of the similarity ontology specified by human experts and data sets (see Section 35.12).

A training set consists of 4052 situations (when plans of treatment have been assigned), whereas a testing set consists of 2970 situations, when plans have been generated by automated method and expert plans were known (in order to compare both plans). The average similarity between plans for all tested situations was 0.82.

35.14 Conclusion

We have discussed methods for modeling of compound granules used in behavioral pattern identification and planning tasks. In particular, these are granules used for approximation of vague concepts from domain knowledge supporting these tasks.

In this chapter, we have presented the complete approach to the identification of behavioral patterns. In our approach behavioral patterns have been treated as a complex spatiotemporal concepts (granules) and were approximated by complex granules on the basis of behavioral graphs. In order to test the quality and effectiveness of classifier construction methods based on behavioral patterns, there have been performed experiments on data generated from the road simulator (see, e.g., [6, 18]). The experiments showed that presented in this work algorithmic methods provide very good results of detecting behavioral patterns and may be useful with monitoring of complex dynamical systems.

We also presented an automated planning method of actions performed by complex objects or groups of complex objects. In order to check the effectiveness of suggested automatic planning methods, there were performed experiments concerning planning of treatment of infants suffering from respiratory failure (see [10, 11]). Experimental results showed that the suggested automatic planning method gives good results, also in the opinion of medical experts (compatible enough with the plans suggested by the experts) and may be applied in medical practice as a supporting tool for planning the treatment of infants suffering from respiratory failure.

In our further study we plan to investigate granular-computing-based adaptive methods for behavioral pattern identification and planning tasks.

Acknowledgment

The author thanks professor Andrzej Skowron and the anonymous reviewers for their insights and many helpful comments and corrections for this chapter.

The research has been supported by the grant from Ministry of Science and Higher Education of the Republic of Poland and by the grant Innovative Economy Operational Programme 2007–2013 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland.

References

- [1] P. Doherty, W. Lukaszewicz, A. Skowron, and A. Szałas. *Knowledge Engineering: A Rough Set Approach*. Springer, Heidelberg, Germany, 2006.
- [2] S.H. Nguyen, A. Skowron, and J. Stepaniuk. Granular computing: A rough set approach. *Comput. Intell.* 17(3) (2001) 514–544.
- [3] A. Skowron. Toward intelligent systems: Calculi of information granules. In: T. Terano, T. Nishida, A. Namatame, S. Tsumoto, Y. Ohsawa, and T. Washio (eds), *New Frontiers in Artificial Intelligence, Joint JSAI 2001 Workshop Post-Proceedings*, Vol. 2253 of *Lecture Notes in Artificial Intelligence*, pp. 251–260, Matsue, Japan, May 20–25, 2001. Springer-Verlag.
- [4] A. Skowron. Towards granular multi-agent systems. In: S.K. Pal and A. Ghosh (eds), *Soft Computing Approach to Pattern Recognition and Image Processing*. World Scientific, Singapore, 2002, pp. 215–234.
- [5] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data. Volume 9 of System Theory, Knowledge Engineering and Problem Solving*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.
- [6] J. Bazan, J.F. Peters, and A. Skowron. Behavioral pattern identification through rough set modelling. In: *Proceedings of RSFDGrC'2005*, LNAI, 3641, Springer, Heidelberg, 2005, pp. 688–697.
- [7] J. Bazan. Behavioral pattern identification through rough set modelling. *Fundam. Inf.* 72(1–3) (2006) 37–50.
- [8] J.G. Bazan, P. Kruczak, S. Bazan-Socha, A. Skowron, and J.J. Pietrzyk. Rough set approach to behavioral pattern identification. *Fundam. Inf.* 75(1–4) (2007) 27–47.
- [9] J.G. Bazan, P. Kruczak, S. Bazan-Socha, A. Skowron, and J.J. Pietrzyk. Risk pattern identification in the treatment of infants with respiratory failure through rough set modeling. In: *Proceedings of IPMU'2006*, Paris, France, July 2–7, 2006, pp. 2650–2657.
- [10] J.G. Bazan, P. Kruczak, S. Bazan-Socha, A. Skowron, and J.J. Pietrzyk. Automatic planning based on rough set tools: Towards supporting treatment of infants with respiratory failure. In: *Proceedings of CSP'2006*, Wandlitz, Germany, September 27–29, 2006, pp. 388–399.
- [11] J.G. Bazan, P. Kruczak, S. Bazan-Socha, A. Skowron, and J.J. Pietrzyk. Automatic planning of treatment of infants with respiratory failure through rough set modeling. In: *Proceedings of RSCTC'2006*, LNAI, 4259, Springer, Heidelberg, 2006, pp. 418–427.

- [12] Y. Bar-Yam. *Dynamics of Complex Systems*. Addison Wesley, Reading, MA, 1997.
- [13] C. Urmson *et al.* *High Speed Navigation of Unrehearsed Terrain: Red Team Technology for Grand Challenge*. Report CMU-RI-TR-04-37. The Robotics Institute, Carnegie Mellon University, 2004.
- [14] M. Luck, P. McBurney, and C. Preist. *Agent technology: Enabling Next generation. A roadmap for Agent Based Computing*. University of Southampton, UK, 2003.
- [15] J.F. Peters. Rough Ethology: Towards a Biologically-Inspired Study of Collective Behavior in Intelligent Systems with Approximation Spaces. In: *Transactions on Rough Sets, III*, LNCS, 3400, 2005, 153–174.
- [16] J.F. Peters, C. Henry, and S. Ramanna. Rough ethograms: Study of intelligent system behavior. In: *Proceedings of IIS05*, Gdańsk, Poland, June 13–16, 2005.
- [17] H.S. Nguyen, J. Bazan, A. Skowron, and S.H. Nguyen. Layered learning for concept synthesis. In: *LNCS, 3100, Transactions on Rough Sets, I*. Springer, Heidelberg, Germany, 2004, pp. 187–208.
- [18] The Road Simulator Homepage. logic.mimuw.edu.pl/~bazan/simulator, accessed January 29, 2008.
- [19] J.F. Roddick, K. Hornsby, and M. Spiliopoulou. Yabtsstdmr – yet another bibliography of temporal, spatial and spatiotemporal data mining research. In: R. Uthurusamy and K.P. Unnikrishnan (eds), *SIGKDD Temporal Data Mining Workshop*, ACM Press, pp. 167–175, San Francisco, CA, 2001. Springer-Verlag.
- [20] J.F. Roddick, K. Hornsby, and M. Spiliopoulou. An updated bibliography of temporal, spatial and spatiotemporal data mining research. In: J.F. Roddick and K. Hornsby (eds), *Post-Workshop Proceedings of the International Workshop on Temporal, Spatial and Spatio-Temporal Data Mining TSDM 2001*, Vol. 2007 of *Lecture Notes in Artificial Intelligence*, pp. 147–163, Berlin, 2001. Springer-Verlag.
- [21] T. Poggio and S. Smale. The mathematics of learning: Dealing with data. *Not. Am. Math. Soc.* 5(50) (2003) 537–544.
- [22] L.A. Zadeh. A new direction in AI: Toward a computational theory of perceptions. *AI Mag.* 22(1) (2004) 73–84.
- [23] M. Jarrar. *Towards Methodological Principles for Ontology Engineering*. Ph.D. Thesis, Supervisor: R. Meersman. Vrije Universiteit Brussel (2005).
- [24] J. Bazan and A. Skowron. Classifiers based on approximate reasoning schemes. In: B. Dunin-Keplicz, A. Jankowski, A. Skowron, and M. Szczuka (eds), *Monitoring, Security, and Rescue Tasks in Multiagent Systems MSRAS, Advances in Soft Computing*. Springer, Heidelberg, 2005, pp. 191–202.
- [25] The RSES Homepage. logic.mimuw.edu.pl/~rses, accessed June 14, 2006.
- [26] J. Bazan and M. Szczuka. The rough set exploration system. *Trans. Rough Sets* 3400(3) (2005) 37–56.
- [27] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Elsevier, Morgan Kaufmann, CA, 2004.
- [28] W. Van Wezel, R. Jorna, and A. Meystel. *Planning in Intelligent Systems: Aspects, Motivations, and Methods*. John Wiley & Sons, Hoboken, 2006.

36

Rough Sets and Granular Computing in Hierarchical Learning

Sinh Hoa Nguyen and Hung Son Nguyen

36.1 Introduction

In AI, approximate reasoning is a crucial problem occurring, e.g., during an interaction between two intelligent (human/machine) beings which are using different languages to talk about objects from the same universe. The intelligence skill of those beings (also called agents) is measured by the ability of understanding the other agents. This skill appears in different ways, e.g., as a learning or classification in machine learning and pattern recognition theory, or as an adaptation in evolutionary computation theory.

A great effort of researchers in machine learning and data mining has been made to develop efficient methods for approximation of concepts from data [1]. In a typical process of concept approximation we assume that there is given information consisting of values of conditional and decision attributes on objects from a finite subset (training set, sample) of the object universe and on the basis of this information, one should induce approximations of the concept over the whole universe. Nevertheless, there exist many problems that are still unsolvable for existing state of the art methods because of the high complexity of learning algorithms or even unlearnability of hypothesis spaces.

The existing learning algorithms may be inaccurate if the following issues are not taken under consideration:

- **The target concept is too complex:** It cannot be approximated directly from feature value vectors. Usually the target concept depends on some simpler concepts which may be either approximated from data or given as domain knowledge. The approximation of compound concepts should be somehow synthesized from the approximations of corresponding subconcepts.
- **Low level of readability and interpretation:** The approximation of the target concept which is expressed by feature values is not much human readable. The readability can be improved if it is translated into a higher level description in natural language. For example, the rule ‘*if car speed is high and a distance to a front car is small then a traffic situation is dangerous*’ is more understandable for human user than the ‘IF... THEN...’ rule ‘*if car speed (X) = 200 + ε and a distance to front car (X) = 10 + ε then a traffic situation is dangerous*’.

- **Difficulty of using domain knowledge in the learning process:** In many applications a target concept can be decomposed into simpler components according to the domain knowledge. Each component can be learned separately on a piece of a data set and independent components can be learned in parallel. Moreover, dependencies between component concepts and their consequences can be approximated using domain knowledge and experimental data.

Utilization of domain knowledge into learning process becomes a big challenge for improving and developing more efficient concept approximation methods. In previous papers we have assumed that the domain knowledge was given in form of a concept hierarchy [2, 3]. The concept hierarchy, the simplest form of ontology, is a treelike structure with the target concept located at the root, with attributes located at leaves, and with some additional concepts located in internal nodes. We have adopted the layered learning approach [4] and rough set methods to propose a multilayered algorithm for induction of ‘multilayer rough classifier’ (MLRC) from data [2] using a partial information about the concept hierarchy as a domain knowledge. The main idea is to gradually approximate the target concept from the approximation of other simpler concepts. The learning process can be imagined as a treelike structure (or acyclic graph structure). At the lowest layer, primitive concepts are approximated using feature values available from a data set. This step is sometime called *the information granulation*. At the next layer more complex concepts are synthesized from primitive concepts. This process is repeated for successive layers until the target concept which is located at the highest layer is reached. To distinguish with the layered learning approach presented in [4], the proposed method will be called the hierarchical learning approach.

Therefore, the hierarchical learning approach to concept approximation can be seen as a composition of granular computing and machine learning. The importance of hierarchical concept synthesis is recognized and investigated by many researchers (see, e.g., [5–7]). An idea of hierarchical concept synthesis in the rough mereological and granular computing frameworks has been developed (see, e.g., [5, 8–10]) and problems connected with compound concept approximation are discussed, e.g., in [9, 11–13].

This chapter summarizes the most recent information about application of rough sets and granular computing in hierarchical learning. We present the general framework of rough-set-based hierarchical learning algorithm; we will also discuss some related issues and illustrate our ideas in the corresponding case study problems. In particular, we investigate several strategies of choosing the appropriate learning algorithm for first-level concepts. We also present the method of learning the intermediate concepts and some methods of embedding the domain knowledge into the learning process in order to improve the quality of hierarchical classifiers.

The chapter presents the topics outlined above in the following way. In Section 36.2 we present some basic notions related to the rough set theory and some parameterized methods for approximation of primitive concepts. In Section 36.3 we discuss the main principles of hierarchical learning and present a general schema for concept synthesis based on rough set theory. In Section 36.4 we present extended hierarchical learning scheme for the case when the concept hierarchies are embedded with additional knowledge in a form of relations or constraints among subconcepts. Case studies on application of the proposed methods as well as experimental results are presented in the last section.

36.2 Basic Notions

The problem of concept approximation can be treated as a problem of searching for description (expressible in a given language) of an unknown concept.

Formally, given a universe \mathcal{X} of objects and a concept C , which can be interpreted as a subset of \mathcal{X} , the problem is to find a description of C , which can be expressed in a predefined descriptive language \mathcal{L} . We assume that \mathcal{L} consists of such formulas that are interpretable as subsets of \mathcal{X} . The approximation is required to be as *close* to the original concept as possible.

In this chapter, we assume that objects from \mathcal{X} are described by finite set of attributes (features) $A = \{a_1, \dots, a_k\}$. Each attribute $a \in A$ corresponds to the function $a : \mathcal{X} \rightarrow V_a$, where V_a is called the *domain* of a . For any non-empty set of attributes $B \subseteq A$ and any object $x \in \mathcal{X}$, we define the *B-information vector* of x by $\inf_B(x) = \{(a, a(x)) : a \in B\}$. The set $INF_B(\mathbb{S}) = \{\inf_B(x) : x \in U\}$ is called the

B-information set. The language \mathcal{L} , which is used to describe approximations of the given concept, consists of Boolean expressions over descriptors of the form (*attribute* = *value*) or (*attribute* ∈ *set_of_values*).

Usually, the concept approximation problem is formulated in terms of the *inductive learning problem*, i.e., the problem of searching for a (approximated) description of a concept C based on a *finite set of examples* $U \subset \mathcal{X}$, called the training set. The closeness of the approximation to the original concept can be measured by different criteria like accuracy, description length, etc., which can also be estimated by *test examples*.

The input data for concept approximation problem are given by *decision table*, which is a tuple $\mathbb{S} = (U, A, dec)$, where U is a non-empty, finite set of *training objects*, A is a non-empty, finite set, of *attributes*, and $dec \notin A$ is a distinguished attribute called *decision*. If $C \subset \mathcal{X}$ is a concept to be approximated, then the decision attribute dec is a characteristic function of concept C ; i.e., if $x \in C$, we have $dec(x) = yes$; otherwise, $dec(x) = no$. In general, the decision attribute dec can describe several disjoint concepts. Therefore, without loss of generality, we assume that the domain of the decision dec is finite and equal to $V_{dec} = \{1, \dots, d\}$. For any $k \in V_{dec}$, the set $CLASS_k = \{x \in U : dec(x) = k\}$ is called the *kth decision class of \mathbb{S}* . The decision dec determines a partition of U into decision classes; i.e., $U = CLASS_1 \cup \dots \cup CLASS_d$.

The approximated description of a concept can be induced by any learning algorithm from inductive learning area. In the next section we recall some well-known methods based on layered learning and rough set theory.

36.2.1 Rough Set Approach to Concept Approximation

Let $C \subseteq \mathcal{X}$ be a concept and let $\mathbb{S} = (U, A, dec)$ be a decision table describing the training set $U \subseteq \mathcal{X}$. Any pair $\mathbb{P} = (\mathbf{L}, \mathbf{U})$ is called *rough approximation of C* (see [12, 14]) if it satisfies the following conditions:

1. $\mathbf{L} \subseteq \mathbf{U} \subseteq \mathcal{X}$.
2. \mathbf{L}, \mathbf{U} are expressible in the language \mathcal{L} .
3. $\mathbf{L} \cap U \subseteq C \cap U \subseteq \mathbf{U} \cap U$.
4. \mathbf{L} is maximal and \mathbf{U} is minimal among those \mathcal{L} -definable sets satisfying point (3).

The sets \mathbf{L} and \mathbf{U} are called the *lower approximation* and the *upper approximation* of the concept C , respectively. The set $\mathbf{BN} = \mathbf{U} - \mathbf{L}$ is called the *boundary region of approximation* of C . For objects $x \in \mathbf{U}$, we say that ‘probably, x is in C .’ The concept C is called *rough* with respect to its approximations (\mathbf{L}, \mathbf{U}) if $\mathbf{L} \neq \mathbf{U}$; otherwise, C is called *crisp* in \mathcal{X} .

The condition (4) in the above list can be substituted by inclusion to a degree to make it possible to induce approximations of higher quality of the concept on the whole universe \mathcal{X} . In practical applications the last condition in the above definition can be hard to satisfy. Hence, by using some heuristics we construct suboptimal instead of maximal or minimal sets.

36.2.2 Rough Classifier

The rough approximation of a concept can also be defined by means of a rough membership function. A function $\mu_C : \mathcal{X} \rightarrow [0, 1]$ is called a rough membership function of the concept $C \subseteq \mathcal{X}$ if and only if $(\mathbf{L}_{\mu_C}, \mathbf{U}_{\mu_C})$ is a rough approximation of C , where $\mathbf{L}_{\mu_C} = \{x \in \mathcal{X} : \mu_C(x) = 1\}$ and $\mathbf{U}_{\mu_C} = \{x \in \mathcal{X} : \mu_C(x) > 0\}$ (see [12]). The rough membership function can be treated as a fuzzification of rough approximation. It makes the translation from rough approximation into membership function. The main feature that stands out rough membership functions is related to the fact that it is derived from data. Any algorithm that computes the value of a rough membership function $\mu_C(x)$ having information vector $inf(x)$ of an object $x \in \mathcal{X}$ as an input is called the *rough classifier*.

Rough classifiers are constructed from training decision table. Many methods of construction of rough classifiers have been proposed, e.g., the classical method based on reducts [14, 15], the method based on kNN classifiers [12], or the method based on decision rules [12]. Let us remind the rough-set-based algorithm, called *RS algorithm*, which constructs rough classifiers from decision rules. This method will be improved in the next section.

Note that the proposed concept approximations are not defined uniquely from information about X on the sample U . They are obtained by inducing the concept $X \subseteq \mathcal{U}$ approximations from such information. Hence, the quality of approximations should be verified on new objects and information about classifier performance on new objects can be used to improve gradually concept approximations. Parameterizations delivered by rough membership functions corresponding to classifiers make it possible to discover relevant patterns on the object universe extended by adding new (testing) objects. In the following sections we present illustrative examples of such parameterized patterns. By tuning parameters of such patterns one can obtain patterns relevant for concept approximation of the extended training sample by testing objects.

36.2.3 Case-Based Rough Approximations

For case-based reasoning methods, like kNN (k nearest neighbors) method, a distance (similarity) function between objects $\delta : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}^+$, where \mathbb{R}^+ is the set of nonnegative reals, is defined. The problem of the distance function definition for a given data set is not trivial, but in this chapter, we assume that such similarity function has already been defined for all object pairs.

In case of kNN classification method, the decision for a new object $x \in \mathcal{U} - U$ is made on the basis of decisions on objects from the set $NN(x; k) := \{x_1, \dots, x_k\} \subseteq U$, with k objects from U that are nearest to x with respect to the distance function δ ; i.e., for any object $x_i \in NN(x; k)$ and for any object $u \in U - NN(x; k)$, we have $\delta(x, x_i) \leq \delta(x, u)$. Usually, k is a parameter set up by expert or constructed in experiments from data. The kNN classifiers often use voting algorithm for decision making; i.e., the decision for any new object x is predicted by

$$dec(x) = Voting(\langle n_1(x), \dots, n_d(x) \rangle), \quad (1)$$

where $ClassDist(NN(x; k)) = \langle n_1(x), \dots, n_d(x) \rangle$ is the class distribution of the set $NN(x, k)$; obviously, $n_1(x) + \dots + n_d(x) = k$ for any $x \in U$. The voting function can return the most frequent decision value occurring in $NN(x, k)$; i.e., $dec(x) = i$ if and only if $n_i(x)$ is the largest component value of the vector $\langle n_1(x), \dots, n_d(x) \rangle$. In case of imbalanced data, the vector $\langle n_1(x), \dots, n_d(x) \rangle$ can be scaled with respect to global class distribution first, and after that the voting algorithm can be employed.

The rough approximation based on the set $NN(x; k)$ that is an extension of kNN classifier can be defined as follows. Assume that $0 \leq t_1 < t_2 < k$ and let us consider for i th decision class $CLASS_i \subseteq \mathcal{U}$ a function with parameters t_1, t_2 defined on any object $x \in \mathcal{U}$ by

$$\mu_{CLASS_i}^{t_1, t_2}(x) = \begin{cases} 1 & \text{if } n_i(x) \geq t_2 \\ \frac{n_i(x) - t_1}{t_2 - t_1} & \text{if } n_i(x) \in (t_1, t_2), \\ 0 & \text{if } n_i(x) \leq t_1 \end{cases} \quad (2)$$

where $n_i(x)$ is the i th coordinate in the class distribution $ClassDist(NN(x; k)) = \langle n_1(x), \dots, n_d(x) \rangle$ of $NN(x; k)$.

Let us assume that parameters t_1^o, t_2^o have been chosen in such a way that the above function satisfies for every $x \in U$ the following conditions:

$$\text{if } \mu_{CLASS_i}^{t_1^o, t_2^o}(x) = 1, \text{ then } [x]_A \subseteq CLASS_i \cap U \quad (3)$$

$$\text{if } \mu_{CLASS_i}^{t_1^o, t_2^o}(x) = 0, \text{ then } [x]_A \cap (CLASS_i \cap U) = \emptyset, \quad (4)$$

where $[x]_A = \{y \in U : inf_A(x) = inf_A(y)\}$ denotes the indiscernibility class defined by x relatively to a fixed set of attributes A .

Then the function $\mu_{CLASS_i}^{t_1^o, t_2^o}$ considered on \mathcal{U} can be treated as the rough membership function of the i th decision class. It is the result of induction on \mathcal{U} of the rough membership function of i th decision class restricted to the sample U . The function $\mu_{CLASS_i}^{t_1^o, t_2^o}$ defines a rough approximations $\mathbf{L}_{kNN}(CLASS_i)$ and $\mathbf{U}_{kNN}(CLASS_i)$ of i th decision class $CLASS_i$. For any object $x \in \mathcal{U}$, we have

$$x \in \mathbf{L}_{kNN}(CLASS_i) \Leftrightarrow n_i(x) \geq t_2^o \text{ and } x \in \mathbf{U}_{kNN}(CLASS_i) \Leftrightarrow n_i(x) \geq t_1^o. \quad (5)$$

Certainly, one can consider in conditions (3)–(4) an inclusion to a degree and equality to a degree instead of crisp inclusion and crisp equality. Such degrees parameterize additionally extracted patterns and by tuning them one can search for relevant patters.

As we mentioned above kNN methods have some drawbacks. One of them is caused by assumption that the distance function is defined a priori for all pairs of objects, which is not the case for many complex data sets. In the next section we present an alternative way to define the rough approximations.

36.2.4 Rule-Based Rough Approximations

In this section we describe the rule-based rough set approach to approximations. Let $\mathbb{S} = (U, A, dec)$ be a given decision table. The first step of RS algorithm is construction of some decision rules; i.e., implications of a form

$$\mathbf{r} \underset{df}{\equiv} (a_{i_1} = v_1) \wedge \dots \wedge (a_{i_m} = v_m) \Rightarrow (dec = k), \quad (6)$$

where $a_{ij} \in A$, $v_j \in V_{a_{ij}}$, and $k \in V_{dec}$. Searching for *short, strong* decision rules *with high confidence* from a given decision table is a big challenge for data mining. Some methods based on rough set theory have been presented in [15–17]. Let $\mathbf{RULES}(\mathbb{S})$ be a set of decision rules induced from \mathbb{S} by one of the mentioned rule extraction methods. One can define the rough membership function $\mu_k : \mathcal{X} \rightarrow [0, 1]$ for the concept determined by $CLASS_k$ as follows:

1. For any object $x \in \mathcal{X}$, let $MatchRules(\mathbb{S}, x)$ be the set of rules which are supported by x . Let \mathbf{R}_{yes} be the set of all decision rules from $MatchRules(\mathbb{S}, x)$ for k th class and let \mathbf{R}_{no} be the remainder of \mathbf{R}_{yes} .
2. We define two real values w_{yes} and w_{no} by

$$w_{yes} = \sum_{\mathbf{r} \in \mathbf{R}_{yes}} strength(\mathbf{r}) \text{ and } w_{no} = \sum_{\mathbf{r} \in \mathbf{R}_{no}} strength(\mathbf{r}),$$

where $strength(\mathbf{r})$ is a normalized function, depending on *length, support, confidence* of \mathbf{r} , and some global information about the decision table \mathbb{S} like table size and class distribution (see [12, 17]).

3. The value of $\mu_k(x)$ is defined by these weights as follows (see Figure 36.1):

$$\mu_k(x) = \begin{cases} \text{undetermined} & \text{if } \max(w_{yes}, w_{no}) < \omega \\ 0 & \text{if } w_{no} \geq \max\{w_{yes} + \theta, \omega\} \\ 1 & \text{if } w_{yes} \geq \max\{w_{no} + \theta, \omega\} \\ \frac{\theta + (w_{yes} - w_{no})}{2\theta} & \text{in other cases} \end{cases}$$

Parameters ω and θ should be tuned by the user to control the size of boundary region. They are very important in layered learning approach based on rough set theory.

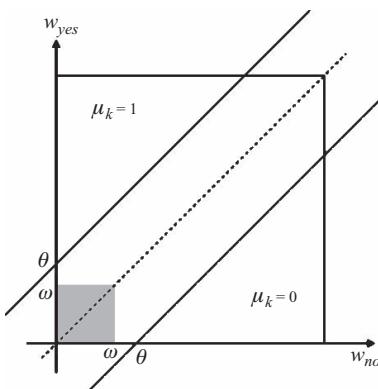


Figure 36.1 An illustration of the rule-based rough membership function

36.2.5 Classifier Roughification

We have so far presented two examples of how to modify a traditional classifier to make it a parameterized rough classifier. The general idea is to change the binary output of the original classifier into a multivalue rough membership function. By this way, rough classifier can be constructed from any classifier including decision tree, neural network, SVM (support vector machine) classifier, etc.

Even though both rough and fuzzy membership functions describe the degree of being an element of a concept, there is an essential difference between them. Unlike fuzzy set theory, the rough membership function is constructed from the training data. Moreover, rough membership functions of compound concepts (e.g., $\mu_{A \cup B}$, $\mu_{A \cap B}$, $\mu_{A \times B}$) are not required to be derived from membership functions of simpler concepts (μ_A and μ_B) by using some predefined t -norm and t -conorm (see [18–20]).

Rough membership functions play a key role in our layered learning approach to approximation of compound concepts.

36.3 Hierarchical Granulation

In this section we present general hierarchical learning scheme for concept synthesizing. We recall the main principles of the layered learning paradigm [4].

1. Layered learning is designed for domains that are too complex for learning a mapping directly from the input to the output representation. The hierarchical learning approach consists of breaking a problem down into several task layers. At each layer, a concept needs to be acquired. A learning algorithm solves the local concept-learning task.
2. Layered learning uses a bottom-up incremental approach to hierarchical concept decomposition. Starting with low-level concepts, the process of creating new subconcepts continues until the high-level concepts, which deal with the full-domain complexity, are reached. The appropriate learning granularity and subconcepts to be learned are determined as a function of the specific domain. The concept decomposition in hierarchical learning is not automated. The layers and concept dependencies are given as a background knowledge of the domain.
3. Subconcepts may be learned independently and in parallel. Learning algorithms may be different for different subconcepts in the decomposition hierarchy. Layered learning is effective for huge data set and it is useful for adaptation when a training set changes dynamically.
4. The key characteristic of hierarchical learning is that each learned layer directly affects the learning at the next layer.

When using the layered learning paradigm, we assume that the target concept can be decomposed into simpler ones called subconcepts. A hierarchy of concepts has a treelike structure. Basic concepts are located at the lowest level and the target concept at the highest level. Basic concepts are learned directly from input data. A higher level concept is composed from existing concepts in lower levels. We assume that a concept decomposition hierarchy is given by domain knowledge [9, 10]. However, one should observe that concepts and dependencies among them represented in domain knowledge are expressed often in natural language. Hence, there is a need to approximate such concepts and such dependencies as well as the whole reasoning process. This issue is directly related to the computing with words paradigm [21, 22] and to rough-neural approach [7], in particular to rough mereological calculi on information granules (see, e.g., [5, 8, 9, 23, 24]).

In this chapter we consider only such concept hierarchies that have a treelike structure. We also assume that input attributes (sensors, measures, ...) are the most basic concepts and should be placed in leaves of the tree, while the decision attribute is the most complex attribute and should be located at the root of the hierarchy. Formally, these can be defined as follows.

Definition 1. By concept hierarchy over a set of attributes A we denote a tuple $\mathcal{H} = (A, D, R)$, where D is a set of concepts $D \cap A \neq \emptyset$, and $R \subset D \times (A \cap D)$ is a directed acyclic graph (DAG) describes the relationship between concepts and attributes.

Each concept C in the concept hierarchy \mathcal{H} together with all its descendants forms a subtree $\mathcal{H}|_C$. We say that the concept C is in the layer h if and only if $h = \text{height}(\mathcal{H}|_C)$. In this way, elements of the concept hierarchy are divided into layers according to the heights of corresponding subtrees. Thus all input attributes should be placed in the ground layer (layer 0), while the decision attribute is on the highest level. Some examples of concept hierarchies are presented in the next section.

We assume that a concept hierarchy \mathcal{H} is given. A training set is represented by decision table $\mathbb{S}_s = (U, A, D)$, where D is a set of decision attributes related to concepts. Decision values indicate whether an object belongs to a concept in the hierarchy. Remind that by *primitive*, *intermediate*, and *target* concepts we denote the concepts that are located at the lowest, the intermediate, and the highest layer of the concept hierarchy, respectively.

36.3.1 Supervised Granulation Framework

The goal of hierarchical learning algorithm is to construct a schema for concept composition. Our method operates from the lowest level to the highest one. The idea is as follows:

Assume that the set of concepts in the i th layer is denoted by $L_i = \{C_1, \dots, C_n\}$. Then each concept C_k is associated with a tuple

$$C_k = (U_k, A_k, O_k, ALG_k, h_k), \quad (7)$$

where

- U_k is the set of training objects used for learning the concept C_k .
- A_k is the set of attributes relevant for learning the concept C_k .
- O_k is the set of outputs used to define the concept C_k .
- ALG_k is the algorithm used for learning the function mapping vector values over A_k into O_k .
- h_k is the hypothesis, which is a result of running the algorithm ALG_k on the training set U_k .

Hypothesis h_k of the concept C_k in a current level directly affects on a next level in the following ways:

1. h_k is used to construct a set of training examples U of a concept C in the next level if C is a direct ancestor of C_k in the decomposition hierarchy.
2. h_k is used to construct a set of features A of a concept C in the next level if C is a direct ancestor of C_k in the decomposition hierarchy.

To construct a hierarchical classifier, for any concept C_k in the concept decomposition hierarchy, one should solve out the following issues:

1. Define a set of training examples U_k used for learning C_k . A training set in the lowest level is subset of an input data set. In a higher level the training set U_k is composed from training sets of subconcepts of C_k .
2. Define an attribute set A_k relevant to expressing the concept C_k . In the lowest level the attribute set A_k is a subset of an available attribute set. In higher levels the set A_k is created from attribute sets of subconcepts of C_k , from an attribute set of input data and/or they are new created attributes. The attribute set A_k is chosen depending on the domain of the concept C_k .
3. Define an output set to describe the concept C_k .
4. Choose an algorithm to learn the concept C_k based on a defined object set and a defined attribute set.

Algorithm 1 summarizes the above-discussed ideas concerning hierarchical learning algorithm.

Algorithm 1. Hierarchical learning schema.

Input: Training data set S , concept decomposition hierarchy H

Output: Schema for target concept composition

```

1: for  $l := 0$  to  $\max Level$  do
2:   for (any concept  $C_k$  in level  $l$ ) do
3:     if  $l = 0$  ( $C_k$  is a primitive concept) then
4:       Create a training set  $U_k \subseteq U$ , where  $U$  is the set of objects in the training set  $S$ ;
5:       Create an attribute set  $A_k \subseteq A$ , where  $A$  is the set of attributes in the training set  $S$ ;
6:     else
7:       Create a training set  $U_k$  from training sets of sub-concepts of  $C_k$ ;
8:       Create an attribute set  $A_k$  from:
9:         - attribute sets of sub-concepts of  $C_k$  and/or
10:        - output sets of sub-concepts of  $C_k$  and/or
11:        - set of attributes in the training set  $S$ ;
12:     end if
13:     Generate the hypothesis  $h_k$  for the concept  $C_k$  using the algorithm  $AGL_k$  defined for  $C_k$ ;
14:     Using the hypothesis  $h_k$  generate output vectors for objects in  $U_k$ ;
15:   end for
16: end for

```

36.3.2 Unsupervised Granulation

The supervised method presented in the previous section is applicable only for data sets with decision attributes (i.e., O_k) for concepts in the hierarchy.

In situation, when our knowledge about the concept C_k is limited to the fact that it is dependent on a set A_k of other concepts in the lower level, we propose to modify the previous algorithm as follows:

- Granulate the sample of objects U_k using a clustering algorithm according to the available information from A_k .
- Define the decision attribute O_k as the membership function of objects to clusters.

In the next section we will describe in detail some important issues that should be settled when applying the hierarchical learning schema in the synthesis of compound concepts. We start the discussion about learning methods for approximation of primitive concepts. Next, we will present some problems related to approximation of intermediate concepts and the target concept and some of their solving strategies based on rough set theory.

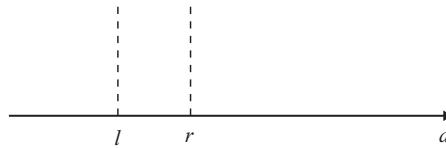


Figure 36.2 The illustration of a soft cut

36.3.3 Granulation Method for Primitive Concepts

Usually, primitive concepts are approximated using input features available from the data set. The choice of the proper algorithm is the most important in this step. In case of supervised learning, using information available from a concept hierarchy for each primitive concept C_b , one can create a training decision system $\mathbb{S}_{C_b} = (U, A_{C_b}, dec_{C_b})$, where $A_{C_b} \subseteq A$ and $dec_{C_b} \in D$. To approximate the concept C_b , one can apply any classical method (e.g., kNN, decision tree, or rule-based approach [25, 26]) to the table \mathbb{S}_{C_b} .

For example, one can use the rule-based reasoning approach proposed in Section 36.2.4 for primitive concept approximation.

Let us point out a special case when a concept is a generalization of another concept. This problem is very intensively investigated in data mining and knowledge discovery from database (KDD) [27]. Many methods have been proposed to create a whole concept hierarchy for one attribute.

In case of real-value attributes, this process is called the discretization. The usual discretization methods define the more general concept by boundary points (cuts).

The rough set approach to granular computing in this situation utilizes the idea of ‘soft cuts’ instead of traditional ‘crisp cuts’ [28].

Definition 2. A soft cut is any triple $p = \langle a, l, r \rangle$ (see Figure 36.2), where $a \in A$ is an attribute, $l, r \in \mathfrak{N}$ are called the left and right bounds of p ($l \leq r$), and the value $\varepsilon = \frac{r-l}{2}$ is called the uncertainty radius of p . We say that a soft cut p discerns pair of objects x_1, x_2 that $a(x_1) < a(x_2)$ if $a(x_1) < l$ and $a(x_2) > r$.

For example, let us consider a real-value attribute *salary* and a more general concept called *salary-group* with three values from the set {low, medium, high}. Traditional discretization methods are searching for two cuts on the attribute *salary*. For example, the cuts $c_1 = 10,000$ and $c_2 = 25,000$ induce the definition $low = [0, 10,000]$, $medium = [10,000, 25,000]$ and $high = [25,000, \infty)$. Using crisp cuts may cause the anomaly situation when two persons are classified by the system into two different classes because one of them gains 10,000 US, while the second gains only 9999 US.

In the soft discretization method, we find out two soft cuts, e.g., $c_1 = [9000, 11,000]$ $c_2 = [24,000, 26,000]$ and define a rough membership functions $\mu_{low}, \mu_{medium}, \mu_{high}$ for general concept *salary-group* according to the proposed soft cuts.

Efficient searching methods for soft cuts were presented in [28, 29].

36.3.4 Granulation Method for Compound Concepts

In this section we present some searching strategies for approximation of concepts that are established on top of already-approximated concepts. This concept composition method is the crucial point in concept synthesis. We will discuss the method that offers an ability to control the level of approximation quality along all the way from primitive concepts to the target concept.

Further discussion will consider such compound concepts on layer i for which we assume that all its subconcepts in the lower layers were already approximated by rough classifiers (see previous sections) derived from relevant decision tables (see Figure 36.3).

To avoid overly complicated notation let us limit our consideration to the case of constructing compound concept approximation on the basis of two simpler concept approximations. Assume we have two concepts C_1 and C_2 that are given to us in the form of rule-based approximations derived from

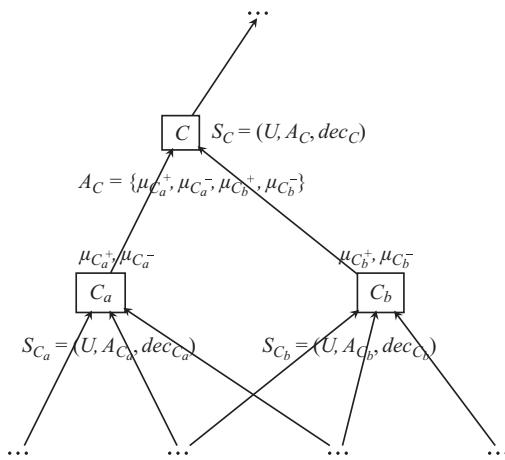


Figure 36.3 The construction of compound concept approximation using rough description of simpler concepts

decision systems $S_{C_1} = (U, A_{C_1}, dec_{C_1})$ and $S_{C_2} = (U, A_{C_2}, dec_{C_2})$. Henceforth, we are given two rough membership functions $\mu_{C_1}(x)$ and $\mu_{C_2}(x)$. These functions are determined with use of parameter sets $\{w_{yes}^{C_1}, w_{no}^{C_1}, \omega^{C_1}, \theta^{C_1}\}$ and $\{w_{yes}^{C_2}, w_{no}^{C_2}, \omega^{C_2}, \theta^{C_2}\}$, respectively. We want to establish similar set of parameters $\{w_{yes}^C, w_{no}^C, \omega^C, \theta^C\}$ for the target concept C , which we want to describe with use of rough membership function μ_C . As previously, the parameters ω and θ controlling of the boundary region are user configurable. But, we need to derive $\{w_{yes}^C, w_{no}^C\}$ from the data. The issue is to define a decision system from which rules used to define approximations can be derived. This problem can be described by *uncertain decision table* as follows:

The uncertain decision system $S_C = (U, A_C, dec_C)$, which is necessary for learning an approximation of concept C , contains conditional attributes $A_C = \{a_{C_1}, a_{C_2}\}$ related to simpler concepts C_1 and C_2 . There are two possibilities of defining the evaluated functions $v_{a_{C_1}}$ and $v_{a_{C_2}}$:

1. By rough membership functions; i.e.,

$$v_{a_{C_1}}(u) = [\mu_{C_1}(x), 1 - \mu_{C_1}(x)].$$

2. By voting weights,

$$v_{a_{C_1}}(u) = [w_{yes}^{C_1}, w_{no}^{C_1}].$$

We propose the following methods for learning approximation of compound concept from uncertain decision tables:

Naive method: One can treat uncertain decision table S_C as a normal decision table S' with more attributes. By extracting rules from S' (using discretization as preprocessing), the rule-based approximations of the concept C are created.

It is important to observe that such rules describing C use attributes that are in fact classifiers themselves. Therefore, in order to have more readable and intuitively understandable description as well as more control over quality of approximation (especially for new cases) it pays to stratify and interpret attribute domains for attributes in A_C .

Stratification method: Instead of using just a value of membership function or weight we would prefer to use linguistic statements such as '*the likeliness of the occurrence of C_1 is low*'. In order to do that we have to map the attribute value sets onto some limited family of subsets. Such subsets

are then identified with notions such us ‘*certain*,’ ‘*low*,’ ‘*high*,’ etc. It is quite natural, especially in case of attributes being membership functions, to introduce linearly ordered subsets of attribute ranges, e.g., {*negative*, *low*, *medium*, *high*, *positive*}. That yields fuzzy-like layout, or linguistic variables, of attribute values. One may (and in some cases should) also consider the case when these subsets overlap.

Stratification of attribute values and introduction of linguistic variable attached to inference hierarchy serves multiple purposes. First, it provides a way for representing knowledge in more human-readable format since if we have a new situation (new object $x^* \in \mathcal{U} \setminus U$) to be classified (checked against compliance with concept C), we may use rules like: *If compliance of x^* with C_1 is high or medium and compliance of x^* with C_2 is high then $x^* \in C$* .

The proposed ideas are gathered in algorithm called *multilayered rough classifier*.

Algorithm 2. Multilayered rough classifier MIRC.

Input: Decision system $\mathbb{S} = (U, A, d)$, concept hierarchy H ;

Output: Schema for concept composition

```

1: for  $l := 0$  to maxLevel do
2:   for (any concept  $C_k$  at the level  $l$  in  $H$ ) do
3:     if  $l = 0$  then
4:        $U_k := U$ ;
5:        $A_k := B$ , where  $B \subseteq A$  is a set relevant to define  $C_k$ 
6:     else
7:        $U_k := U$ 
8:        $A_k = \bigcup O_i$ , for all sub-concepts  $C_i$  of  $C_k$ , where  $O_i$  is the output vector of  $C_i$ ;
9:       Generate a rule set determining of the concept  $C_k$  approximation;
10:      Generate the output vector  $[\mu_{C_k}(x), \mu_{\overline{C_k}}(x)]$  for any object  $x \in U_k$ 
11:    end if
12:   end for
13: end for

```

Another advantage of imposing the division of attribute value sets lays in extended control over flexibility and validity of system constructed in this way. As we may define the linguistic variables and corresponding intervals, we gain the ability of making system more stable and inductively correct. In this way we control the general layout of boundary regions for simpler concepts that contribute to construction of the target concept. The process of setting the intervals for attribute values may be performed by hand, especially when additional background information about the nature of the described problem is available. One may also rely on some automated methods for such interval construction, such as, e.g., clustering, template analysis, and discretization. Some extended discussion on foundations of this approach, which is related to rough-neural computing [7] and computing with words, can be found in [13, 30].

36.3.5 Case Studies

The presented above approach to construction of hierarchical classifier can be treated as a hierarchical process of information granulation. It is a general framework and can be lightly modified for each specific application. We will present some case study problems and consider different assumptions about the input to the concept hierarchy algorithms.

We have implemented the proposed solution on the basis of rough set exploration system (RSES) [31]. To verify a quality of hierarchical classifiers we performed the following experiments.

Nursery Data Set

This is a real-world model developed to rank applications for nursery schools [32]. The taxonomy of concepts is presented in Figure 36.4. The data set consists of 12,960 objects and 8 input attributes which are printed in lowercase.

NURSERY	not_recom, recommend, very_recom, priority, spec_prior
. . EMPLOY	<i>Undefined (employment of parents and child's nursery)</i>
. . . parents	usual, pretentious, great_pret
. . . has_nurs	proper, less_proper, improper, critical, very_crit
. . . STRUCT_FINAN	<i>Undefined (family structure and financial standings)</i>
. . . . STRUCTURE	<i>Undefined (family structure)</i>
. form	complete, completed, incomplete, foster
. children	1, 2, 3, more
. housing	convenient, less_conv, critical
. finance	convenient, inconv
. . . SOC_HEALTH	<i>Undefined (social and health picture of the family)</i>
. . . . social	non-prob, slightly_prob, problematic
. . . . health	recommended, priority, not_recom

Figure 36.4 The taxonomy of concepts in NURSERY data set

Besides the target concept (NURSERY) the model includes four *undefined intermediate concepts*: EMPLOY, STRUCT_FINAN, STRUCTURE, and SOC_HEALTH.

This data set is interesting for our consideration, because the values of intermediate concepts are unknown. We applied a clustering algorithm to approximate intermediate concepts (see Section 36.3.2). Next, we use rule-based algorithm (in RSES system) to approximate the target concept.

Table 36.1 presents experiment results for the traditional (flat) learning method and the proposed layered learning method. Comparing these results, one can observe a significant improvement not only in classification accuracy but also in the compactness of the achieved classifier (the number of rules).

Road Traffic Simulation Data Problem

Learning to recognize and predict traffic situations on the road is the main issue in many unmanned vehicle aircraft projects. It is a good example of hierarchical concept approximation problem. We demonstrate the proposed layered learning approach on our own simulation system.

ROAD SIMULATOR is a computer tool generating data sets consisting of recording used to learn and test complex concept classifiers working on information coming from different devices (sensors) monitoring the situation on the road. The visual effect of the simulation process is presented in Figure 36.5. Let us present some most important features of this system.

During the simulation the system registers a series of parameters of the local simulations, which is simulations connected with each vehicle separately, as well as two global parameters of the simulation that is parameters connected with driving conditions during the simulation. The local parameters are related to driver's profile, which is randomly determined, when a new vehicle appears on the board, and may not be changed until it disappears from the board. The global parameters like visibility, and weather conditions are set randomly according to some scenario. We associate the simulation parameters with the readouts of different measuring devices or technical equipment placed inside the vehicle or in the outside

Table 36.1 Comparison results for Nursery data set achieved on the partition of data into training and test data sets with the ratio 60–40%

	Rule-based classifier using original attributes only	Layered learning using intermediate concepts
Classification accuracy	83.4	99.9%
Coverage	85.3%	100%
Nr of rules	634	42 (for the target concept) 92 (for intermediate concepts)

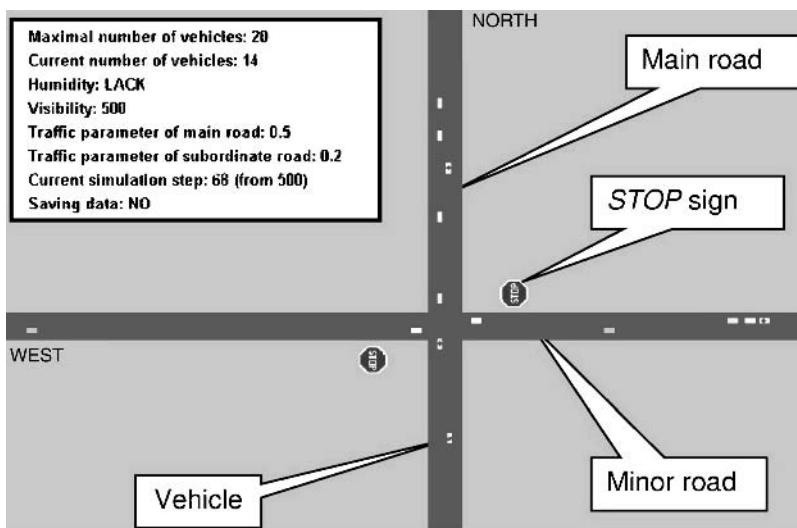


Figure 36.5 The board of simulation

environment (e.g., by the road, in a police car, etc.). Apart from those sensors, the simulator registers a few more attributes, whose values are determined based on the sensor's values in a way determined by an expert. These parameters in the present simulator version take over the binary values and are therefore called concepts. Concepts definitions are very often in a form of a question which one can answer YES, NO, or NULL (does not concern). In Figure 36.6 there is an exemplary relationship diagram for the above-mentioned concepts; we present some exemplary concepts and dependency diagram between these concepts.

During the simulation data may be generated and stored in a text file in the form of a rectangle board (information system). Each line of the board depicts the situation of a single vehicle and the sensors' and

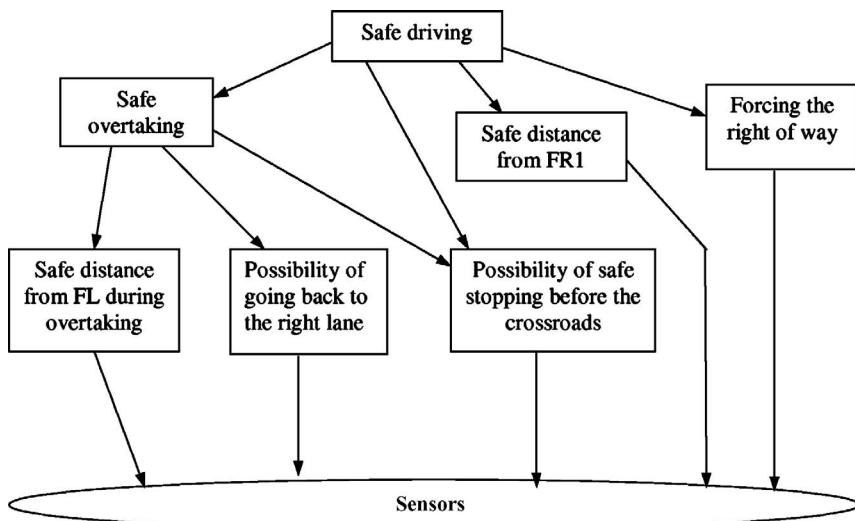


Figure 36.6 The relationship diagram for presented concepts

concepts' values are registered for a given vehicle and its neighboring vehicles. Within each simulation step descriptions of situations of all the vehicles are saved to file.

Experiment setup: We have generated six training data sets: $c10_s100$, $c10_s200$, $c10_s300$, $c10_s400$, $c10_s500$, and $c20_s500$ and six corresponding testing data sets named by $c10_s100N$, $c10_s200N$, $c10_s300N$, $c10_s400N$, $c10_s500N$, and $c20_s500N$. All data sets consists of 100 attributes. The smallest data set consists of above 700 situations (100 simulation units) and the largest data set consists of above 8000 situations (500 simulation units).

We compare the accuracy of two classifiers, i.e., **RS**: the standard classifier induced by the rule set method and **RS-L**: the hierarchical classifier induced by the RS-layered learning method. In the first approach, we employed the RSES system [31] to generate the set of minimal decision rules. We use the simple voting strategy for conflict resolution in new situation classification.

In the RS-layered learning approach, from training table we create five subtables to learn five basic concepts (see Figure 36.6): C_1 : 'safe distance from FL during overtaking,' C_2 : 'possibility of safe stopping before crossroads,' C_3 : 'possibility of going back to the right lane,' C_4 : 'safe distance from FRI,' and C_5 : 'forcing the right of way.'

These tables are created using information available from the concept relationship diagram presented in Figure 36.6. A concept in the next level is C_6 : 'safe overtaking.' To approximate concept C_6 , we create a table with three conditional attributes. These attributes describe fitting degrees of object to concepts C_1 , C_2 , and C_3 , respectively. The decision attribute has three values YES, NO, or NULL, corresponding to the cases of safe overtaking, dangerous overtaking, and not applicable (the overtaking has not been made by car.) The target concept C_7 : 'safe driving' is located in the third level of the concept decomposition hierarchy. To approximate C_7 we also create a decision table with three attributes, representing fitting degrees of objects to the concepts C_4 , C_5 , and C_6 , respectively. The decision attribute has two possible values YES or NO if a car is satisfying global safety condition or not respectively.

The comparison results are performed with respect to the following criteria: (1) accuracy of classification, (2) covering rate of new cases (generality), and (3) computing time necessary for classifier synthesis.

Classification accuracy

Similarly to real-life situations, the decision class 'safe driving = YES' is dominating. The decision class 'safe driving = NO' takes only 4–9% of training sets. Searching for approximation of 'safe_driving = NO' class with the high precision and generality is a challenge of leaning algorithms. In experiments we concentrate on quality of the 'NO' class approximation.

In Table 36.2 we present the classification accuracy of RS and RS-L classifiers. One can observe that the accuracy of 'YES' class of both standard and hierarchical classifiers is high. Whereas accuracy of 'NO' class is very poor, particularly in case of the standard classifier. The hierarchical classifier showed

Table 36.2 Classification accuracy of a standard and hierarchical classifiers

Accuracy	Total		Class YES		Class of NO	
	RS	RS-L	RS	RS-L	RS	RS-L
$c10_s100N$	0.94	0.97	1.0	1.0	0.0	0.0
$c10_s200N$	0.99	0.96	1.0	0.98	0.75	0.60
$c10_s300N$	0.99	0.98	1.0	0.98	0.0	0.78
$c10_s400N$	0.96	0.77	0.96	0.77	0.57	0.64
$c10_s500N$	0.96	0.89	0.99	0.90	0.30	0.80
$c20_s500N$	0.99	0.89	0.99	0.88	0.44	0.93
Average	0.97	0.91	0.99	0.92	0.34	0.63

Table 36.3 Covering rate for standard and hierarchical classifiers

Covering rate	Total		Class YES		Class NO	
	RS	RS-L	RS	RS-L	RS	RS-L
<i>c10_s100N</i>	0.44	0.72	0.44	0.74	0.50	0.38
<i>c10_s200N</i>	0.72	0.73	0.73	0.74	0.50	0.63
<i>c10_s300N</i>	0.47	0.68	0.49	0.69	0.10	0.44
<i>c10_s400N</i>	0.74	0.90	0.76	0.93	0.23	0.35
<i>c10_s500N</i>	0.72	0.86	0.74	0.88	0.40	0.69
<i>c20_s500N</i>	0.62	0.89	0.65	0.89	0.17	0.86
Average	0.62	0.79	0.64	0.81	0.32	0.55

to be much better than the standard classifier for this class. Accuracy of ‘NO’ class of the hierarchical classifier is quite high when training sets reach a sufficient size.

Covering rate

Generality of classifiers is usually evaluated by the recognition ability for unseen objects. In this section we analyze covering rate of classifiers for new objects. One can observe the similar scenarios to the accuracy degree. The recognition rate of situations belonging to ‘NO’ class is very poor in the case of the standard classifier. One can see in Table 36.3 the improvement on coverage degree of ‘YES’ class and ‘NO’ class of the hierarchical classifier.

Computing speed

With respect to time the layered learning approach shows a tremendous advantage in comparison with the standard learning approach. In the case of standard classifier, computational time is measured as a time required for computing a rule set using to decision class approximation. In the case of hierarchical classifier computational time is a total time required for all subconcepts and target concept approximation. One can see in Table 36.4 that speed up ratio of the layered learning approach to the standard one reaches from 40 to 130 times. (All experiments were performed on computer with processor AMD Athlon 1.4GHz, 256MB RAM.)

Rule set size

In this section we consider the complexity of concept description. We approximate concepts by decision rule sets. Hence the size of a rule set is characterized by rule lengths and cardinality of the rule set. Table 36.5 presents the averaged length of decision rules (AL) and the number of decision rules (S) generated by the standard (flat) and the layered learning approaches. One can observe that rules generated by standard approach are long. They consists of more than 40 descriptors on average.

Table 36.4 Time for standard and hierarchical classifier generation

Tables	RS	RS-L	Speed up ratio
<i>c10_s100</i>	94 s	2.3 s	40
<i>c10_s200</i>	714 s	6.7 s	106
<i>c10_s300</i>	1450 s	10.6 s	136
<i>c10_s400</i>	2103 s	34.4 s	60
<i>c10_s500</i>	3586 s	38.9 s	92
<i>c20_s500</i>	10209 s	98 s	104
Average			90

Table 36.5 Description length of the rule sets generated by the standard learning approach (the column ‘Standard’) and the layered learning method (columns $C_1, C_2, C_3, C_4, C_5, C_6, C_7$)

Tables	Standard		C_1		C_2		C_3		C_4		C_5		C_6		C_7	
	AL	S	AL	S	AL	S	AL	S	AL	S	AL	S	AL	S	AL	S
$c10_s100$	34.1	12	5.0	10	5.3	22	4.5	22	4.5	22	1.0	2	2.2	6	3.5	8
$c10_s200$	39.1	45	5.1	16	4.5	27	4.6	41	4.6	42	4.7	14	1.3	3	3.7	13
$c10_s300$	44.7	94	5.2	18	6.6	61	4.1	78	5.2	90	3.4	9	2.4	7	3.6	18
$c10_s400$	42.9	85	7.3	47	7.2	131	4.9	71	6.0	98	4.7	16	2.5	11	3.7	27
$c10_s500$	47.6	132	5.6	21	7.5	101	4.7	87	5.8	146	4.9	15	2.6	8	3.7	30
$c20_s500$	60.9	426	6.5	255	7.7	1107	5.8	249	5.4	554	5.3	25	2.9	16	3.8	35
Average	44.9	—	5.8	—	6.5	—	4.8	—	5.2	—	4.0	—	2.3	—	3.7	—

AL, average length of rules; S, size of the rule set.

One can observe that rules approximating subconcepts are short. The average rule length reaches from 4.0 to 6.5 for the primitive concepts and from 2.0 to 3.7 for the superconcepts. Therefore rules generated by hierarchical learning approach are more understandable and easier to interpret than rules induced by the standard learning approach.

36.4 Hierarchical Learning Enhanced by Additional Constraints

In previous section we have presented a hierarchical learning algorithm for approximation of compound concept, assuming that a hierarchy of concepts is given as domain knowledge. One can observe that concept hierarchy represents a set of concepts and a binary relation which connects a ‘child’ concept with its ‘parent.’ This model facilitates the user to represent his/her knowledge about relationships between input attributes and target concepts. If no such information is available, one can assume the flat hierarchy with the target concept on top and all attributes in the downstairs layer.

In many real-life problems, besides the ‘child–parent’ relations, different relations between concepts may exist. Utilization of this kind of domain knowledge can improve the quality of classifiers. In this section, we discuss how to extend the basic hierarchical learning scheme to the case when concept hierarchies are embedded with additional knowledge in a form of relations or constraints among subconcepts.

36.4.1 Knowledge Representation Model with Constraints

Recall that concept hierarchy represents a set of concepts and a binary relation which connects a ‘child’ concept with its ‘parent.’ The most important relation types are the subsumption relations (written as ‘is-a’ or ‘is-part-of’), defining which objects (or concepts) are members (or parts) of another concepts in the hierarchy. Besides the ‘child–parent’ relations, we consider new kinds of relations associating with concepts in the hierarchy. We call them *domain-specific constraints*. We consider two types of constraints: (1) constraints describing relationships between a ‘parent’ concept and its ‘child’ concepts, and (2) constraints connecting the ‘sibling’ concepts (having the same parent).

Formally, the extended concept hierarchy is a triple $\mathcal{H} = (\mathbb{C}, \mathbb{R}, \text{Constr})$, where $\mathbb{C} = \{C_1, \dots, C_n\}$ is a finite set of concepts including primitive concepts (attributes), intermediate concepts, and the target concept; $\mathbb{R} \subseteq \mathbb{C} \times \mathbb{C}$ is child–parent relation in the hierarchy; and Constr is a set of constraints. In this chapter, we consider constraints expressed by association rules of the form $\mathbf{P} \rightarrow_{\alpha} \mathbf{Q}$, where

- \mathbf{P}, \mathbf{Q} are Boolean formulas over the set $\{c_1, \dots, c_n, \bar{c}_1, \dots, \bar{c}_n\}$ of Boolean variables corresponding to concepts from \mathbb{C} and their complements.
- $\alpha \in [0, 1]$ is the confidence of this rule.

36.4.2 Extended Hierarchical Learning Algorithm

Let us assume that an extended concept hierarchy $\mathcal{H} = (\mathbb{C}, \mathbb{R}, \text{Constr})$ is given. For compound concepts in the hierarchy, we can use the rough classifiers as a building block to develop a multilayered classifier. Basic multilayered classifier was presented in Section 36.3.4. The main idea can be summarized as follows:

Let $\text{prev}(C) = \{C_1, \dots, C_m\}$ be the set of concepts, which are connected with C in the hierarchy. The rough approximation of the concept C , can be determined by performing two steps: (1) construct a decision table $\mathbb{S}_C = (U, A_C, dec_C)$ relevant for the concept C , and (2) induce a rough classifier for C using decision table \mathbb{S}_C . In the previous section (Section 36.3), the training table $\mathbb{S}_C = (U, A_C, dec_C)$ was constructed as follows:

- The set of objects U is common for all concepts in the hierarchy.
- $A_C = h_{C_1} \cup h_{C_2} \cup \dots \cup h_{C_m}$, where h_{C_i} is the output of the hypothetical classifier for the concept $C_i \in \text{prev}(C)$. If C_i is an input attribute $a \in A$, then $h_{C_i}(x) = \{a(x)\}$; otherwise, $h_{C_i}(x) = \{\mu_{C_i}(x), \mu_{\overline{C_i}}(x)\}$.

Repeating these steps for each concept through the bottom to the top layer we obtain a ‘hybrid classifier’ for the target concept, which is a combination of classifiers of various types. In the second step, the learning algorithm should use the decision table $\mathbb{S}_C = (U, A_C, dec_C)$ to ‘resolve conflicts’ between classifiers of its children.

One can observe that, if sibling concepts C_1, \dots, C_m are independent, the membership function values of these concepts are ‘sent’ to the ‘parent’ C , without any correction. Thus the membership value of weak classifiers may disturb the training table for the parent concept and cause the misclassification when testing new unseen objects. We present two techniques that enable the expert to improve the quality of hybrid classifiers by embedding their domain knowledge into learning process.

Using Constraints to Refine Weak Classifiers

Let $R := c_1 \wedge c_2 \wedge \dots \wedge c_k \rightarrow_\alpha c_0$ be a sibling–sibling constraint connecting concepts C_1, \dots, C_k with the concept C_0 . We say that the constraint R fires for C_0 if

- Classifiers for concepts C_1, \dots, C_k are strong (of a high quality).
- Classifier of concepts C_0 is weak (of a low quality).

The refining algorithm always starts with the weakest classifier for which there exist a constraint that fires. The refining algorithm is presented in Algorithm 3.

Algorithm 3. Classifier refining.

Input: classifier $h(C_0)$, constraint $R := c_1 \wedge c_2 \dots \wedge c_k \rightarrow_\alpha c_0$

Output: Refined classifier $h(C_0)$

```

1: for each object  $x \in U$  do
2:   if  $x$  are recognized by classifiers of  $C_1, \dots, C_k$  with high degree then
3:     if  $c_0$  is a positive literal then
4:        $\mu_{C_0}(x) := \alpha \cdot \min\{\mu_{C_1}(x), \mu_{C_2}(x), \dots, \mu_{C_k}(x)\}; \quad \mu_{\overline{C_0}}(x) := 1 - \mu_{C_0}(x);$ 
5:     else  $\{c_0$  is a negative literal $\}$ 
6:        $\mu_{\overline{C_0}}(x) := \alpha \cdot \min\{\mu_{C_1}(x), \mu_{C_2}(x), \dots, \mu_{C_k}(x)\}; \quad \mu_{C_0}(x) := 1 - \mu_{\overline{C_0}}(x)$ 
7:     end if
8:   end if
9:    $h'(C_j) := (\mu_{C_0}, \mu_{\overline{C_0}});$ 
10: end for
```

Using Constraints to Select Learning Algorithm

Another problem is how to assign a suitable approximation algorithm for an individual concept in the concept hierarchy? In the previous papers [2] the type of approximation algorithm (kNN, decision tree,

or rule set) for each concept was settled by the user. In this chapter we show that the constraints can be treated as a guide to semiautomatic selection of best learning algorithms for concepts in the hierarchy.

Assume that there is a ‘children–parent’ constraints: $\bigwedge_i c_i \rightarrow_{\alpha} p$ (or $\bigwedge_i c_i \rightarrow_{\alpha} \bar{p}$) for a concept $P \in \mathbb{C}$. The idea is to choose the learning algorithm that maximizes the confidence of constraints connecting P ’s children with himself. Let **RS_ALG** be a set of available parameterized learning algorithms, we define an objective function $\Psi_P : \mathbf{RS_ALG} \rightarrow \mathbb{R}^+$ to evaluate the quality of algorithms. For each algorithm $\mathbf{L} \in \mathbf{RS_ALG}$, the value of $\Psi_P(\mathbf{L})$ is dependent on two factors:

- classification quality of $\mathbf{L}(\mathbb{S}_P)$ on a validation set of objects;
- confidence of the constraints $\bigwedge_i c_i \rightarrow_{\alpha} p$.

The function $\Psi_P(\mathbf{L})$ should be increasing w.r.t. quality of the classifier $\mathbf{L}(\mathbb{S}_P)$ for the concept P (induced by \mathbf{L}) and the closeness between the real confidence of the association rule $\bigwedge_i c_i \rightarrow p$ and the parameter α . The function Ψ can be used as an objective function to evaluate a quality of approximation algorithm.

A complete scheme of hierarchical learning algorithm with concept constraints is presented in Algorithm 4.

Algorithm 4. Multilayered rough classifier using constraints.

Input: Decision system $\mathbb{S} = (U, A, d)$, extended concept hierarchy $\mathcal{H} = (\mathbb{C}, \mathbb{R}, Constr)$; a set **RS_ALG** of available approximation algorithms

Output: Schema for concept composition

```

1: for  $l := 0$  to max_Level do
2:   for (any concept  $C_k$  at the level  $l$  in  $\mathcal{H}$ ) do
3:     if  $l = 0$  then
4:        $U_k := U$ ;
5:        $A_k := B$ , where  $B \subseteq A$  is a set relevant to define  $C_k$ 
6:     else
7:        $U_k := U$ ;
8:        $A_k = \bigcup O_i$ , for all  $C_i \in prev(C_k)$ , where  $O_i$  is the output vector of  $C_i$ ;
9:       Choose the best learning algorithm  $\mathbf{L} \in \mathbf{RS\_ALG}$  with a maximal objective function  $\Psi_{C_k}(\mathbf{L})$ 
10:      Generate a classifier  $H(C_k)$  of concept  $C_k$ ;
11:      Refine a classifier  $H(C_k)$  using a constraint set Constr.
12:      Send output signals  $O_k = \{\mu_C(x), \mu_{\bar{C}}(x)\}$  to the parent in a higher level.
13:    end if
14:  end for
15: end for
```

36.4.3 Case Studies

To verify a quality of a hierarchical learning approach, we have implemented both algorithms: the basic hierarchical learning algorithm presented in Section 36.3.4 and the extended learning algorithm presented in Section 36.4.2.

A first group of experiments is carried out on artificial data sets generated by a traffic road simulator. The purpose of the experiments is to examine the effectiveness of the hierarchical learning approach by comparing it with the flat learning approach. Quality of the new approach is verified using the following criteria: generality of concept approximation, preciseness of concept approximation, computation time required for inducing of concept approximation, and concept description length.

A second group of experiments is concerned with real digital sunspot images obtained by NASA SOHO/MDI satellite [33]. The purpose of the experiments is to examine the effectiveness of hierarchical learning enhanced by additional domain knowledge. We compare effectiveness of extended vs. basic hierarchical learning approach.

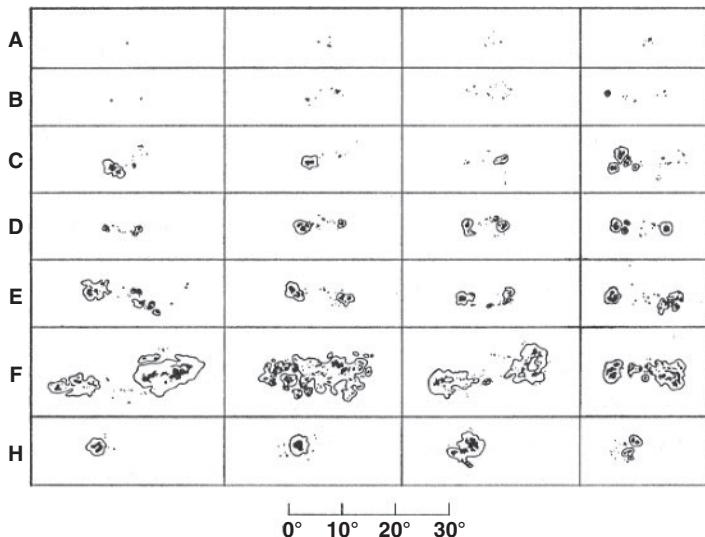


Figure 36.7 Possible visual appearances for each class. There is a wide allowable margin in the interpretation of the classification rules making automatic classification difficult

Experimental result: Sunspot Classification Problem

Sunspots are the subject of interest to many astronomers and solar physicists. Sunspot observation, analysis, and classification form an important part of furthering the knowledge about the Sun. Sunspot classification is a manual and very labor intensive process that could be automated if successfully learned by a machine. The main goal of the first attempt to sunspot classification problem is to classify sunspots into one of the seven classes $\{A, B, C, D, E, F, H\}$, which are defined according to the McIntosh/Zurich sunspot classification scheme. More detailed description of this problem can be found in [34].

The data were obtained by processing NASA SOHO/MDI satellite images to extract individual sunspots and their attributes characterizing their visual properties like size, shape, and positions. The data set consists of 2589 observations from the period September 2001 to November 2001. The main difficulty in correctly determining sunspot groups concerns the interpretation of the classification scheme itself. There is a wide allowable margin for each class (see Figure 36.7). Therefore, classification results may differ between different astronomers doing the classification.

Now, we will present the application of the proposed approach to the problem of sunspot classification.

In [3], we have presented a method for automatic modeling the domain knowledge about sunspots concept hierarchy. The main part of this ontology is presented in Figure 36.8.

We have shown that rough membership function can be induced using different classifiers, e.g., kNN, decision tree, or decision rule set. The problem is to choose the proper type of classifiers for every node of the hierarchy. In experiments with sunspot data, we applied the rule-based approach for concepts in the lowest level, decision-tree-based approach for the concepts in the intermediate levels, and the nearest-neighbor-based approach for the target concept.

Figure 36.9 (left) presents the classification accuracy of ‘hybrid classifier’ obtained by composition of different types of classifiers and ‘homogenous classifier’ obtained by composition of one type of classifiers. The first three bars show qualities of homogenous classifiers obtained by composition of kNN classifiers, decision tree classifiers, and rule-based classifiers, respectively. The fourth bar (the gray one) of the histogram displays the accuracy of the hybrid classifier.

The use of constraints also give a profit. In our experiment, these constraints are defined for concepts at the second layer to define the training table for the target concept *AllClasses*. It is because the noticeable breakdown of accuracy has been observed during experiments. We use the strategy proposed in Section 36.4 to settle the final rough membership values obtained from its children *A-H-B-C-DEF, D-EF*,

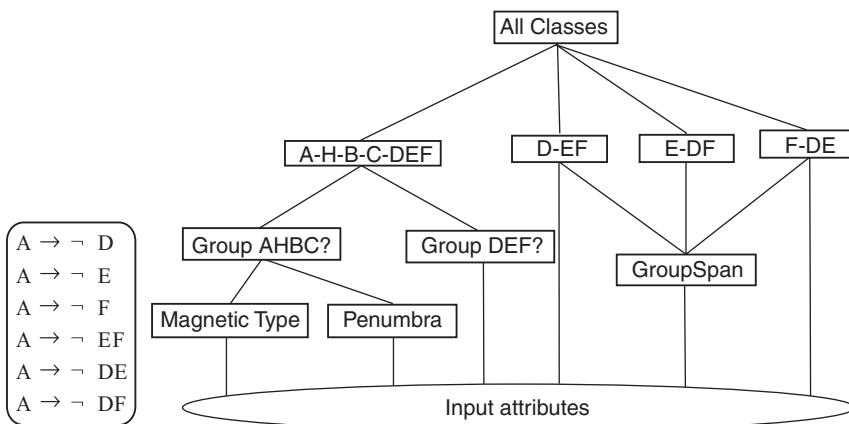


Figure 36.8 The concept hierarchy for sunspot recognition problem

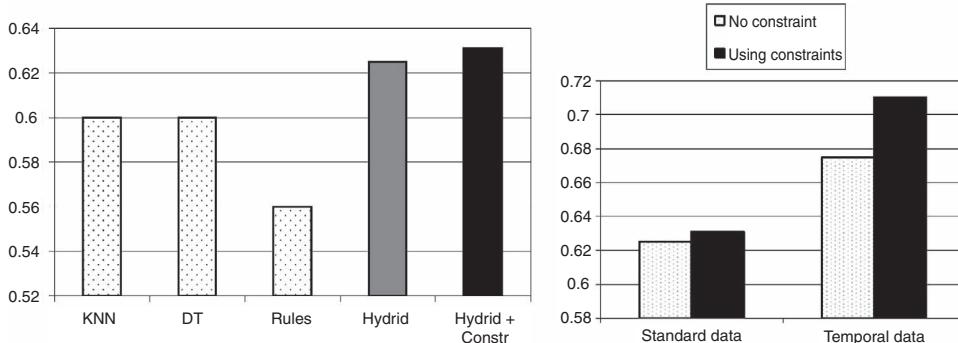


Figure 36.9 Accuracy comparison of different hierarchical learning methods

E-DF, and *F-DE* (see the concept hierarchy). One can observe that using constraints we can promote good classifiers in a composition step. A better classifier has higher priority in a conflict situation. The experiment results are shown in Figure 36.9. The gray bar of the histogram displays the quality of the classifier induced without concept constraints and the black bar shows the quality of the classifier generated using additional constraints.

Another approach to manage with sunspot recognition problem is related to temporal features. Comparative results are shown in Figure 36.9 (right). The first two bars in the graph describe the accuracy of classifiers induced *without* temporal features and the last two bars display the accuracy of classifiers induced *with* temporal features. One can observe a clear advantage of the last classifiers over the first ones. The experimental results also show that the approach for dealing with temporal features and concept constraints considerably improves approximation quality of the complex groups such as *B*, *D*, *E*, and *F*.

36.5 Conclusion

We presented a new method for concept synthesis. It is based on the hierarchical learning approach. Unlike traditional approach, hierarchical learning methods induce the approximation of concepts not only from accessible data but also from the domain knowledge given by experts. In this chapter, we assume that knowledge is represented by a concept taxonomy which is in fact a hierarchical description of dependency between the target concept and input attributes through additional intermediate concepts. The hierarchical

learning approach showed to be promising for the compound concept synthesis. Experimental results with road traffic simulation are showing advantages of this new approach in comparison to the standard learning approach. The main advantages of the hierarchical learning approach can be summarized as follows:

1. high precision of concept approximation,
2. high generality of concept approximation,
3. simplicity of concept description,
4. high computational speed,
5. possibility of localization subconcepts difficult to approximate. It is an important information because it is specifying a task on which we should concentrate to improve the quality of the target concept approximation.

In this chapter, besides a concept-dependency hierarchy we have also considered additional domain knowledge in the form of concept constraints. We proposed an approach to deal with some forms of concept constraints. Experimental results with sunspot classification problem have shown advantages of these new approaches in comparison to the standard learning approach.

Acknowledgment

The research has been partially supported by the grant N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland and by the grant Innovative Economy Operational Programme 2007–2013 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland and the research grant of Polish-Japanese Institute of Information Technology.

References

- [1] W. Kloesgen and J. Źytkow (eds). *Handbook of Knowledge Discovery and Data Mining*. Oxford University Press, Oxford, 2002.
- [2] S.H. Nguyen, J. Bazan, A. Skowron and H.S. Nguyen. Layered Learning for Concept synthesis. In: J.F. Peters, A. Skowron, J.W. Grzymala-Busse, B. Kostek, R.W. Swiniarski, and M.S. Szczuka (eds), *Transactions on Rough Sets I*, volume LNCS 3100 of *Lecture Notes on Computer Science*. Springer, Berlin, Heidelberg, 2004, pp. 187–208.
- [3] S.H. Nguyen, T.T. Nguyen, and H.S. Nguyen. Rough set approach to sunspot classification problem. In: D. Ślęzak, J. Yao, J.F. Peters, W. Ziarko, and X. Hu (eds), *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 10th International Conference, RSFDGrC 2005, Regina, Canada, August 31–September 3, 2005, Proceedings, Part II*, LNCS 3642. Springer-Verlag, Berlin, Heidelberg, 2005, pp. 263–272.
- [4] P. Stone. *Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer*. The MIT Press, Cambridge, MA, 2000.
- [5] L. Polkowski and A. Skowron. Rough mereology: A new paradigm for approximate reasoning. *Int. J. Approx. Reason.* 15(4) (1996) 333–365.
- [6] T. Poggio and S. Smale. The mathematics of learning: Dealing with data. *Not. AMS* 50(5) (2003) 537–544.
- [7] S.K. Pal, L. Polkowski, and A. Skowron (eds). *Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies*. Springer-Verlag, Heidelberg, Germany, 2003.
- [8] L. Polkowski and A. Skowron. Towards adaptive calculus of granules. In: L.A. Zadeh and J. Kacprzyk (eds), *Computing with Words in Information/Intelligent Systems*. Physica-Verlag, Heidelberg, Germany, 1999, pp. 201–227.
- [9] A. Skowron and J. Stepaniuk. Information granules and rough-neural computing. In: S.K. Pal, L. Polkowski, and A. Skowron (eds), *Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies*. Springer-Verlag, Heidelberg, Germany, 2003, pp. 43–84.
- [10] A. Skowron. Approximate reasoning by agents in distributed environments. In: N. Zhong, J. Liu, S. Ohsuga, and J. Bradshaw (eds), *Intelligent Agent Technology Research and Development: Proceedings of the 2nd Asia-Pacific*

- Conference on Intelligent Agent Technology IAT01, Maebashi, Japan, October 23–26, 2001.* World Scientific, Singapore, 2001, pp. 28–39.
- [11] A. Skowron. Approximation spaces in rough neurocomputing. In: M. Inuiguchi, S. Tsumoto, and S. Hirano (eds), *Rough Set Theory and Granular Computing*, Vol. 125 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag, Heidelberg, Germany, 2003, pp. 13–22.
 - [12] J. Bazan, H.S. Nguyen, and M. Szczuka. A view on rough set concept approximations. *Fundam. Inf.* 59(2–3) (2004) 107–118.
 - [13] A. Skowron and M. Szczuka. Approximate reasoning schemes: Classifiers for computing with words. In: *Proceedings of SMPS 2002, Advances in Soft Computing*, Heidelberg, Canada, 2002, Springer-Verlag, pp. 338–345.
 - [14] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*, Vol. 9 of *System Theory, Knowledge Engineering and Problem Solving*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.
 - [15] Z. Pawlak and A. Skowron. A rough set approach for decision rules generation. In *Thirteenth International Joint Conference on Artificial Intelligence IJCAI*, Chambéry, France, 1993. Morgan Kaufmann Publishers, San Mateo, CA, 1993, pp. 114–119.
 - [16] J. Grzymala-Busse. A new version of the rule induction system lers. *Fundam. Inf.* 31(1) (1997) 27–39.
 - [17] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron. Rough sets: A tutorial. In: S.K. Pal and A. Skowron (eds), *Rough Fuzzy Hybridization: A New Trend in Decision-Making*. Springer-Verlag, Singapore, 1999, pp. 3–98.
 - [18] P. Hajek. Metamathematics of fuzzy logic. *Trends in Logic: Studia Logica Library*, Vol. 4. Kluwer Academic Publishers, Dordrecht, 1998.
 - [19] M.R. Klement and E. Pap. Triangular norms. *Trends in Logic: Studia Logica Library*, Vol. 8. Kluwer Academic Publishers, Dordrecht, 2000.
 - [20] L.A. Zadeh. Fuzzy sets. *Inf. Control* 8 (1965) 338–353.
 - [21] L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Trans. Fuzzy Syst.* 4 (1996) 103–111.
 - [22] L.A. Zadeh. A new direction in AI: Toward a computational theory of perceptions. *AI Mag.* 22(1) (2001) 73–84.
 - [23] L. Polkowski and A. Skowron. Rough mereological calculi of granules: A rough set approach to computation. *Comput. Intell.* 17(3) (2001) 472–492.
 - [24] A. Skowron and J. Stepaniuk. Information granules: Towards foundations of granular computing. *Int. J. Intell. Syst.* 16(1) (2001) 57–86.
 - [25] J.H. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, Heidelberg, Germany, 2001.
 - [26] T. Mitchell. *Machine Learning*. McGraw-Hill, Boston, 1998.
 - [27] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2000.
 - [28] H.S. Nguyen. A soft decision tree. In: M.A. Kłopotek, S. Wierzchon, and M. Michalewicz (eds), *Intelligent Information Systems 2002 (Proc. IIS'2002)*, *Advances in Soft Computing*, Springer, Berlin, Heidelberg, 2002, pp. 57–66.
 - [29] H.S. Nguyen. On efficient handling of continuous attributes in large data bases. *Fundam. Inf.* 48(1) (2001) 61–81.
 - [30] A. Skowron and J. Stepaniuk. Information granule decomposition. *Fundam. Inf.* 47(3–4) (2001) 337–350.
 - [31] J.G. Bazan and M. Szczuka. RSES and RSESlb – a collection of tools for rough set computations. In: W. Ziarko and Y. Yao (eds), *Second International Conference on Rough Sets and Current Trends in Computing RSCTC*, Vol. 2005 of *Lecture Notes in Artificial Intelligence*, Banff, Canada, October 16–19, 2000. Springer-Verlag, Berlin, Heidelberg, pp. 106–113.
 - [32] M. Olave, V. Rajkovic, and M. Bohanec. An application for admission in public school systems. In: I.T.M. Snellen, W.B.H.J. van de Donk, and J.-P. Baquista (eds), *Expert Systems in Public Administration*. Elsevier Science Publishers (North-Holland), Amsterdam, 1989, pp. 145–160.
 - [33] T.T. Nguyen, C.P. Willis, D.J. Paddon, and H.S. Nguyen. On learning of sunspot classification. In: M.A. Kłopotek, S.T. Wierzchon, and K. Trojanowski (eds), *Intelligent Information Systems, Proceedings of IIPWM'04, May 17–20, 2004, Zakopane, Poland, Advances in Soft Computing*. Springer, Berlin, Heidelberg, 2004, pp. 59–68.
 - [34] T.T. Nguyen, S.H. Nguyen, and H.S. Nguyen. Learning sunspot classification. *Fundam. Inf.* 72(1–3) (2006) 295–309.

37

Outlier and Exception Analysis in Rough Sets and Granular Computing

Tuan Trung Nyuyen

37.1 Introduction

Conceptually, *outliers/exceptions* are kind of atypical samples that stand out from the rest of their group or behave very differently from the norm [1]. While there is still no universally accepted formal definition of being an outlier, several descriptions seem to reflect the essential spirit. According to Hawkin, ‘an outlier is an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism,’ while Barnett and Lewis define an outlier as ‘an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data’ [2]. These samples previously would usually be treated as bias or noisy input data and were frequently discarded or suppressed in subsequent analyses. However, the rapid development of data mining, which aims to extract from data as much knowledge as possible, has made outlier identification and analysis one of its principal branches. Dealing with outliers is crucial to many important fields in real life, such as fraud detection in electronic commerce, intrusion detection, network management, or even space exploration. At the same time, there is an increasing effort in the machine learning community to develop better methods for outlier detection/analysis, as outliers often carry useful subtle hints on the characteristics of the sample domain and, if properly analyzed, may provide valuable guidance in discovering the causalities underlying the behavior of a learning system. As such, they may prove valuable as an additional source of search control knowledge and as a mean for the construction of better classifiers.

Most popular measures to detect outliers [3] are based on either probabilistic density analysis [4] or distance evaluation [5]. Knorr made an attempt to elicit intensional knowledge from outliers through the analysis of the dynamicity of outliers set against changes in attribute subsets [6]. However, no thorough model or scheme for the discovery of intensional knowledge from identified outliers has been established. In particular, there is almost no known attempt to develop methods for outlier analysis among structured objects, i.e., objects that display strong inner dependencies between their own features or components. Perhaps the reason for this is the fact that while many elaborated computation models for the detection of outliers have been proposed, their effective use in eliciting additional domain knowledge, as well as the elicitation of intensional knowledge within outliers, is believed difficult without support of a human expert.

In this chapter, we approach the detection and analysis of outliers in data from a machine learning perspective. We propose a framework based on the granular computing paradigm, using tools and methods originated from rough set and rough mereology theories. The process of outlier detection is refined by the evaluation of classifiers constructed employing intensional knowledge elicited from suspicious samples. The internal structures of the sample domain will be dealt with using hierarchical approximate reasoning schemes and layered learning. We show the role of an external domain knowledge source by human experts in outlier analysis and present methods for the successful assimilation of such knowledge. Introduced methods and schemes are illustrated with an example handwritten digit recognition system.

37.2 Basic Notions

37.2.1 Granular Computing

The term *granular computing* refers to an emerging paradigm that encompasses theories, methods, techniques, and tools for such fields as problem solving, information processing, human perception evaluation, analysis of complex systems, and many others [7]. It is built around the concept of *information granules*, which can be understood as collections of ‘values that are drawn together by indistinguishability, equivalence, similarity, or proximity’ [8]. Granular computing follows the human ability to perceive things in different levels of abstraction (*granularity*) to concentrate on a particular level of interest while preserving the ability to instantly switch to another level in case of need. This allows to obtain different levels of knowledge and, which is important, a better understanding of the inherent structure of this knowledge.

The concept of information granules is closely related to the imprecise nature of human reasoning and perception. Granular computing therefore provides excellent tools and methodologies for problems involving flexible operations on imprecise or approximated concepts expressed in natural language. Better known as *computing with words*, this approach provides a natural ground for our investigations with approximated reasoning schemes.

37.2.2 Rough Sets

Rough set, invented by Pawlak [9] as a theoretical framework to address vague concepts and uncertainty in data, emerged as a powerful tool for dealing with many problems in domains such as information processing, problem solving, classification, approximate reasoning, complex system analysis, and others. Its core concept of an assumed information system provides a simple yet powerful methodology for approximate reasoning about data. Its concept of a indistinguishability relation is closely related to information granularity and at the same time allows for effective methods and techniques to handle operations on granules. In particular, rough mereology theory [10], with its central concept of being part to a degree, provides excellent ground for flexible analysis and tolerant matching of concept and allows for effective realizations of information granularity and concept approximation. Rough mereology makes it possible to introduce the notion of being outlier to a degree, as well as to develop approximate reasoning schemes.

37.2.3 Machine Learning

A machine learning problem can be viewed as a search within a space of hypotheses H for a hypothesis h that best fits a set of training samples T . Among the most popular approaches to such problems are, e.g., discriminant analysis, statistical learning, decision trees, neural networks, or genetic algorithms, commonly referred to as *inductive* learning methods, i.e., methods that generalize from observed training examples by finding features that empirically distinguish positive from negative training examples. Although these methods allow for highly effective learning systems, there often exist proven bounds on the performance of the classifiers they can construct, especially when the samples involved exhibit

complex internal structures, such as optical characters, facial images, or time-series data. It is believed that *analytical* learning methods based on structural analysis of training examples are more suitable in dealing with such samples. In practice, best results are obtained using a combination of the two learning methods [11].

37.2.4 Domain Knowledge in Learning

An analytical learning algorithm, in addition to the training set T and a hypothesis space H , assumes a *domain theory* D which carries prior knowledge about the samples being learned. The search is for a hypothesis h that best fits T and at the same time conforms to D . In other words, a background or domain knowledge is available to the learning system and may help facilitate the search for the target hypothesis. One of the widely used approach to analytical learning is the *explanation-based learning* (EBL) method, which uses specific training examples to analyze, or explain, which features are relevant or irrelevant to the target classification function. The explanations can therefore serve as *search control knowledge* by establishing initial search points or by subsequent altering search directions.

In contrast to the majority of existing outlier detection methods which rely on the information gathered a priori about a set of samples, we investigate an architecture in which certain hints may come from an external, possibly human, expert. Moreover, the explanations will not come as predefined, but rather will be provided by the expert in a two-way dialog along with the evolution of the learning system.

37.3 Ontology Matching and Knowledge Elicitation

The knowledge on training samples that comes from an expert obviously reflects his/her perception about the samples. The language used to describe this knowledge is a component of the expert's ontology, which is an integral part of the expert's perception. In a broad view, an ontology consists of a vocabulary, a set of concepts organized in some kind of structures, and a set of binding relations among those concepts [12]. We assume that the expert's ontology when reasoning about complex structured samples will have the form of a multilayered hierarchy, or a *lattice*, of concepts. A concept on a higher level will be synthesized from its children concepts and their binding relations. The reasoning thus proceeds from the most primitive notions at the lowest levels and works bottom-up toward more complex concepts at higher levels.

37.3.1 External Knowledge Transfer

The knowledge elicitation process assumes that samples, for which the learning system deems it needs additional explanations, are submitted to the expert, which returns not only their correct class identity, but also an explanation on *why* and, perhaps more importantly, *how* he arrived at his decision. This explanation is passed in the form of a rule:

$$[\text{CLASS}(u) = k] \equiv \Im(E\text{Feature}_1(u), \dots, E\text{Feature}_n(u)),$$

where $E\text{Feature}_i$ represents the expert's perception of some characteristics of the sample u , while synthesis operator \Im represents his/her perception of some relations between these characteristics. In a broader view, \Im constitutes of a *relational structure* that encompasses the hierarchy of experts' concepts expressed by $E\text{Feature}_i$.

The ontology matching aims to translate the components of the expert's ontology, such as $E\text{Feature}_i$ and binding relations embedded in the \Im structure, expressed in the foreign language L_f , into the patterns (or classifiers) expressed in a language familiar to the learning system, e.g.,

- $[\text{FaceType}(Ed) = \text{'Square'}] \equiv (\text{Ed.Face().Width} - \text{Ed.Face().Height}) \leq 2\text{cm}$
- $[\text{Eclipse}(p) = \text{'True'}] \equiv (s=p.\text{Sun}()) \wedge (m=p.\text{Moon}()) \wedge (s \cap m.\text{Area} \geq s.\text{Area} \cdot 0.6)$.

As the human perception is inherently prone to variation and deviation, the concepts and relations in a human expert's ontology are approximate by design. To use the terms of granular computing, they are information granules that encapsulate the autonomous yet interdependent aspects of human perception.

The matching process, while seeking to accommodate various degrees of variation and tolerance in approximating those concepts and relations, will follow the same hierarchical structure of the expert's reasoning. This allows parent concepts to be approximated using the approximations of children concepts, essentially building a *layered approximate reasoning scheme*. Its hierarchical structure provides a natural realization of the concept of granularity, where nodes represent clusters of samples/classifiers that are similar within a degree of resemblance/functionality, while layers form different levels of abstraction/perspectives on selected aspects of the sample domain.

On the other hand, with such an established multilayered reasoning architecture, we can take advantages of the results obtained within the granular computing paradigm, which provides frameworks and tools for the fusion and analysis of compound information granules from previously established ones, in a straightforward manner. The intermediate concepts used by external experts to explain their perception are vague and ambiguous, which makes them natural subjects to granular calculi.

The translation must

- allow for a flexible matching of variations of similar domestic patterns to a foreign concept; i.e., the translation result should not be a single pattern, but rather a collection or cluster of patterns;
- find approximations for the foreign concepts and relations while preserving their hierarchical structure. In other words, inherent structure of the provided knowledge should be intact;
- ensure robustness, which means independence from noisy input data and incidental underperformance of approximation on lower levels, and stability, which guarantees that any input pattern matching concepts on a lower level to a satisfactory degree will result in a satisfactory target pattern on the next level.

We assume an architecture that allows a learning system to consult a human expert for advices on how to analyze a particular sample or a set of samples. Typically, this is done in an iterative process, with the system subsequently incorporating knowledge elicited on samples that could not be properly classified in previous attempts [13] (see Figure 37.1).

37.3.2 Approximation of Concepts

A foreign concept C is approximated by a domestic pattern (or a set of patterns) p in terms of a rough inclusion measure $\text{Match}(p, C) \in [0, 1]$. Such measures take root in the theory of rough mereology [10]

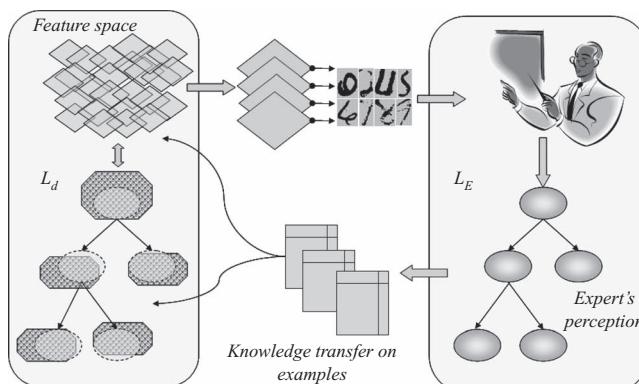


Figure 37.1 Expert's knowledge elicitation

and are designed to deal with the notion of inclusion to a degree. An example of concept inclusion measures would be

$$\text{Match}(p, C) = \frac{|\{u \in T : \text{Found}(p, u) \wedge \text{Fit}(C, u)\}|}{|\{u \in T : \text{Fit}(C, u)\}|},$$

where T is a common set of samples used by both the system and the expert to communicate with each other on the nature of expert's concepts, $\text{Found}(p, u)$ means that a pattern p is present in u , and $\text{Fit}(C, u)$ means u is regarded by the expert as fit to his/her concept C .

Our principal goal is, for each expert's explanation, to find sets of patterns Pat , $\text{Pat}_1, \dots, \text{Pat}_n$ and a relation \mathfrak{I}_d so as to satisfy the following *quality requirement*:

```
if ( $\forall i : \text{Match}(\text{Pat}_i, E\text{Feature}_i) \geq p_i$ )  $\wedge (\text{Pat} = \mathfrak{I}_d(\text{Pat}_1, \dots, \text{Pat}_n))$ ,
then  $\text{Quality}(\text{Pat}) > \alpha,$ 
```

where $p, p_i : i \in \{1, \dots, n\}$ and α are certain cutoff thresholds, while the *Quality* measure, intended to verify if the target pattern Pat fits into the expert's concept of sample class k , can be any, or combination, of popular quality criteria, such as *support*, *coverage*, or *confidence* [14]:

- $\text{Support}_{\text{CLASS}=k}(\text{Pat}) = |\{u \in U : \text{Found}(\text{Pat}, u) \wedge \text{CLASS}(u) = k\}|$,
- $\text{Confidence}_{\text{CLASS}=k}(\text{Pat}) = \frac{\text{Support}(\text{Pat})}{|\{u \in U : \text{Found}(\text{Pat}, u)\}|}$,
- $\text{Coverage}_{\text{CLASS}=k}(\text{Pat}) = \frac{\text{Support}(\text{Pat})}{|\{u \in U : \text{CLASS}(u) = k\}|}$,

where U is the training set.

In other words, we seek to translate the expert's knowledge into the domestic language so as to generalize the expert's reasoning to the largest possible number of training samples. More refined versions of the inclusion measures would involve additional coefficients attached to, e.g., *Found* and *Fit* test function. Adjustment of these coefficients based on feedback from actual data may help optimize the approximation quality.

For example, let us consider a handwritten digit recognition task:

When explaining his/her perception of a particular digit image sample, the expert may employ concepts such as '*circle*', '*vertical strokes*', or '*west open belly*'. The expert will explain what he/she means when he/she says, e.g., '*circle*', by providing a decision table (U, d) with reference samples, where d is the expert decision to which degree the expert considers that '*circle*' appears in samples $u \in U$. The samples in U may be provided by the expert or may be picked up by him/her among samples explicitly submitted by the system, e.g., those that had been misclassified in previous attempts.

The use of rough inclusion measures allows for a very flexible approximation of foreign concept. A stroke at 85° to the horizontal in a sample image can still be regarded as a vertical stroke, though obviously not a 'pure' one. Instead of just answering in a *Yes/No* fashion, the expert may express his/her degrees of belief using such natural language terms as '*strong*', '*fair*', or '*weak*' (see Figure 37.2).

The expert's feedback will come in the form of a decision table (see Table 37.1):

The translation process attempts to find domestic feature(s)/pattern(s) that approximate these degrees of belief (e.g., such as presented in Table 37.2). Domestic patterns satisfying the defined quality requirement can be quickly found, taking into account that sample tables submitted to experts are usually not very large. Since this is essentially a rather simple learning task that involves feature selection, many strategies can be employed. In [15], genetic algorithms equipped with some greedy heuristics are reported successful for a similar problem. Neural networks also prove suitable for effective implementation.

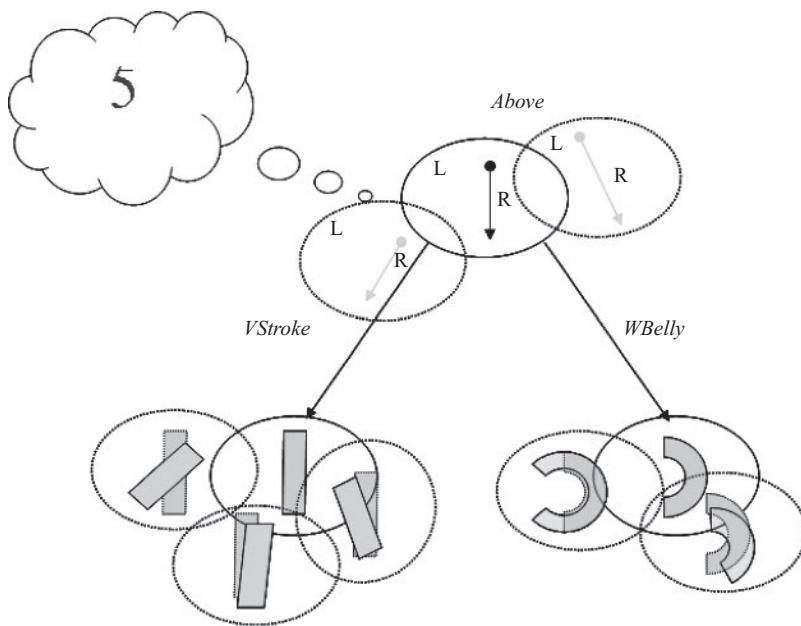


Figure 37.2 Tolerant matching by expert

It can be observed that the intermediate concepts like ‘circle’ or ‘vertical strokes,’ provided by a human expert, along with satisfiability assessments like ‘strong,’ ‘fair,’ or ‘weak’ form information granules within the perception of the expert. The granules correspond to different levels of abstraction, or focus, of his/her reasoning about a particular class of samples. The translation process transforms these information granules into classifiers capable of matching particular parts of actual samples with intermediate expert’s concepts, which essentially incorporates the human perception, by way of using information granules, into the learning process.

Table 37.1 Perceived features

	<i>Circle</i>
u_1	Strong
u_2	Weak
...	...
u_n	Fair

Table 37.2 Translated features

	<i>DPat</i>	<i>Circle</i>
u_1	252	Strong
u_2	4	Weak
...
u_n	90	Fair

37.3.3 Approximation of Relations

37.3.3.1 Perception Structures

The approximation of higher level relations between concepts has been formalized within the framework of perception structures, recently developed by Skowron [16]. A *perception structure* S , in a simpler form, is defined as

$$S = (U, M, F, \models, p),$$

where U is a set of samples, F is a family of formulas expressed in domestic language that describe certain features of the samples, and M is a family of relational structures in which these formulas can be evaluated, while $p: U \rightarrow M \times F$ is a *perception function*, such that $\forall u \in U : p_1(u) \models p_2(u)$ (p_1 and p_2 are the first and second component projections of p), which means that $p_2(u)$ is satisfied (is true) in the relational structure $p_1(u)$. This may express that some relations among features within samples are observed.

For a given sample u , we define a set

$$M(u) = \{R \in M : R \models p_2(u)\},$$

which contains all possible relational structures for which formulas or, in other words, features observed in u yield.

37.3.3.2 Approximate Clusters

Given a perception structure S , an approximate cluster of a given sample u is defined as

$$[u]_S = \bigcup_{R \in M(u)} p_1^{-1}(R).$$

This cluster contains samples from U that have similar structures to u , with regard to the perception p , i.e., those with similar relational structures that also hold true the features observed in u (see Figure 37.3).

For example, if we construct a perception structure that contains a formula describing a part of a digit is ‘above’ another part, then within this perception, the approximate cluster of a digit ‘6,’ which has a slant stroke over a circle, would comprise all digits that have similar structure, i.e., containing a slant stroke over a circle.

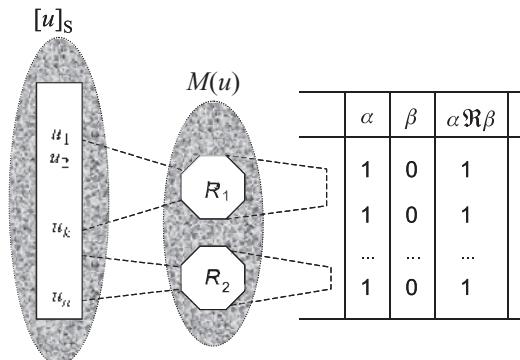


Figure 37.3 Approximate cluster

Perception structures, following natural constructs in the expert's foreign language, should involve tolerant matching. Let us suppose that we allow a 'soft' perception on samples of U by introducing a similarity relation τ between them. This relation, for example, might assume that two samples resemble each other to a degree. This naturally leads to clusters of similar relational structures in M . With samples now perceived as similar to each other in a degree, we shall allow for a similarity relation in M . Two relational structures might be considered approximately *the same* if they allow for similar formulas to yield similar results in majority of cases when these formulas are applicable. The family M thus becomes granulated by τ and is denoted by M_τ .

The same follows for the family F of features, or formulas, that for instance, do not always have the same value, but are equivalent in most cases, or in all or majority, of a cluster of similar relational structures. Formulas' evaluation might be extended to comprise degrees of truth values, rather than plain binary constants. The family F hence becomes granulated with regard to τ and is denoted by F_τ .

The perception structure S hence becomes, for a given similarity measure, τ in U : $S = (U, M_\tau, F_\tau, \models, p)$, which permits a much more flexible space and a variety of methods for concept approximation.

In the above-mentioned example, a similarity-induced perception might consider as the approximate cluster of a digit '5' the set of every sample that will have a *stroke* over a *closed curve* (not just slant strokes and circles as before). Moreover, the new perception also allows for a greater variation of configurations considered to fit into the concept of '*above*'.

The definition of an approximate cluster becomes

$$[u]_S = \bigcup_{R \in M_\tau(u)} p_1^{-1}(R).$$

The task of approximating an expert's concept involving relations between components is now equivalent to finding a perception function that satisfies some quality criteria. Let us suppose that the expert provides us a set C of samples he/she considers fit to his/her concept. We have to find a perception function p , such that

$$\text{Confidence: } \frac{|[u]_S \cap C|}{|[u]_S|} > c$$

and/or

$$\text{Support: } \frac{|[u]_S \cap C|}{|U|} > s,$$

where u is some sample from C , and $0 < c, s < 1$.

Having approximated the expert's features $EFeature_i$, we can try to translate his/her relation \mathfrak{I} into our \mathfrak{I}_d by asking the expert to go through U and provide us with the additional attributes of how strongly the expert considers the presence of $EFeature_i$ and to what degree he/she believes that the relation \mathfrak{I} holds. Again, let us consider the handwritten recognition case (see Table 37.3).

We then replace the attributes corresponding to $EFeature_i$ with the rough inclusion measures of the domestic feature sets that approximate those concepts (computed in the previous step). In the next stage, we try to add other features, possibly induced from original domestic primitives, in order to approximate

Table 37.3 Perceived relations

	VStroke	WBelly	Above
u_1	Strong	Strong	Strong
u_2	Fair	Weak	Weak
...
u_n	Fair	Fair	Weak

Table 37.4 Translated relations

	#V_S	#NES	$S_y < B_y$	Above
u_1	0.8	0.9	(Strong, 1.0)	(Strong, 0.9)
u_2	0.9	1.0	(Weak, 0.1)	(Weak, 0.1)
...
u_n	0.9	0.6	(Fair, 0.3)	(Weak, 0.2)

the decision d . Such a feature may be expressed by $S_y < B_y$, which tells whether the median center of the stroke is placed closer to the upper edge of the image than the median center of the belly (see Table 37.4)

The expert's perception ‘A “6” is something that has a “vertical stroke” “above” a “belly open to the west”’ is eventually approximated by a classifier in the form of a rule:

if S(#BL_SL > 23) **AND** B(#NESW > 12%) **AND** $S_y < B_y$ **then** CL = ‘6’,

where S and B are designations of pixel collections, #BL_SL and #NESW are numbers of pixels with particular topological feature codes, and $S_y < B_y$ reasons about centers of gravity of the two collections.

Approximate reasoning schemes embody the concept of information granularity by introducing a hierarchical structure of abstraction levels for the external knowledge that comes in the form of a human expert's perception. The granularity helps reduce the cost of the knowledge transfer process, taking advantage of the expert's hints. At the same time, the hierarchical structure ensures to preserve approximation quality criteria that would be hard to obtain in a flat, single-level learning process.

From yet another perspective, the reasoning schemes that encompass a human expert's intermediate concepts like ‘vertical stroke,’ ‘above,’ and their satisfiability assessments such as ‘strong’ or ‘fair’ represent the way humans reason about samples through different levels of abstraction. The connections between intermediate concepts and transitions from lower to upper levels allow to shift the perception focus from smaller parts of objects to more abstract, global features. These reasoning schemes also provide off-the-shelf recipes as to how to assemble more compound information granules from simpler, already-established ones. Translated into domestic languages, they become powerful classifiers that help expand the human perception structures to actual samples.

37.4 Outlier Identification

As mentioned in Section 37.1, most existing outlier identification methods employ either probabilistic density analysis or distance measures evaluation. Probabilistic approach typically runs a series of statistical discordancy tests on a sample to determine whether it can be qualified as an outlier. Sometimes this procedure is enhanced by a dynamic learning process. Their main weakness is the assumption of an underlying distribution of samples, which is not always available in many real-life applications. Difficulties with their scalability in numbers of samples and dimensions are also a setback of primary concern.

Another approach to outlier detection relies on certain distance measures established between samples. Known methods are data clustering and neighbor analysis. While this approach can be applied to data without any assumed a priori distribution, they usually entail significant computation costs.

Let C_k be a cluster of samples for class k during the training phase and d_k be the distance function established for that class. For a given cutoff coefficient $\alpha \in (0, 1]$, a sample u^* of class k is considered ‘difficult,’ ‘hard,’ or ‘outlier’ if, e.g.,

$$d_k(u^*, C_k) \geq \alpha \cdot \max\{(v, C_K) : v \in TR \wedge CLASS(v) = k\},$$

which means u^* is far from the ‘norm’ in term of its distance to the cluster center, or

$$|\{v : v \in C_k \wedge d_k(u^*, v) \leq d_k(v, C_k)\}| \leq \alpha \cdot |C_k|,$$

which means u^* is among the most outreaching samples of the cluster.

Another popular definition of outlier is

$$|\{v : v \in C_k \wedge d_k(u^*, v) \geq D\}| \leq \alpha \cdot |C_k|,$$

which means at least a fraction α of objects in C_k lies in a greater distance than D from u^* .

It can be observed that both approaches pay little attention to the problem of eliciting intensional knowledge from outliers meaning no elaborated information that may help explain the reasons why a sample is considered outlier. This kind of knowledge is important for the validity evaluation of identified outliers and is certainly useful in improving the overall understanding of the data.

Knorr and Ng made an attempt to address this issue by introducing the notion strength of outliers, derived from an analysis of dynamicity of outlier sets against changes in the features’ subsets [6, 17]. Such analyses belong to the very well-established application domain of rough sets, and indeed a formalization of a similar approach within the framework of rough sets has been proposed by Jiang *et al.* [18].

Our approach to outlier detection and analysis will assume a somewhat different perspective. It focuses on two main issues:

1. elicitation of intensional knowledge from outliers by approximating the perception of external human experts;
2. evaluation of suspicious samples by verification of the performance of classifiers constructed using knowledge elicited from these samples.

Having established a mechanism for eliciting expert’s knowledge as described in previous sections, we can develop outlier detection tests that might be completely independent of the existing similarity measures within the learning system as outlined in Figure 37.4.

For a given training sample u^* ,

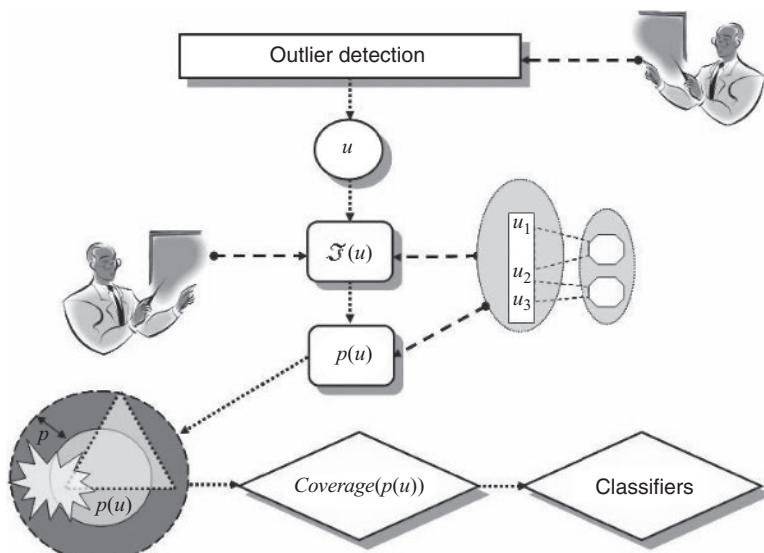


Figure 37.4 Outlier analysis scheme

Step 1. We ask the expert for his/her explanation on u^* .

Step 2. The expert provides a foreign knowledge structure $\mathfrak{I}(u^*)$.

Step 3. We approximate $\mathfrak{I}(u^*)$ under restrictive matching degrees to ensure that only the immediate neighborhood of u^* is investigated. Let us say that the result of such an approximation is a pattern (or set of patterns) p_u^* .

Step 4. It is now sufficient to check $Coverage(p_u^*)$. If this coverage is high, it signifies that u^* may bear significant information that is also found in many other samples. The sample u^* cannot therefore be regarded as an outlier despite the fact that there may not be many other samples in its vicinity in terms of existing domestic distance measures of the learning system.

This test shows that distance-based outlier analysis and expert's elicited knowledge are complementary to each other.

In our architecture, outliers may be detected as samples that defied previous classification efforts or samples that pass the above-described outlier test, but may also be selected by the expert him/herself. In this way, we can benefit from the best of both sources of knowledge.

37.5 Experiments

In order to illustrate the developed methods, we conducted a series of experiments on the NIST Handwritten Segmented Character Special Database 3. We compared the performances gained by a standard learning approach with and without the aid of the domain knowledge. The additional knowledge, passed by a human expert on popular classes, as well as some atypical samples allowed to reduce the time needed by the learning phase from 205 min to 168 min, which means an improvement of about 22% without loss in classification quality. In case of screening classifiers, i.e., those that decide a sample *does not* belong to given classes, the improvement is around 60%. The representational samples found are also slightly simpler than those computed without using the background knowledge (see Table 37.5).

37.6 Conclusion

Granular computing is a natural, elegant, yet powerful, paradigm that helps to better understand the mechanisms of human reasoning in complex situations. Its techniques and methods enable machine learning systems to incorporate the human way of thinking into their knowledge-acquiring and problem-solving processes, which allow such systems to overcome obstacles and pitfalls, most usually related to the vagueness and imprecision of descriptions of complex phenomena and processes involved therein, which might be impenetrable otherwise.

We presented in details an approach to the problem of outlier detection and analysis in data from a machine learning perspective. We focus on the elicitation of intensional knowledge from outliers using additional background information provided by an external human expert. We described an interactive scheme for the knowledge transfer between the expert and the learning system, using methodologies and tools originated from granular computing paradigm, rough set and rough mereology theories, as well

Table 37.5 Comparison of performances

	No domain knowledge	With domain knowledge	Gain
Total learning time	205 s	168 s	22%
Negative classifier learning time	3.7 s	2.2 s	40%
Positive classifier learning time	28.2 s	19.4 s	31%
Skeleton graph size	3–5 nodes	2–5 nodes	

as techniques pertaining to the approximate reasoning schemes. Proposed approach proves capable to yield effective implementations and offers a complementary perspective compared with other existing approaches in the field of outlier detection and analysis.

The challenge to this approach in particular, and to granular computing in general, is to develop efficient algorithms and computing techniques for better implementations of proposed methodologies for the description, analysis, decomposition, and fusion of compound information granules, especially those pertaining to various intermediate components of the human perception and reasoning.

Acknowledgment

The author expresses his deepest gratitude to Professor Andrzej Skowron for valuable comments and support during the work on this chapter. This work has been supported by the grant from Ministry of Science and Higher Education of the Republic of Poland and by the grant Innovative Economy Operational Programme 2007–2013 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland.

References

- [1] C.C. Aggarwal and P.S. Yu. Outlier detection for high dimensional data. In: *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, ACM Press, New York, NY, 2001. pp. 37–46.
- [2] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley and Sons Ltd, New York, July 1978.
- [3] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artif. Intell. Rev.* 22(2) (2004) 85–126.
- [4] M.M. Breunig, H.-P. Kriegel, R.T. Ng, and J. Sander. Lof: identifying density-based local outliers. *SIGMOD Rec.* 29(2) (2000) 93–104.
- [5] E.M. Knorr, R.T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. *VLDB J.* 8(3) (2000) 237–253.
- [6] E.M. Knorr and R.T. Ng. Finding intensional knowledge of distance-based outliers. In: *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1999, pp. 211–222.
- [7] W. Pedrycz (ed.). *Granular Computing: An Emerging Paradigm*. Physica-Verlag GmbH, Heidelberg, Germany, 2001.
- [8] L.A. Zadeh. From imprecise to granular probabilities. *Fuzzy Sets Syst.* 154(3) (2005) 370–374.
- [9] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Norwell, MA, 1992.
- [10] L. Polkowski and A. Skowron. Rough mereology: A new paradigm for approximate reasoning. *J. Approx. Reason.* 15(4) (1996) 333–365.
- [11] T.M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.
- [12] D. Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag New York, Inc., Secaucus, NJ, 2003.
- [13] T.T. Nguyen. Eliciting domain knowledge in handwritten digit recognition. In: S.K. Pal, S. Bandyopadhyay and S. Biswas (eds), *First International Conference on Pattern Recognition and Machine Intelligence*, Vol. 3776 of *Lecture Notes in Computer Science*, Springer-Verlag GmbH, Kolkata, India, 2005, pp. 762–767.
- [14] L. Polkowski and A. Skowron. Constructing rough mereological granules of classifying rules and classifying algorithms. In: B. Bouchon-Meunier, J. Gutierrez-Rios, L. Magdalena, R.R. Yager, and J. Kacprzyk (eds), *Technologies for Constructing Intelligent Systems I*, Physica-Verlag, GmbH, Heidelberg, Germany, pp. 57–70.
- [15] L.S. Oliveira, R. Sabourin, F. Bortolozzi, and C.Y. Suen. Feature selection using multi-objective genetic algorithms for handwritten digit recognition. In: *Proceedings of the 16th International Conference on Pattern Recognition (ICPR02)*, IEEE Computer Society, Quebec City, Canada, 2002, pp. I: 568–571.
- [16] A. Skowron. Rough sets in perception-based computing. In: S.K. Pal, S. Bandyopadhyay, and S. Biswas (eds), *Proceedings of the First International Conference on Pattern Recognition and Machine Intelligence (PReMI'05)*, Springer-Verlag, Kolkata, India, 2005.
- [17] E.M. Knorr. *Outliers and Data Mining: Finding Exceptions in Data*. Ph.D. Thesis. University of British Columbia, April 2002.
- [18] F. Jiang, Y. Sui, and C. Cao. Outlier detection using rough set theory. In: D. Slezak, M. Duentsch, I. Wang, G. Szczuka, and Y. Yao (eds), *10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 2005*, Regina, Canada, 2005, pp. 79–87.

38

Information Access and Retrieval

Gloria Bordogna, Donald H. Kraft, and Gabriella Pasi

38.1 Introduction

The amount of information available on the Internet has increased to a point that there is a great demand for effective systems that allow easy and flexible access to information [1–3]. By flexibility we mean the capability of the system to manage imperfect, i.e., vague and/or uncertain, information, as well as to adapt its behavior to user's context.

The usual approach to access relevant information to specific user needs is to use Information Retrieval (IR) systems. In textual information retrieval (i.e., the automatic process of retrieving texts relevant to specific user needs), the terms extracted from or associated with the texts (called the index terms) constitute the information granules on which IR systems work. Recently, there has been an increasing interest in the topic called the semantic web, which requires the definition of a flexible infrastructure in order to organize semantically the available information so as to allow a better communication between humans and computers.

Search engines are but the tip of the iceberg of IR on the Internet [4]: most search engines are based on retrieval models defined several years ago. They are generally based on a combination of the vector space model and of the Boolean retrieval model [5]. Surprisingly, the query language which most, if not all, of these systems employ is based on Boolean logic, defined as the first formal query language for IR systems. Boolean logic traditionally does not allow the expression of soft requirements to specify the information content of desired documents. Thus, it is intolerant in terms of incorporating imprecision or of modeling user context. In fact, two distinct users formulating the same query would obtain the same retrieval results in spite of the fact that their choices could be defined by different criteria and for different aims.

To overcome the limitations of current IR systems and search engines, in recent years a great deal of research has been undertaken to the main aim of modeling the subjectivity, vagueness, and imprecision that is intrinsic to the process of locating relevant information. To this purpose, soft computing techniques have been widely applied as a means to obtain a greater flexibility in designing systems for information access [6–8]. In particular, fuzzy set theory has been applied to IR, starting in the 1970s. This application of fuzzy set theory has allowed the definition of retrieval techniques capable of modeling, at least to some extent, human subjectivity in terms of estimating the (partial) relevance of documents to user needs. For example, users can 'precisiate' the semantics of *significant* or *very significant* of the index terms, depending on their position within a document, considering all documents subparts, or just some of them, thus allowing to build a fuzzy representation of semistructured documents that accounts for a given level of granularity of the document content. Also, at the level of query expressions, users can employ

linguistic terms to qualify the desired importance of the search keywords, thus incorporating also at this level the possibility to granulate the relevance scores.

The objective of this chapter is to provide an overview of the role of fuzzy set theory in designing flexible IR systems. The chapter is organized as follows: in the next section, the challenge of modeling flexibility in IR systems is analyzed. An overview of the main approaches for applying fuzzy set theory to model flexible IR systems is presented. In Section 38.3 the problem of document indexing is introduced and then some recent and promising approaches to fuzzy indexing are described. Section 38.4 is devoted to the description of flexible query languages for IR systems, based on the specification of soft constraints expressed by linguistic selection conditions which capture the vagueness of the user needs and simplify the query formulation. In Section 38.5 document-clustering techniques are introduced. Finally in Section 38.6 some fuzzy set approaches to distributed IR are described and the conclusions summarize the main contents of this chapter.

38.2 Flexibility in Information Retrieval

IR is the activity that aims at providing a fast and effective content-based access to a large amount of stored information, usually organized in documents (information items) [5, 9–12]. A user accesses the IR system by explicitly formulating a query through a set of constraints that the relevant information items must satisfy. The aim of an IR system is to evaluate the user query and retrieve all documents estimated to be relevant to that query. This is achieved by comparing the formal representation of the documents with the formal user query. This is a decision-making problem: how to identify the information items that correspond to the users' information preferences (i.e., documents relevant to their information needs)? What a user expects from an IR system is a list of the relevant documents ordered according to her/his preferences. The IR system then acts as an intermediary in this decision process: it 'simulates' the decision process that the user would personally undertake. The documents constitute the alternatives on which the decision process has to be performed, with the aim of identifying the relevant documents [13, 14]. The input of these systems is constituted by a user query; their output is usually an ordered list of selected items, which have been estimated relevant to the information needs expressed in the user query.

A representation of the main functional modules of an IR system is sketched in Figure 38.1.

A central, but ill-defined, concept in IR is the concept of relevance: only the user can determine the true relevance of a document, i.e., the usefulness, pertinence, appropriateness, or utility of that document with respect to a given query. Thus, relevance is time, situation, and user specific. Moreover, studies show [15] that users are influenced by many factors that go beyond whether or not a given document is 'about' the topics covered in the user queries. For example, it has been shown that the order of a document as presented in a ranked list can affect the user's opinion. Moreover, the number of documents needed sometimes dictates relevance; for example, one trusted WWW page that tells a user what the

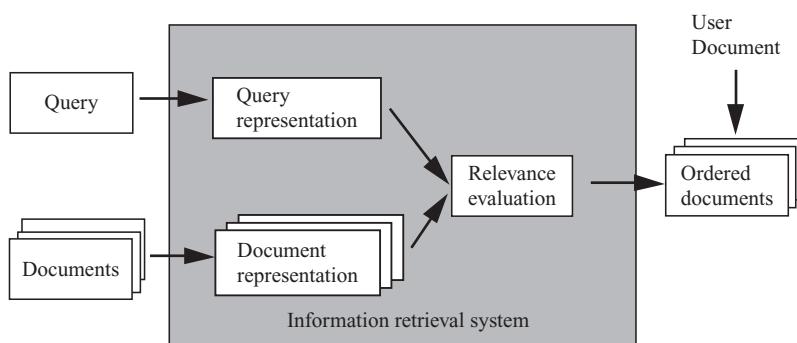


Figure 38.1 Scheme of a system for the storage and retrieval of information

weather is going to be tomorrow in Milan suffices, so that additional pages repeating that information become non-relevant. This is why studies using the judgments of subject experts who can determine, and imprecisely at that, what topics are covered by which documents are seen as limited. What is worse is that users can accidentally find a good document that is seemingly not on topic and which the user finds useful. However, it is also possible that documents which the IR system construes as useful, on topic, and good for the user are seen as non-relevant. Thus, the retrieval system is not able to retrieve all the relevant documents and nothing but the relevant documents. Since relevance depends on many complex factors related to the user, an effective automatic estimation of the relevance of documents to a query should attempt, in some fashion, to take into account a user model. This model can be simplified by providing vague descriptions of user needs.

In order to estimate a document's relevance with respect to a specific user query, the IR system is based on a formal model that provides a consistent representation of both documents and user queries. Most of the existing IR systems and search engines offer a very simple model of IR – a model that can emphasize efficiency at the expense of effectiveness.

Nowadays, one of the main directions for improving effectiveness is to model the user context and to take into account user subjectivity. This is done in both documents indexing and querying. An important aspect affecting the effectiveness of IR systems is related to the way document content is represented. Document representations are usually extremely simple, based on keywords extraction and weighting. Moreover, the IR systems generally produce a unique representation of documents to all users, not taking into account the idea that each user will look at document content in a subjective way, possibly emphasizing some document subparts over other subparts. This adaptive view of the document has not often been modeled. Another important aspect is related to the fact that on the World Wide Web (WWW), some standardization for the representation of semistructured information (e.g., XML) is becoming more often employed. For this reason it is important to exploit this structure in order to represent the information these documents carry.

Another crucial aspect that affects IR system effectiveness is related to the characteristics of the query language, which should accurately represent a user's information needs. Current query languages are usually based on keywords and thus do not allow a user to express tolerant selection conditions. In real situations, however, a user could be satisfied with the retrieval of at least something close to his/her request as opposed to be given nothing at all.

In recent years, a goodly amount of research that dealt with means to improve IR systems was mostly devoted to the modeling of the concept of partiality intrinsic in the IR process and to making these systems adaptive, i.e., capable of 'learning' a user's concept of relevance [1, 7, 8, 16]. These approaches often apply some of what has been called soft computing techniques, among which is fuzzy set theory.

To be specific, fuzzy set theory has been extensively applied to extend IR in order to model better some of the aspects of the vagueness and subjectivity that characterize the retrieval process [2, 17–24]. In particular, one sees the objectives:

- to define new IR models;
- to deal with the imprecision and subjectivity that characterize the document indexing process;
- to manage the user's vagueness in query formulation;
- to soften the associative mechanisms, such as thesauri and document-clustering algorithms, which are often employed to extend the functionalities of the basic IR scheme;
- to define meta search engines and to define flexible approaches to distributed IR;
- to represent and inquiry semistructured information (XML).

Surveys of fuzzy IR models and of fuzzy generalizations of the Boolean IR model can be found in [1, 2, 25]. More recently, some interesting approaches have been defined to probabilistic-based information retrieval [26–28].

At the level of document indexing, some fuzzy techniques have been defined for providing more specific and personalized representations of documents than those generated by existing indexing procedures. The main idea is to explicitly model an indexing strategy that adapts a formal document representation to a user's personalized view of the document's information content [18, 29–31].

Fuzzy associative mechanisms based on thesauri or document-clustering techniques [32–34] have been constructed in order to cope with the incompleteness and ambiguity characterizing either the representation of documents or user queries. Fuzzy clustering techniques can be applied to identify fuzzy partitions of the documents in a collection so as to support an automatic document categorization [35, 36]. For each fuzzy cluster, a label is identified, summarizing that cluster's content. This allows one to support various associative retrieval techniques, e.g., the expansion of documents retrieved by a query with associated documents in the same cluster, or the direct matching of a query with the clusters' representations and then retrieving of documents in each relevant cluster.

38.3 Document Indexing

The most used automatic indexing procedures are based on term extraction and weighting; documents are represented by a collection of index terms with associated weights (the index term weights). These weights can be generated either subjectively or by using frequency analysis [11, 12]. Frequency analysis for the computation of the index term weights is based on the quantification of heuristic considerations such as the notion that if a given term has a high occurrence within a given document and a low occurrence within the entire collection or archive, it is more significant. A typical definition of the index term weight is $F(d, t) = f(d, t)^*IDF(t)$, in which $f(d, t)$ is the frequency of the term t in document d and $IDF(t)$ is a normalized inverse document frequency of t within the whole collection.

Often, there are also rules for the assignment of terms, along with term relationships. Term relationships include synonyms, narrower terms, broader terms, and related terms.

Another complex task, one that generates ambiguities and uncertainties, applies stemming algorithms in order to strip off suffixes [11]. This enhances the frequencies of the specific stems. This is also true of those terms representing distinct concepts, such as in the case of the root ‘ambi’ that can represent ‘ambiguous’ and ‘ambidextrous.’ A related problem is the computation of the significance degree of the ‘roots’ in the documents, i.e., the ‘root’ weights. This process is generally based on summing up all the frequencies of the index terms having that ‘root’ in the document. However, a matching mechanism evaluating such a representation based on ‘roots’ is faced with further uncertainty, for the information on the original index terms of documents and their index term weights is lost.

Based on keywords indexing, a document can be seen as a fuzzy set of terms [25, 37]. Formally, we have $R_d = \sum_{t \in T} \mu_d(t)/t$, in which the membership function is $\mu_d: D \times T \rightarrow [0, 1]$ with $\mu_d(t) = F(d, t)$, the index term weight. The query-evaluation mechanism is regarded as a fuzzy decision process that evaluates the degree of satisfaction of the query constraints by each document representation by applying a partial matching function. This degree, named Retrieval Status Value (RSV), is interpreted as the degree of relevance of the document to the query and is used to rank the documents. Then, as a result of a query evaluation, a fuzzy set of documents is retrieved in which the RSV is the membership value. The definition of the partial matching function is strictly dependent on the query language definition. In the case of a Boolean query, the degree of satisfaction (RSV) of a query term t by a document d is the index term weight $F(d, t)$. The increased complexity of evaluating a Boolean query in the fuzzy context with respect to a crisp Boolean context is due to the computation of the AND and OR, usually interpreted as min and max operators respectively. Nevertheless, we achieve the benefit to model the vagueness of the relevance concept, which is considered gradual within this framework.

38.3.1 Personalized Fuzzy Indexing of Semistructured Documents

The diffusion of semistructured textual documents has encouraged the definition of more sophisticated representations of document content, taking into account information conveyed by the ‘structure’ of the documents.

The usual weighted representation of documents has the limitation of not taking into account the idea that a term can play a different role within a given document according to the distribution of its occurrences.

For example, consider an XML document organized in ‘logical’ sections, such as with scientific papers that are usually organized into such sections as *title*, *authors*, *abstract*, *introduction*, *references*, and so on. An occurrence of a term in the *title* has a distinct informative role that can be different than if this term’s occurrence is in the *references* section. Moreover, indexing procedures presented so far produce the same document representation to all users; this enhances the system’s efficiency but implies the possibility of a severe loss of effectiveness. In fact, when examining a structured document, users have their personal views of the document’s information content. Users should naturally privilege the search in some subparts of the document structure, depending on their preferences. This has supported the idea of dynamic and personalized indexing, as proposed in [18, 29], intended as an indexing procedure that takes into account the user indications explicitly specified by constraints on the document structure (preference elicitation on the structure of a document). This preference specification can be exploited by the matching mechanism to privilege the search within the most preferred sections of the document. The user/system interaction can then generate a personalized document representation, reflecting a given level of granulation of the document content, which is distinct for distinct users.

One such model of personalized indexing of structured documents [18, 29] is constituted by a static component and by an adaptive query-evaluation component; the static component provides an a priori computation of an index term weight for each logical section of the document. The formal representation of a document becomes a fuzzy binary relation defined on the Cartesian product $T \times S$ (where T is the set of index terms and S is the set of identifiers of the documents’ sections). With each pair $\langle \text{section}, \text{term} \rangle$, a significance degree in $[0, 1]$ is computed, expressing the significance of the term in the document section.

The adaptive component is activated by a user query and provides an aggregation strategy of the n index term weights (where n is the number of sections) into an overall index term weight. At the first level, the user may express preferences on the document sections, outlining those that the system should more heavily take into account in evaluating the relevance of a document to a user query. This user preference on the document structure is exploited to enhance the computation of index term weights: the importance of index terms is strictly related to the importance to the user of the logical sections in which they appear. At the second level, the user can decide the kind of aggregation to apply for producing the overall significance degree (see Figure 38.2). This is done by the specification of a linguistic quantifier such as *at least k* and *most* [38]. In the fuzzy indexing model defined in [18, 29] linguistic quantifiers are formally defined as Ordered Weighted Averaging (OWA) operators [39].

By adopting the document representation as shown in Figure 38.2 the same query can select documents in different relevance orders, depending on the user preferences on the documents sections.

In [30, 31] this personalized fuzzy representation is customized for documents written in HyperText Markup Language (HTML). In this context, tags are seen as syntactic elements, carrying an indication of the importance of the associated text. The underlying assumption here is that the writer associates an implicit, distinct importance with the documents’ various subparts by delimiting them by means of appropriate tags. On the basis of these considerations, an indexing function has been proposed, which computes the significance of a term in a document by taking into account the distinct

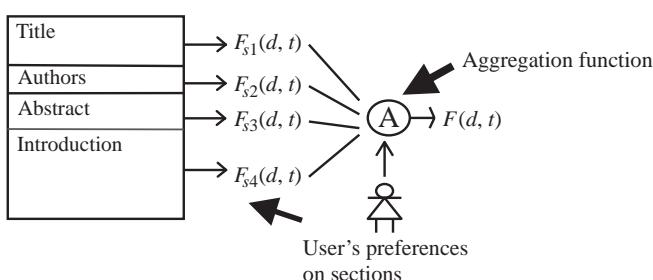


Figure 38.2 Sketch of the personalized indexing procedure

role of term occurrences according to the importance of the sections (as indicated by the tags) in which they appear.

38.3.2 Fuzzy Concept-Based Indexing

Of late, an increasing number of IR models have been proposed that are based on concepts rather than on keywords. This can be seen as modeling document representations at a higher level of granularity, trying to describe the topical content and structure of documents [40]. These efforts have given rise to what is now called concept-based (or conceptual) IR, which aims at retrieving relevant documents on the basis of their meaning rather than on their keywords. The main idea at the foundation of conceptual IR is that the meaning of a text depends on conceptual relationships to objects in the world rather than to linguistic relations found in text or dictionaries [41]. This means that one has to have sets of words, phrases, and names be related to the concepts that they encode.

A fuzzy concept-based IR model that relies on the existence of a conceptual hierarchical structure to encode the contents of the collection's application domain has been proposed. Here, both documents and queries are represented as weighted trees. The evaluation of a conjunctive query is interpreted as computing a degree of inclusion between two subtrees. The basic principle of the proposal is that if a document explicitly includes some terms, the document can also be related, at least to some extent, to more general concepts. This latter point is handled at the technical level by a completion procedure: this consists in assessing positive weights to terms that do not appear directly in the documents [42]. The possibility of the completion of queries has also been raised [27]. Another approach to concept indexing of documents is the augmentation of the index terms of a document with those appearing in a thesaurus or ontology. The definition of a thesaurus for a set of terms, or of pseudothesauri or even ontologies, is a complex task, depending on the expert indexers' competence in the field(s) covered by the collection and on the purpose of the indexing and retrieval activities. It is often desirable to distinguish the strength of the association link between pairs of terms, but often this strength is difficult to quantify by a number, since it depends on criteria, which are not defined well enough nor are clear enough to the experts. Some researchers have attempted to generate fuzzy thesauri, or fuzzy pseudothesauri, automatically by a statistical analysis based, for example, on the cooccurrences of index terms in documents or documents' subparts [32–34, 43].

When one wants to augment the representation of a document with the fuzzy set of related terms, one is unavoidably faced with uncertainty. In fact, if the index term t of document d is highly related with term t' in the thesaurus, one might still not feel that term t' is appropriate to represent the content of document d . Further, there is the problem of computing the index term weight of term t' in d by taking into account both the weight of the original term t in d and the degree of strength of the relation between t and t' . Moreover, rough sets have been employed [44] to mine a controlled vocabulary.

38.4 Query Languages

When expressing needs for information, users often formulate queries in a natural language (NL), at least at first, in order to describe those needs [45]. If a retrieval system is expected to respond properly, it will need to parse those NL statements and interpret their meaning, operations that inevitably force the system designers to face up to the ambiguity, vagueness, and imprecision of NL. In an attempt to ease the task of an NL interpreter, artificial query languages for expressing content-based requests have been defined for IR systems in which the reduction of complexity of the NL has been done at the expense of its expressiveness. Thus, a query language based on Boolean logic and the query language underlying the vector space model, the latter of which is based on a list of terms, have been proposed and are still employed by many commercial systems, despite their inability to represent well the vagueness of user needs.

38.4.1 Flexible Query Languages

Fuzzy set theory has been applied at various levels in order to allow the expression of vague queries, to improve the expressiveness of Boolean queries, and, at the same time, to simplify the Boolean formulations of information needs. In this context a flexible query may consist of either one or both of the following soft components:

- the association of weights with query terms, where weights are interpreted as flexible (i.e., elastic) constraints on the index term weights in each document representation; a weight can be either a numeric value w in $[0, 1]$ (which is associated with a soft constraint such as *close to w* or *above w*) or a linguistic value or qualifier, such as *very important* or *fairly important*, for the linguistic variable *importance*; this allows to express soft constraints with distinct granularity;
- the aggregation of query terms by means of linguistic aggregation operators as expressed by means of linguistic quantifiers used as aggregation operators.

The notion of a linguistic variable [46] is suitable to represent and manage linguistic concepts, and for this reason it has been used to formalize the semantics of linguistic terms introduced in the generalized Boolean query language. When flexible constraints are specified, the query-evaluation mechanism is regarded as performing a fuzzy decision process that evaluates the degree of satisfaction of the query constraints by each document representation by applying a partial matching function.

Flexible constraints are defined as fuzzy subsets of the set $[0, 1]$ of the index term weights; the membership value $\mu_{\text{weight}}(F(d, t))$ is the degree of satisfaction of the flexible constraint imposed by the *weight* associated with query term t by the index term weight of t in document d . The result of the evaluation is a fuzzy set: $\sum_{d \in D} \mu_{\text{weight}}(F(d, t))/d$. Linguistic extensions of the Boolean query language have been defined, based on the concept of a linguistic variable, so that a user can add linguistic qualifiers, such as associate query terms like *important*, *very important*, and *fairly important* to the query terms. This would serve to qualify the desired importance of the search terms in the query. A pair $\langle t, \text{important} \rangle$ expresses a flexible constraint evaluated by the function $\mu_{\text{important}}$ on the term significance values (the $F(d, t)$ values) for term t . The evaluation of the relevance of a given document d to a query consisting of the pair $\langle t, \text{important} \rangle$ is then computed by applying the function $\mu_{\text{important}}$ to the value $F(d, t)$ [47, 48].

A second problem concerns the specification of soft aggregation operators besides the AND operator (conjunction), the NOT operator (negation), and the OR operator (disjunction). Within fuzzy set theory, a generalization of the Boolean query language has been defined, based on the concept of linguistic quantifiers: these quantifiers are employed to specify both crisp and vague aggregation criteria of the selection constraints [49]. New aggregation operators can be specified with a self-expressive meaning, operators such as *at least k* and *most of*. They are defined with a mean-like behavior that allows partial compensation lying between *all* and *at least one* of the selection conditions. The linguistic quantifiers are defined through OWA operators [39]. An alternative approach is proposed in [21].

With the development of the WWW and the diffusion of the de facto standards for the definition of structured documents, such as XML, the need has emerged for flexible query languages based on semistructured documents. In the context of semistructured databases and flexible query languages it is crucial to define query languages that take into account the lack of a rigid schema of the database, thus allowing a user to query both data and the type/schema [50]. In the context of IR systems, modeling flexibility means taking into account the possibility of making explicit a non-uniform structure of the documents when formulating queries.

In [51] fuzzy set theory has been applied to define an extension of the XPath query language for expressing soft selection conditions on both the document structure and its contents. The extensions of XPath proposed in [51] yield:

- fuzzy subtree matching to the aim of providing a ranked list of retrieved information items rather than the usual set oriented one;
- use of fuzzy predicates to the aim of specifying flexible selection conditions;
- fuzzy quantification, with the aim of allowing the specification of linguistic quantifiers as aggregation operators.

This research constitutes a step toward solving the popular problem of querying XML documents, not only from a structural point of view, but also from a content-based point of view [52].

A generalization of the Boolean query language allowing personalization of search in structured documents has been proposed. Here, both content-based selection constraints and soft constraints on the document structure can be expressed. The atomic component of the query (basic selection criterion) is simple; for example, ‘ t in Q preferred sections,’ in which t is a search term expressing a content-based selection constraint, and Q is a linguistic quantifier such as *all*, *most*, or *at least k%*. Q expresses a part of the structure-based selection constraint. It is assumed that the quantification refers to the sections that are semantically meaningful to the user. Q is used to aggregate the significance degrees of t in the desired sections and then to compute the global RSV of the document d with respect to the atomic query condition [29]. The complexity of the evaluation of quantified queries mainly depends on the number of search terms to aggregate since the evaluation of the quantifier is based on the application of an OWA operator that requires an ordering of its arguments.

Another approach that tries to generate a query formulation that represents a user’s information need is based on query expansion through relevance feedback. Once a query has been processed, the user is presented with a list of retrieved records. The user indicates which records on at least a subset of that list have been examined and found to be either relevant or non-relevant. Then, the system refines the query to incorporate that information, so that when the refined query is processed, records similar to those the user says are relevant will be retrieved (or at least ranked high in the list of retrieved records), while those records similar to those the user says are non-relevant will not be retrieved (or at least will be ranked lower in the list of retrieved records). Assuming a weighted Boolean query, research has been conducted using genetic algorithms (genetic programming, to be precise) to determine an optimal (or near-optimal) refined query [53–58].

38.5 Document Clustering

The concept of clustering represents the notion of putting similar objects, such as documents, together. One can consider documents as vectors of term weights and use similarity measures such as the cosine measure to cluster them [11]. The similarity between documents in the same cluster should be large, while this similarity between documents in different clusters should be small.

A common method to perform clustering of documents is based on the simultaneous occurrences of citations in pairs of documents. Documents are clustered using a measure defined on the space of the citations. Generated clusters can then be used as an index for IR; i.e., documents that belong to the same cluster(s) as the documents directly indexed by the terms in the query are retrieved. Often, similarity measures are suggested empirically or heuristically. When adopting fuzzy clustering techniques, a fuzzy partition of the document space is created in which each fuzzy cluster is defined by a fuzzy set of documents. In this way, each document is assigned a distinct membership value to each cluster [36, 59–63]. In a pure fuzzy clustering, a complete overlap of clusters is allowed. Modified fuzzy clustering, or soft clustering, uses threshold mechanisms to limit the number of documents belonging to each cluster. The main advantage of using modified fuzzy clustering is that the degree of fuzziness is controlled. In [35] an incremental hierarchical fuzzy clustering algorithm has been defined to the aim of identifying the main categories of news in a news-stream information filtering system. It generates a hierarchy of fuzzy clusters, which may deal with multiple topics with different levels of granularity.

Another issue is the use of clustering in IR on the WWW. Noting that search engines retrieve multiple occurrences of the same documents with possibly different degrees of relevance, in [64] it was observed that fuzzy multisets provide an appropriate modeling for both term clustering and document clustering.

38.6 Fuzzy Approaches to Distributed Information Retrieval

With the increasing use of network technology, the need to generate distributed retrieval applications has emerged. In distributed IR, there are two main models. In the first model, the information is considered as belonging to a unique, perhaps large, centralized database, which is distributed, but centrally indexed, for

retrieval purposes. This is the model adopted by search engines on the WWW. A second model is based on the distribution of the information on distinct databases, independently indexed, and thus constituting distinct sources of information. This last model gives rise to the so-called distributed, or multisource, IR problem. In this second case, the databases reside on distinct servers, each of which can be provided with its own search engine (IR system).

The multisource IR paradigm is more complex than the centralized model. This paradigm presents additional problems, such as the selection of an appropriate information source for a given information need. This task of distributing retrieval is affected by uncertainty, since a decision must be taken based on an incomplete description of the information source.

Furthermore, a common problem in both models is list fusion task. In the case in which we have a centralized information repository and distinct IR systems (or search engines) to search overlapping collections, meta-search engines have been defined to improve the effectiveness of the individual search engines. The main aim of a meta-search engine is to submit the same query to distinct search engines and to fuse the individual resulting lists into an overall ranked list of documents that is presented to the user. In this case we typically have overlapping individual lists, since more than a single search engine may retrieve a document. A fusion method must be employed in order to then be able to handle situations in which a document may appear in more than one list and in different retrieved list positions.

In the case of multisource IR, the problem is to merge the lists resulting from the processing of the same query by (generally distinct) search engines on the distinct databases residing on distinct servers. However, in this case, we generally do not have overlapping lists as a result of the same query evaluation. Typically, a document will be retrieved by just one single search engine, and thus the fusion problem is simplified with respect to the previous case.

Lately, defining effective solutions to the problem of retrieving information on a network has been addressed. Some approaches to the definition of meta-search engines have been presented in [65], while some solutions to the problem of multisource IR have been described [66]. These approaches are based on soft computing techniques to model more flexibly the resource selection problem in distributed IR and the list fusion problem. In particular, in [65] soft fusion of overlapping ordered lists into an overall ordered list is modeled as a group decision-making activity in which the search engines play the role of the experts, the documents are the alternatives that are evaluated based on a set of criteria expressed in a user query, and the decision function is a soft aggregation operator modeling a specific user retrieval attitude.

38.7 Conclusion

IR is characterized by a lack of precision at several different levels. The indexing process can generate a document representation affected by incompleteness and a pervasive vagueness. User queries are often vague. Then, the retrieval mechanism is faced with the uncertainty by which the results satisfy the query. The retrieval models designed till now do not distinguish among the different kinds of imperfect information. In this chapter, some approaches to the definition of flexible IR, including the application of fuzzy set theory, have been presented. Based on soft approaches, some commercial systems have incorporated fuzzy matching techniques among their functionalities. Examples of applications are the adoption of fuzzy rules for stemmers or string partial matching and soft operators to model the implicit aggregation when listing search terms in a query. In particular, in this chapter some promising research directions that should guarantee the development of more effective IR systems have been outlined. Among these, the research efforts aimed at defining new indexing techniques of semistructured documents, such as XML documents, are very important. The possibility of creating, in a user-driven manner, document surrogates would ensure a modeling of a user's interests at the indexing level, instead of the usual manner of limiting this to the query formulation level. Other promising directions are constituted by conceptual document indexing, flexible IR, and distributed IR. One key issue that is going to be more and more investigated is the extension of IR to other than text documents, i.e., non-print media. This includes not just images, video, and audio, but also thematic maps of the territory [67, 68]. GeoMap services are being more and more diffused over the Internet and the integration of their functionalities within traditional

search engines is a challenge. Another important issue is cross-language IR. This topic, often seen under the acronym CLIR, refers to the notion of retrieving information written in a language different from the language of the user query [69]. For example, a user could pose his/her query in English but retrieve relevant documents written in Italian. This aspect implies either an automatic translation of a query and document representations or the conceptual indexing of both query and documents.

References

- [1] G. Bordogna and G. Pasi. Modelling vagueness in information retrieval. In: M. Agosti, F. Crestani, and G. Pasi (eds), *Lectures in Information Retrieval*. Springer-Verlag, Berlin, 2001, pp. 207–241.
- [2] D. Kraft, G. Bordogna, and G. Pasi. Fuzzy set techniques in information retrieval. In: J.C. Bezdek, D. Dubois, and H. Prade (eds), *Fuzzy Sets in Approximate Reasoning and Information Systems*, Kluwer Academic Publishers, MA, 1999, pp. 469–510.
- [3] G. Pasi. Modelling users' preferences in systems for information access, *Int. J. Intell. Syst.* 18 (2003) 793–808.
- [4] E.J. Glover, S. Lawrence, M.D. Gordon, W.P. Birmingham, and C. Giles. Web search – YourWay. *Commun. ACM* 44(12) (1999) 97–102.
- [5] G. Salton. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison Wesley, Reading, MA, 1989.
- [6] O. Cordon and E. Herrera Viedma. Special issue on soft computing applications to intelligent information retrieval on the internet. *Int. J. Approx. Reason.* 34 (2003) 2–3.
- [7] F. Crestani and G. Pasi (eds). *Soft Computing in Information Retrieval: Techniques and Applications, Studies in Fuzziness Series*, Physica-Verlag, Heidelberg, 2000.
- [8] E. Herrera-Viedma, G. Pasi, and F. Crestani (eds). *Soft Computing in Web Information Retrieval: Models and Applications, Studies in Fuzziness and Soft Computing Series*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [9] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Wokingham, UK, 1999.
- [10] C.J. van Rijsbergen. *Information Retrieval*. Butterworths & Co., Ltd., London, England, 1979.
- [11] G. Salton, and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.
- [12] K.A. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *J. Doc.* 28(1) (1972) 11–20.
- [13] G. Bordogna and G. Pasi. Multicriteria decision making in information retrieval. In: *Proceedings of 3rd International Conference on Current Issues in Fuzzy Technologies '93*, Roncegno, Italy, 1993, pp. 3–10.
- [14] P. Vincke. *Multicriteria Decision Aid*. John Wiley & Sons, NJ, 1992.
- [15] C.L. Barry. User-defined relevance criteria: An exploratory study. *J. Am. Soc. Inf. Sci.* 45(3) (1994) 149–159.
- [16] F. Crestani and G. Pasi. Soft information retrieval: Applications of fuzzy set theory and neural networks. In: N. Kasabov and R. Kozma (eds), *Neuro-fuzzy Techniques for Intelligent Information Systems*. Springer-Verlag, Telos, Berlin, 1999, pp. 287–313.
- [17] A. Bookstein. Fuzzy requests: An approach to weighted Boolean searches. *J. Am. Soc. Inf. Sci.* 31(4) (1980) 240–247.
- [18] G. Bordogna and G. Pasi. Controlling retrieval through a user-adaptive representation of documents. *Int. J. Approx. Reason.* 12 (1995) 317–339.
- [19] G. Bordogna and G. Pasi. An ordinal information retrieval model. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 9 (2001) 63–75.
- [20] D.A. Buell and D.H. Kraft. Threshold values and Boolean retrieval systems. *Inf. Process. Manage.* 17 (1981) 127–136.
- [21] F. Diaz-Hermida, D. Losada, A. Bugarin, and S. Barro. A probabilistic quantifier fuzzification mechanism: The model and its evaluation for information retrieval. *IEEE Trans. Fuzzy Syst.* 13(1) (2005) 688–700.
- [22] E. Herrera-Viedma. Modeling the retrieval process of an information retrieval system using an ordinal fuzzy linguistic approach. *J. Am. Soc. Inf. Sci. Technol.* 52 (2001) 460–475.
- [23] E. Herrera-Viedma, O. Cordon, M. Luque, A.G. Lopez, and A.N. Muñoz. A model of fuzzy linguistic IRS based on multi-granular linguistic information. *Int. J. Approx. Reason.* 34(3) (2003) 221–239.
- [24] T. Radecki. Fuzzy set theoretical approach to document retrieval. *Inf. Process. Manage.* 15(5) (1979) 247–260.
- [25] D.H. Kraft, G. Bordogna, and G. Pasi. Information retrieval systems: Where is the fuzz? In: *Proceedings of IEEE International Conference on Fuzzy Systems*, Anchorage, Alaska, 1998, pp. 1367–1372.
- [26] M. Bouganem, Y. Loiseau, and H. Prade. Improving document ranking in information retrieval using ordered weighted aggregation and lexicin refinement. In: *Proceedings of 4th Conference of the European Society for Fuzzy Logic and Technology and 11me Rencontres Francophones sur la Logique Floue et ses Applications*, EUSFLAT-LFA, Barcelona, Spain, 2005, pp. 1269–1274.

- [27] A. Brini, M. Boughanem, and D. Dubois, A model for information retrieval based on possibilistic networks. In: *Proceedings of String Processing and Information Retrieval (SPIRE 2005)*, LNCS, Buenos Aires, Argentine. Springer-Verlag, Berlin, 2005, pp. 271–282.
- [28] Y. Loiseau, M. Boughanem, and H. Prade. Evaluation of term-based queries using possibilistic ontologies. In: E. Herrera-Viedma, G. Pasi, and F. Crestani (eds), *Soft Computing for Information Retrieval on the Web*. Springer-Verlag, Berlin, Heidelberg, 2006, pp. 16–26.
- [29] G. Bordogna and G. Pasi. Personalized indexing and retrieval of heterogeneous structured documents. In: *Information Retrieval*, Vol. 8, no. 2. Kluwer, Dordrecht, 2005, pp. 301–318.
- [30] R.A. Marques Pereira, A. Molinari, and G. Pasi. Contextual weighted representations and indexing models for the retrieval of HTML documents. *Soft Comput.* 9(7) (2005) 481–492.
- [31] A. Molinari and G. Pasi. A fuzzy representation of HTML documents for information retrieval systems. In: *Proceedings of the IEEE International Conference on Fuzzy Systems*, Vol. 1, New Orleans, 1996, pp. 107–112.
- [32] S. Miyamoto. *Fuzzy Sets in Information Retrieval and Cluster Analysis*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.
- [33] S. Miyamoto. Information retrieval based on fuzzy associations. *Fuzzy Sets Syst.* 38(2) (1990) 191–205.
- [34] K. Nomoto, S. Wakayama, T. Kirimoto, and M. Kondo. A fuzzy retrieval system based on citation. *Syst. Control* 31(10) (1987) 748–755.
- [35] G. Bordogna, M. Pagani, and G. Pasi. A dynamical hierarchical fuzzy clustering algorithm for document filtering. In: E. Herrera-Viedma, G. Pasi, and F. Crestani (eds), *Soft Computing in Web Information Retrieval: Models and Applications, Studies in Fuzziness and Soft Computing*. Springer-Verlag, 2006, pp. 2–15.
- [36] R.J. Hathaway, J.C. Bezdek, and Y. Hu. Generalized fuzzy c-means clustering strategies using L_p norm distances. *IEEE Trans. Fuzzy Syst.* 8(5) (2000) 576–582.
- [37] D.H. Kraft, and D.A. Buell. Fuzzy sets and generalized Boolean retrieval systems. *Int. J. Man-Mach. Stud.* 19(1) (1983) 45–56.
- [38] L.A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Comput. Math. Appl.* 9 (1983) 149–184.
- [39] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. Syst. Man Cybern.* 18(1) (1988) 183–190.
- [40] L. Azzopardi, M.L. Girolami, and C.J. van Rijsbergen. Topic based language models for ad hoc information retrieval. In: *Proceedings of International Joint Conference on Neural Networks*, Budapest, Hungary, 2004, pp. 3281–3286.
- [41] R. Thomopoulos, P. Buche, and O. Haemmerlé. Representation of weakly structured imprecise data for fuzzy querying. *Fuzzy Sets Syst.* 140 (2003) 111–128.
- [42] M. Boughanem, G. Pasi, H. Prade, and M. Baziz. A fuzzy logic approach to information retrieval using an ontology-based representation of documents. In: E. Sanchez (ed.), *Fuzzy Logic and the Semantic Web*. Elsevier Science, Amsterdam, 2006, pp. 363–377.
- [43] Y. Ogawa, T. Morita, and K. Kobayashi. A fuzzy document retrieval system using the keyword connection matrix and a learning method. *Fuzzy Sets Syst.* 39(2) (1991) 163–179.
- [44] P. Srinivasan, M. Ruiz, D.H. Kraft, and J. Chen. Vocabulary mining for information retrieval: Rough sets and fuzzy sets. *Inf. Process. Manage.* 37 (2001) 15–38.
- [45] D.H. Kraft, Advances in information retrieval: Where is that $\#*\%^@^$ record? In: M. Yovits (ed.), *Advances in Computers*, Vol. 24. Academic Press, New York, 1985, pp. 277–318.
- [46] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. Parts I and II. *Inf. Sci.* 8 (1975) 199–249, 301–357.
- [47] G. Bordogna and G. Pasi. A fuzzy linguistic approach generalizing Boolean information retrieval: A model and its evaluation. *J. Am. Soc. Inf. Sci.* 44(2) (1993) 70–82.
- [48] D.H. Kraft, G. Bordogna, and G. Pasi. An extended fuzzy linguistic approach to generalize Boolean information retrieval. *J. Inf. Sci.* 2(3), (1994) 119–134.
- [49] G. Bordogna and G. Pasi. Linguistic aggregation operators in fuzzy information retrieval. *Int. J. Intell. Syst.* 10(2) (1995) 233–248.
- [50] S. Abiteboul. Querying semi-structured data. In: *Proceedings of the 6th International Conference on Database Theory*, LNCS. Springer-Verlag, Berlin, 1997, pp. 1–18.
- [51] D. Braga, A. Campi, E. Damiani, G. Pasi, and P. Lanzi. XPath: Flexible querying of XML documents. In: *Proceedings of EUROFUSE*, Varenna, Italy, 2002, pp. 15–21.
- [52] N. Fuhr and M. Lalmas. Introduction to the special issue on INEX. *Inf. Retr.* 8(4) (2005) 515–519.
- [53] D.H. Kraft, F.E. Petry, B.P. Buckles, and T. Sadasivan. Applying genetic algorithms to information retrieval systems via relevance feedback. In: P. Bosc and J. Kacprzyk (eds), *Fuzziness in Database Management Systems, Studies in Fuzziness Series*. Physica-Verlag, Heidelberg, 1995, pp. 330–344.

- [54] D.H. Kraft, F.E. Petry, B.P. Buckles, and T. Sadasivan. Genetic algorithms for query optimization in information retrieval: Relevance feedback. In: E. Sanchez, T. Shibata, and L.A. Zadeh (eds), *Genetic Algorithms and Fuzzy Logic Systems*. World Scientific, Singapore, 1997, pp. 155–173.
- [55] F.E. Petry, B.P. Buckles, D.H. Kraft, D. Prabhu, and T. Sadasivan. The use of genetic programming to build queries for information retrieval. In: T. Baeck, D. Fogel, and Z. Michalewicz (eds), *Handbook of Evolutionary Computation*, Section G2.1. Oxford University Press, New York, 1997, pp. 1–6.
- [56] O. Cordon, F. Moya, and C. Zarco. A new evolutionary algorithm combining simulated annealing and genetic programming for relevance feedback in fuzzy information retrieval systems. *Soft Comput.* 6(5) (2002) 308–319.
- [57] O. Cordon, F. Moya, and C. Zarco. Automatic learning of multiple extended Boolean queries by multiobjective GA-P algorithms. In: V. Loia, M. Nikravesh, and L.A. Zadeh (eds), *Fuzzy Logic and the Internet*. Springer-Verlag, Berlin, Heidelberg, 2004, pp. 47–70.
- [58] O. Cordon, E. Herrera-Viedma, and M. Luque. A multiobjective genetic algorithm for linguistic persistent query learning in text retrieval. In: Y. Jin (ed.), *Multi-objective Machine Learning, Studies, Computational Intelligence Series, 16*. Springer-Verlag, Berlin, Heidelberg, 2006, pp. 601–627.
- [59] D. Kraft, J. Chen, M.J. Martin-Bautista, and M.A. Vila. Textual information retrieval with user profiles using fuzzy clustering and inferencing. In: P. Szczepaniak, J. Segovia, J. Kacprzyk, and L.A. Zadeh (eds), *Intelligent Exploration of the Web, Studies in Fuzziness and Soft Computing Series*, Vol. III. Physica-Verlag, Heidelberg, 2003, pp. 152–165.
- [60] D.H. Kraft, M.J. Martin-Bautista, J. Chen, and D. Sanchez. Rules and fuzzy rules in text: Concept, extraction and usage, special issue: Soft computing applications to intelligent information retrieval on the Internet. *Int. J. Approx. Reason.* 34(2–3) (2003) 145–162.
- [61] K. Lin and K. Ravikuma. A similarity-based soft clustering algorithm for documents. In: *Proceedings of the 7th International Conference on Database Systems for Advanced Applications*, March 26–28, 2003, Kyoto, Japan, pp. 40–47.
- [62] M. Mendes, M.E.S. Rodrigues, and L. Sacks. A scalable hierarchical fuzzy clustering algorithm for text mining. In: *Proceedings of the 4th International Conference on Recent Advances in Soft Computing*, RASC’2004, Nottingham, UK, 2004, pp. 269–274.
- [63] W. Pedrycz. Clustering and fuzzy clustering. In: *Knowledge-Based Clustering*. John Wiley and Sons, Hoboken, NJ, 2005, Chapter 1.
- [64] S. Miyamoto. Modelling vagueness and subjectivity in information access. *Inf. Process. Manage.* 39(2) (2003) 195–213.
- [65] G. Bordogna and G. Pasi. Soft fusion of infomation accesses. *Fuzzy Sets Syst.* 148 (2004) 205–218.
- [66] G. Bordogna, G. Pasi, and R.R. Yager. Soft approaches to distributed information retrieval. *Int. J. Intell. Syst.* 34(2003) 105–120.
- [67] S.S. Iyengar. Visual based retrieval systems and web mining – introduction, *J. Am. Soc. Inf. Sci. Technol.* 52 (2001) 829–830.
- [68] J.S. Downie. A sample of music information retrieval approaches. *J. Am. Soc. Inf. Sci. Technol.* 55 (2004) 1033–1036.
- [69] C.C. Yang, and W. Lam. Introduction to the special topic secion on multilingual information systems. *J. Am. Soc. Inf. Sci. Technol.* 57 (2006) 629–631.

39

Granular Computing in Medical Informatics

Giovanni Bortolan

39.1 Introduction

In the last years there is an increasing interest in developing useful tools for data analysis, data mining, and knowledge discovery, with the aim to support human experts to make correct decisions. Fuzzy set theory provides a mathematical basis and a powerful framework for handling the complexity of classification knowledge and it has been applied to many areas extensively [1]. In addition, fuzzy modeling has emerged as an interesting, attractive, and powerful environment applied to numerous system identification tasks [2–5]. In the same time, specific studies in data mining and knowledge discovery are aimed at providing the user with a transparent and meaningful description of significant dependencies in experimental data [6–8]. It has been empathized that the individualization of different abstraction levels, the employment of information granule, and the use of a proper model can improve the generalization capability and the accuracy of the classification or prediction tasks [9,10].

The most frequent approaches of indirect knowledge acquisition are based on a detailed and purely numeric level and they provide a powerful tool that analyzes the input patterns and produces a classification. This means that the initial step of quantification phase is followed by a classification block as illustrated in Figure 39.1. With this schema, all the available data are processed by the classifier model, and some feature selection or several optimization techniques are adopted for increasing the capacity of the learning process. This strategy shows a critical balance between generalization and approximation abilities, and there is no possibility to describe, express, or formalize the different abstraction levels in the learning process. In fact, in this case all the input data points weigh as single points, and they are equally considered at the same level [11].

In case the data points are considered not as single points but as groups or granules, the concept of data abstraction and information granulation occurs. In fact considering that ‘the **information** may be said to be **granular** in the sense that data points within a granule have to be dealt with as a whole rather than individually’ [12], a more realistic schema with a more expressive power is illustrated in Figure 39.2. In this schema a particular descriptive modeling is inserted after the quantification phase, with the main objective to describe or express the input data with a suitable and appropriate description or characterization.

In the process of diagnostic classification, enormous quantities of numerical details are processed and considered at different levels of data abstraction, and particular data quantification and different data

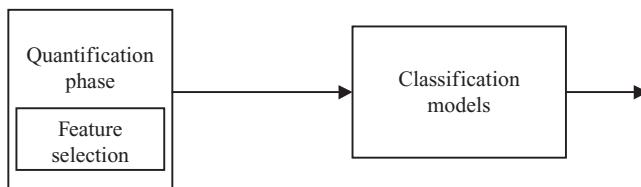


Figure 39.1 A general scheme of the classic approach of diagnostic classification

aggregation are performed. In order to understand this process it is important to model and formalize it with expressive schema/tools, for an accurate approximation of the real process and for an ‘efficient learning.’

In this chapter, first we describe the organization and the characteristics of the ‘information granular computing’ schema, describing the main components of it, with particular emphasis on descriptive models and predictive models. Then in Section 39.3 we present a real complex problem of diagnostic classification: the computerized ECG classification examined with two validated databases. In Section 39.4 three descriptive models will be described in detail: self-organized maps (SOM) model, the radial basis Functions (RBF) model, and the linguistic model. For every model, the application to ECG diagnostic classification will be considered, and the corresponding accuracy of the entire classification task will be described and compared.

39.2 Information Granular Computing

The different abstraction levels of the process of diagnostic classification in medical field are considered in the framework of granular computing. In this process, three different and characteristic blocks can be individuated: (1) quantification phase, (2) descriptive models, and (3) predictive models (see Figure 39.2).

39.2.1 Quantification Phase

The quantification phase represents the first level, in which the input features are quantified or categorized. In this block, a feature selection or a dimensionality reduction can be performed, either for a more efficient and optimized learning process or to restrict the inference model to the particular classification task taken into consideration. These techniques are more effective in case they are supported by the domain knowledge [13–16].

39.2.2 Descriptive Models

The descriptive models represent the next level of abstraction. The descriptive models are aimed at the description of data in the language of well-defined, semantically sound, and user-oriented information

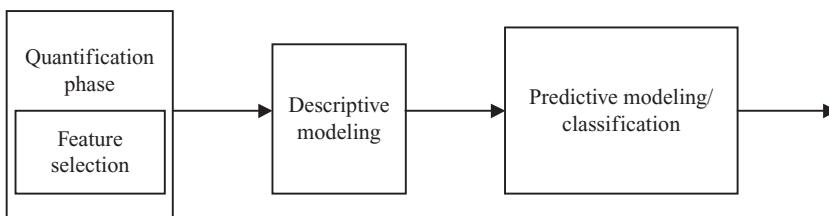


Figure 39.2 Characterization of the classification task: quantification phase, descriptive, and predictive models

granules [10, 12]. Here the designer plays a significant role, and he/she may interact and insert the knowledge in the architecture of the model. In general, they characterize or determine the entire architecture of the classification task.

The descriptive fuzzy model can be developed in different ways:

- It can be a use-oriented development environment (highly interactive, efficient, and user-friendly visualization environment).
- It can be a data-driven model, in which the architecture is based on the data.
- It can be a knowledge model as in the knowledge-based systems.

Three different descriptive models have been considered and described: the SOM models, the RBF models, and the linguistic model.

The SOM model is a powerful tool for structure discovery and visualization. The intent of SOM models is to visualize highly dimensional data in a low-dimensional structure (two- or three-dimensional maps). This low-dimensional map should preserve some topological properties and should extract or highlight some information granules. Similarity relations in the original feature space or in a particular diagnostic space should be preserved or pointed out in the low-dimensional space.

The RBF model and the linguistic model perform a first phase of data abstraction level and they characterize the input patterns with a set of membership functions represented by RBF or by linguistic descriptors of the input features.

39.2.3 Predictive Models/Classification Models

The last level of abstraction is represented by predictive models, which finds the relevant pieces of information useful for the classification task and quantifies the relevance of information granules. In this block, the central point is the accuracy and the robustness of the inferred ‘relevant pieces of information,’ and a strict connection with the ‘domain knowledge’ can increase its efficacy. Generally, these models are strictly connected with the architecture or the structure of the entire classification block. In case of unsupervised models, a highly interactive and user-friendly environment is the optimal solution in order to quantify the accuracy of the relevant pieces of information extracted by the descriptive model. In case of supervised algorithm, the descriptive model represents, in general, the first abstraction level, and the predictive block is then developed in the same framework for a further refinement of the classification task. For this reason, the described predictive models are specific to the three descriptive models considered in this chapter.

39.3 The Problem of ECG Classification

39.3.1 The ECG Classification

The ‘complex’ problem of diagnostic classification of ECG signal is used to test and validate the proposed methods. The automatic ECG analysis is one of the most important medical diagnostic topics, since it allows to diagnose cardiac diseases, often connected to sudden death, using a non-invasive technique. The medical knowledge is not very deterministically structured, since the same signs can point out very different pathologies in different context or a slight change in the data can mean a serious change in the patient state. For this reason, any medical problem needs a very careful data description and the possibility to treat different level of granulation, to manage the whole signal information in a satisfactory manner.

A rest ECG signal is composed by 12 standard leads (of which only 8 are independent) acquired generally at the frequency of 500 samples per second for a period of 10 s, for a total of 60,000 samples.

It is very hard to apply any classification method directly to the raw ECG samples. Consequently, a first data abstraction is performed directly to the digital signal in order to capture specific ECG features. For this process, classical pattern recognition techniques in combination to a data abstraction stage, in order to reduce the data dimension and to save much information as possible, is performed [17, 18].

The most commonly investigated ECG diagnostic classes, which can be confirmed by independent clinical data, refer chronic diseases as left ventricular hypertrophy (LVH), right ventricular hypertrophy (RVH), biventricular hypertrophy (BVH), inferior myocardial infarction (IMI), anterior myocardial infarction (AMI), mixed myocardial infarction (MIX), and obviously the normal/healthy status. The first six diagnostic classes can be grouped at a different granular level: LVH, RVH, and BVH as ventricular hypertrophy (VH), and IMI, AMI, and MIX as myocardial infarction (MI). Three validation indices were considered for the validation and comparison of the different methods: the sensitivity (the proportion of correctly classified subjects with the specific class), the specificity (the proportion of correctly classified subjects without the specific class), and the total accuracy (proportion of correctly classified subjects). In addition, the mean sensitivity and the mean specificity across the seven diagnostic classes have been considered.

39.3.2 The ECG Database

The learning capacity of diagnostic classifiers is dependent on both the composition and the consistency of the database. For this reason it is essential the use of clinically validated databases. In this work, two databases have been considered: the ECG-CORDA and the ECG-UCL databases.

The ECG-CORDA database was developed at the Department of Medical Informatics of the University of Leuven and tested in other studies with other classical classification techniques [19–21]. The database consists of 3266 rest ECG records concerning 2158 males and 1108 females. It is validated by independent ECG clinical data. For supervised techniques, a random set of 2446 patients is selected for the learning phase and 820 for the testing phase. Each record is characterized by 540 primary measurements (45 per 12 leads) obtained by a computerized ECG program. A first feature selection was made by a clinical selection, individuating a set of 166 parameters. By a statistical analysis, the subset of 39 most significant parameters regarding the considered diagnostic classes were selected. This reduced subset of 39 parameters represents the set of input feature of the proposed classifiers.

The ECG-UCL database was developed by Christian Brohet at the Cliniques Universitaires Saint-Luc, Université Catholique de Louvain [17, 19]. It is composed of 2854 rest ECG records (1806 men and 1048 women) with clinical validation and the presence of a single diagnosis. For the present study a subset of 539 records was randomly selected. The learning set (testing set) was composed by a random selection of 404 (135) ECG records. In particular, each group of AMI, IMI, and MIX subjects consists of 75 (25) records and 179 (60) for the ‘other’ group. Every ECG signal was characterized by 276 standard parameters, which were extracted with classic pattern recognition techniques by the Padova program [18].

The composition of the two databases is different in the seven considered diagnostic classes. The ECG-CORDA (ECG-UCL) database has the following composition: 16.5% (37.5%) normal, 17.1% (8.0%) LVH, 9.9% (1.7%) RVH, 13.4% (1.0%) BVH, 11.9% (14.1%) AMI, 20.1% (28.0%) IMI, and 11.1% (9.8%) MIX. This difference will influence the accuracy of any diagnostic classifier. The composition of the ECG-CORDA database is more balanced. In fact, it has 16.5% normal, 40.4% VH, and 43.1% MI, whereas the ECG-UCL is characterized by 37.0% normal, 10.7% VH, and 52.3% MI. For this reason, a table that summarizes the accuracy of different classifiers with the two ECG databases will be considered and reported for helping the validation procedure. Table 39.1 reports the mean sensitivity, the mean specificity, and the total accuracy of linear discriminant analysis (LDA), logistic discriminant analysis (LOG), and neural network (NN) approaches using the mentioned ECG databases in previous works [17, 19–21]. From this table we can evidence that the accuracy and the sensitivity of the two databases are significantly different, and in the validation phase these limits represent an effective comparison.

Table 39.1 Comparison of mean sensitivity, mean specificity, and total accuracy of different classifiers

Database	Method	Sensitivity (%)	Specificity (%)	Accuracy (%)
ECG-CORDA	LDA	63.2	–	67.2
ECG-CORDA	LOG	62.8	–	66.3
ECG-CORDA	NN	64.3	94.6	68.8
ECG-UCL	NN	80.4	95.5	83.0

LDA, linear discriminant analysis; LOG, logistic discriminant analysis; NN, neural networks, with the ECG-CORDA and ECG-UCL databases.

39.4 Descriptive Models

39.4.1 The SOM Models

The SOMs, originally developed by Kohonen [22, 23], are a very powerful tool for structure discovery and visualization in many application areas. It represents an outstanding example of unsupervised learning. The main intent of SOMs is to visualize highly dimensional data in a low-dimensional structure, for example, by two- or three-dimensional maps [24]. This low-dimensional map preserves some topological properties. In particular, two data points ‘similar’ that are close to each other in the original feature space or in the diagnostic space should retain this similarity in the low-dimensional space, and consequently two ‘distant’ patterns in the original feature space should maintain this distance. Usually, they are regarded as regular neural network structures composed by a rectangular grid of artificial neurons.

Each neuron is equipped with a modifiable connection or weight, $w(i, j)$, characterized by a n -dimensional vector:

$$w(i, j) = [w_1(i, j), w_2(i, j), \dots, w_n(i, j)],$$

where n is the dimension of the input vector X , and the indices \mathbf{i} and \mathbf{j} identify the interested node. The neuron calculates the distance d between its connections and a certain input vector X :

$$y(i, j) = d(w(i, j), x).$$

The input vector X affects all the neurons of the map, and the one exhibiting the shortest distance is considered as the ‘winning node’ (i^*, j^*):

$$(i^*, j^*) = \arg \min_{(i,j)} d(w(i, j), x),$$

and the corresponding weights are modified in order to decrease the distance with the input data and upgraded as

$$w_new(i^*, j^*) = w(i^*, j^*) + \alpha(x - w(i^*, j^*)),$$

where α is a learning rate $\alpha > 0$.

The update mechanism of this architecture provides an additional modification of the nodes which are the neighbors of the ‘winning node.’ This is performed through a particular neighbor function

$$\Phi(i, j, i^*, j^*),$$

and consequently the general formula for the update algorithm of the weights of all the nodes is the following:

$$w_new(i, j) = w(i, j) + \alpha\Phi(i, j, i^*, j^*)(x - w(i^*, j^*)).$$

A usual neighbor function is the following:

$$\Phi(i, j, i^*, j^*) = \exp(-\beta((i - i^*)^2 + (j - j^*)^2)),$$

where β is a parameter which models the spread of the neighbor function. The parameter β is modified dynamically, reducing it in the final part of the learning process, for a more selective grouping or cluster. Various distance functions $d(*)$ have been used in the different application area, and the Euclidean distance is the most common preference, like in the present chapter.

An additional useful and usual preprocessing step is given by the normalization process, in order to have a more homogeneous weight of the different input parameters. Once the learning process is performed, the consequent SOM is completely characterized by weight matrix \mathbf{W} , which is the outcome of the unsupervised learning process. In order to develop a highly interactive and user-friendly environment for ECG signal analysis, and permit the designer to proceed in the knowledge discovery, additional tools in the framework of ‘predictive modeling’ may be developed and tested.

At this point, we have at disposal several ways to describe, characterize, or classify the produced map for inferring or deducing significant dependencies in the experimental data and for supporting human experts to make correct decisions. For example, different techniques can be adopted to describe or extract the information granules at different abstraction levels. These techniques may be considered as part of the classification or predictive blocks to extract the relevant pieces of information and to quantify the accuracy and robustness.

Three methods of interpreting the \mathbf{W} matrix are analyzed.

The most immediate way to consider the weight matrix is to view it as a set of n maps:

$$[W_1(*, *), W_2(*, *), W_3(*, *), \dots, W_i(*, *), \dots, W_n(*, *)]$$

one for each considered feature. In this way it is possible either to evaluate the classification ability of single features or to discover possible association between different features.

A second way of interpreting the SOM \mathbf{W} is to consider the median amount of the distances between one node and its closets neighbors, and this is a measure of homogeneity of (i, j) location. It is possible to grade this measure on the entire map $W(*, *)$ in a gray/brightness scale.

A third useful method to interpret the map \mathbf{W} is the use of the data distribution density map. This method considers the distribution of the input data as they are allocated to the individual neurons on the map and it is shown on a certain gray/brightness scale. The gray scale is graduated and proportional to the number of winning times for every node; that is, the darker the color of the neuron, the more patterns invoked the neuron as the winning one. It is possible to define visually different clusters and homogeneous regions. The distribution of the diagnostic classes in these clusters can be considered as a starting point for the classification of input data, and several attempts of grouping the more homogeneous clusters may be conducted. This analysis was used in the considered application for the characterization and the classification of the input data.

This technique was tested with the ECG-CORDA database. The set of 2446 ECG records was used, with the seven diagnostic classes: 402 normal, 417 LVH, 243 RVH, 324 BVH, 295 AMI, 489 IMI, and 276 MIX. Each record consists of 39 parameters, normalized by a linear transformation. Several sizes of the SOM were tested, and the size of 35×35 was a reasonable compromise between the need of a high accuracy and the computation time [24]. The learned map is reported in Figure 39.3. Once the matrix \mathbf{W} was discovered, the most useful way to extract significant dependencies in the experimental data was represented by the analysis and of the data distribution density map. In this way, several clusters were recognized and characterized by a visual inspection. In particular, eight homogeneous regions were identified (see Figure 39.4).

The analysis of the distribution of the seven diagnostic classes in the eight clusters is reported in Table 39.2. The patterns occurring in these regions represent the 38% of all data. It is possible to observe that every cluster is mainly connected with one kind of diagnostic classes. The distribution, the topological properties, and the spatial information can be helpful for the classification task.

In particular, the cluster C_4 is able to capture normal signals, and it is quite compact and it shows a high level of homogeneity.

The region C_2 is quite large and separated from other regions. It consists of MI subjects, mainly of AMI class.

The cluster C_1 involves mainly patients classified as IMI and MIX. Consequently, C_1 , C_2 , and C_3 capture mainly MI subjects.

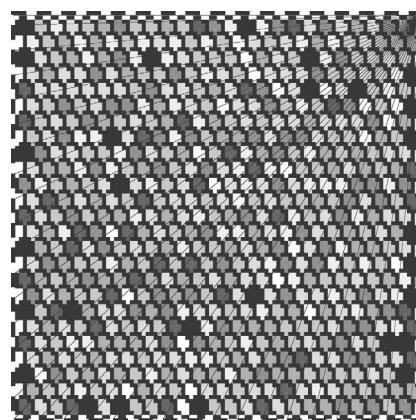


Figure 39.3 Example of a two-dimensional SOM visualized by a data distribution density map

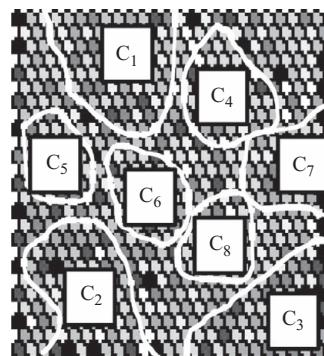


Figure 39.4 Clusters of a distribution density SOM

Table 39.2 Identification of eight homogeneous region in the SOM 35×35

Region	Class						
	Norm	LVH	RVH	BVH	IMI	AMI	MIX
C ₁	1				44	8	61
C ₂					7	246	50
C ₃				2	41	34	18
C ₄	118	38	4	2			
C ₅		8	7	22	1		
C ₆	6	10	47	22	1		
C ₇	84	6	11	6	1		
C ₈		26			1		1

Table 39.3 Classification of the eight clusters with the majority rule

Region	Majority rule	Accuracy (%)
C ₁	MIX	54
C ₂	AMI	81
C ₃	IMI	43
C ₄	Norm	73
C ₅	BVH	58
C ₆	RVH	55
C ₇	Norm	78
C ₈	LVH	93

The next step consists in labeling each cluster with the dominant diagnostic class, that is, using the majority rule for labeling them. For example, Cluster 2 consists of 246 AMI, 50 MIX, and 7 IMI subjects, and consequently this is labeled as AMI. In this case the accuracy of this ‘label’ or ‘classification’ is 81%. Table 39.3 reports the label of all the eight clusters with this rule and the corresponding accuracy. It is possible to verify that two clusters have an accuracy higher than 80% (C₂ and C₈), and two additional higher than 70% (C₄ and C₇).

A third possibility of interpreting the results is given by viewing the clusters with a different information granule, considering the tree the diagnostic groups: Norm, VH, and MI. In this way,

Class A: Norm

Class B: LVH + RVH + BVH

Class C: IMI + AMI + MIX.

In a similar way, we can group homogeneous clusters with the new extended classes:

Regions C₅, C₆, and C₇ are connected mainly with the diagnostic classes LVH, RVH, or BVH, and in this case, 93% of the subjects are correctly classified as VH.

Region C₁, C₂, and C₃ are connected mainly with MI, and considering them as an extended group, 99% of subjects are correctly classified as MI.

Regions C₄ and C₇ are connected with healthy people, with a correct classification of 75%.

The classification table with the new groups of clusters and the new groups of diagnostic classes is reported in Table 39.4.

With this ‘rough’ information granule, a total accuracy of 91.3% has been obtained.

Observing the SOM of Figure 39.4, it is possible to observe a spatial relationship between the three groups of clusters. In fact, C₄ and C₇ are spatially connected. In the same way, C₅, C₆, and C₈ are spatially connected. On the other hand, the regions C₁, C₂, and C₃ are on the border of the SOM. The use of a higher level of information granule has permitted a meaningful description of significant dependencies in experimental data, increasing the capability of the classifier.

Table 39.4 The simplified classification with three clusters and three classes

Group of clusters	Group of classes		
	Class A – Norm	Class B – LV	Class C – MI
C ₄ + C ₇	202	67	1
C ₅ + C ₆ + C ₈	6	142	4
C ₁ + C ₂ + C ₃	1	2	509

39.4.2 The RBF Model

In the RBF model, the descriptive block has been developed with a fuzzy preprocessing phase for characterizing the input patterns in terms of a set of linguistic variables. In this way, a data abstraction step is carried out, and a linguistic description of the input patterns has been performed, representing each input pattern by means of a set of membership functions related to the output classes, using Gaussian functions or radial basis functions. This input RBF layer will feed a normal feedforward neural network completely connected, which represents the classification block [25, 26]. The RBF preprocessing block represents the descriptive modeling, whereas the subsequent connectionist block represents the classification block. In this case the two descriptive and predictive models are performed and developed in the same framework, and the entire structure represents an example of supervised learning technique.

For each input parameter X_j , a number of m radial basis functions units ('rbf' nodes) are introduced with the following activation function:

$$RBF^h(x_j) = \frac{1}{2\sigma_j^h} \exp -\frac{(x_j - \mu_j^h)^2}{(2\sigma_j^h)^2},$$

where μ_j^h and σ_j^h are the central value and the dispersion factor of the bell-shaped functions. The initial values of these parameters can be derived from the statistics of the input features, considering the knowledge of the whole learning set. For example, the estimated mean and standard deviation of input parameter x_j in the diagnostic class h can be utilized for this purpose.

In this way the distribution of input parameters in the different diagnostic classes reveals an important rule for the accuracy of the entire classifier. Different alternatives have been tested for the choice of the number of m RBF, ranging from $m = 1$ to $m = p$, the number of diagnostic classes. The subsequent block is performed by a feedforward, fully connected neural network structure, in which the activation function are sigmoidal units ('s' nodes) and which analyze the linguistic description of the input space.

The resulting architecture is then considered as in the same connectionist framework and the back-propagation algorithm is appropriately modified for the training of both the RBF and the sigmoidal parts. The RMS (root mean squared) error is chosen as the error function. Two independent learning rates μ_1 and μ_2 are defined to update, respectively, the network weights and the RBF parameters along the descent gradient direction; that is,

$$\begin{aligned}\Delta w_{ij} &= -\eta_1 * \delta(RMS)/\delta w, \\ \Delta \mu_i &= -\eta_2 * \delta(RMS)/\delta \mu_i, \\ \Delta \sigma_i &= -\eta_2 * \delta(RMS)/\delta \sigma_i.\end{aligned}$$

The two learning rates are adaptive in order to cope with the problem of local minima.

For every input parameter x_j , a number \mathbf{m} of RBF units are defined and used with a specific activation function. It is clear that the number m of RBF units per input parameter X_j and the definition of the corresponding \mathbf{m} input clusters influence the performance of the classification system. We can suppose that a very fine granulation of the input space would lead to the more accurate description/results. On the other hand, with appropriate choice and representation of the different diagnostic classes, a classification system with a lower number of RBF units, that is, with a rough granulation and consequently with a less complex architecture, can reach comparable results.

In order to test this aspect, two strategies are adopted: rough granulation and fine granulation.

Rough granulation. In the first instance, it is possible to describe each input parameter with a 'rough granulation' by means of only one RBF unit (see Figure 39.5). The choice of the appropriate RBF parameters is a critical point. In this case, the statistics of the normal diagnostic class from the learning set has been considered for the initial shape of the Gaussian functions. This option is supported by the fact that the pathological classes can be described as a light or strong deviation from the input parameters in the normal class.

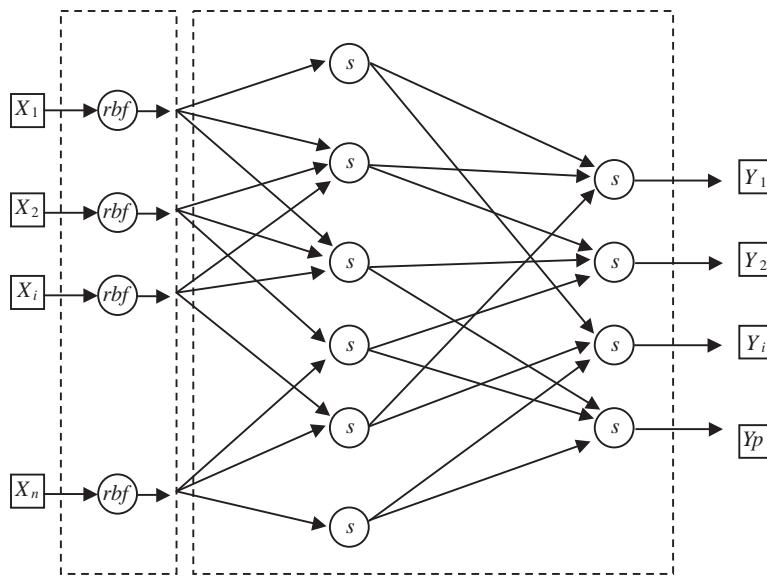


Figure 39.5 RBF preprocessing with a rough granulation

A sufficiently accurate description of this choice can be determined by using μ_j^h centered on the mean value of the normal class and with variance σ_j^h defined as a multiple of the variance of the normal class distribution.

Fine granulation. A description of the input space can be obtained by connecting each input node with p RBF units, where p is the number of diagnostic classes considered (see Figure 39.6). The initial

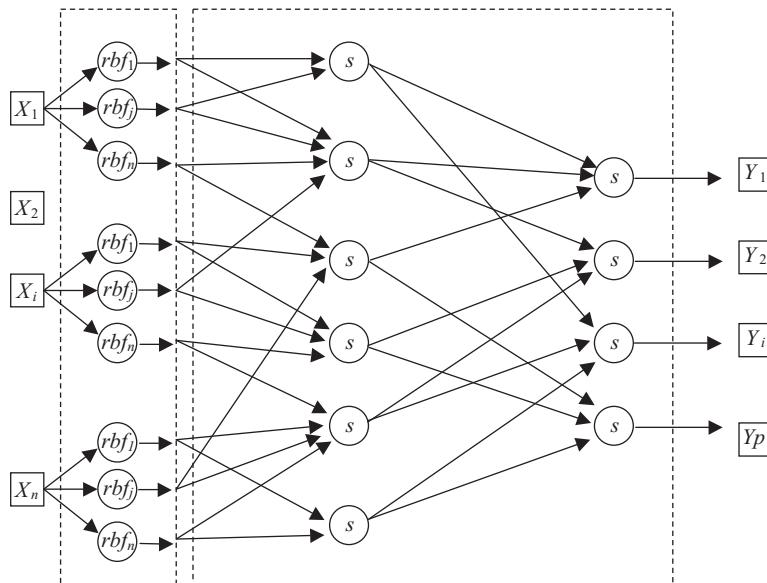


Figure 39.6 RBF preprocessing with a fine granulation

Table 39.5 Mean sensitivity and mean specificity of the RBF-preprocessing architecture with ‘fine’ or ‘rough’ granulation with two initialization strategies, 1*SD and 2*SD, and two classification strategies, R1 and R2

Methods	Training strategies							
	1*SD		2*SD		R1		R2	
	SE (%)	SP (%)	SE (%)	SP (%)	SE (%)	SP (%)	SE (%)	SP (%)
Rough granulation	65	94	61	94	60	93	65	94
Fine granulation	67	94	66	94	66	93	66	94

values of the two characteristic parameters of the RBF units are tuned considering the statistics of the entire learning set. The learning set is clustered according to the known diagnostic classification, and the mean and standard deviation of every feature are considered for determining the initial values of the μ_j^h and σ_j^h parameters. The ECG-CORDA database was used, considering the learning set of 2446 patients and 820 patients in the testing set. The architecture is characterized by 39 input nodes, as many input parameters and by 7 output nodes, as many diagnostic classes. The descriptive layer is composed by seven RBF units in case of rough granulation and by 273 RBF units in the fine granulation. The output nodes assume values in the range $[-1, +1]$, where the positive value corresponds to the presence of the class, whereas the negative one to its absence. The evaluation process has been performed considering the error rate in the classification of the various diagnostic classes. The RBF parameters are dynamically modified and optimized for increasing the speed of convergence. Several experiments have been performed in order to reach the optimal conditions of the classification strategies on the global performance and in order to investigate the robustness of the two architectures. In particular, two strategies for determining the initial shape of the RBF units have been tested: the initial values of σ_j^h parameters are set to once (1*SD) and twice (2*SD) the standard deviation of the normal class distribution along each dimension in the training set. Although the selected sets of the ECG-CORDA database consist of records with single diagnoses, in real cases there is the necessity to cope with multiple diagnoses. For this reason, two classification strategies have been tested: (R1) all the positive output nodes or the highest negative one are considered as valid classes and (R2) the highest output node is selected (max rule).

Table 39.5 reports the results obtained with this architecture in the test set. From this table we can see that the ‘fine granulation’ obtains more robust and stable results, a mean sensitivity of 66–67% and a mean specificity 93–94% in all the considered experiments. This is a satisfactory accuracy if compared to the previous works reported in Table 39.1. The rough granulation presents comparable and slightly lower results, with a higher variability. This architecture shows a mean sensitivity which is more dependent on the particular chosen training strategy.

The main advantage of the two architectures is the possibility to improve the transparency of the classification task, and this point is a crucial aspect in medical informatics.

39.4.3 Linguistic Preprocessing

A further support to human experts to make correct decisions is represented by a specific data abstraction performed by a linguistic description of the input parameters (with a fuzzy preprocessing phase). In this case the data abstraction is performed in the domain knowledge, and the resulting architecture offers the possibility of obtaining a linguistic justification.

Each input parameter is represented by means of a number of linguistic terms or membership functions with a more precise meaning and connection with the diagnostic classification. For example, a family of fuzzy sets {Low, Medium, High} may be adopted to describe the degree of membership of each input

parameter with respect to a certain diagnostic class. In this model the features are extracted or selected by means of the knowledge domain and the subsequent classification task may be more specific [27].

The first level of data abstraction performs a compatibility measure with the linguistic concepts A_i (Low_i , $Medium_i$, $High_i$)

The initial shapes of the fuzzy sets A_i are derived from the distribution information of the features in the learning set.

This input layer will feed a feedforward neural network completely connected, which performs the ‘predictive’ or ‘classification task,’ developing an example of supervised learning.

The entire architecture represents an adaptive network, in which all the membership functions are adapted in the learning phase, in addition to the weights of the connectionist approach.

For validating and testing this approach, the ECG-UCL database has been used, considering 404 records for the learning set and 135 records for the test set. Patients with normal QRS duration and no conduction defects were selected for this study. Four diagnostic classes have been considered: anterior (AMI), inferior (IMI), and combined (MIX) myocardial infarction, and a composite class ‘others,’ which includes normal, left, and right ventricular hypertrophy. The features are selected by means of the knowledge domain and the classification task. In particular, some features describing the ventricular electrical activity, altered by MI, are selected. The following set of eight parameters has been considered from each of the 12 ECG leads:

QRS amplitude and duration
 Q amplitude and duration
 R amplitude and duration
 T amplitude
 Q/R ratio

for a total of 96 morphologic features.

The membership functions can be drawn from the statistical information of the learning set. For example, the distribution of the values of QRS duration distribution in lead I (feature number 7) is reported in Figure 39.7. From this histogram it is possible to extract the fuzzy sets Low_7 , $Medium_7$, and $High_7$. These fuzzy sets can have piecewise linear membership functions (see Figure 39.8) and this option is chosen in order to simplify the dynamical adjustment of the shapes during the learning procedure.

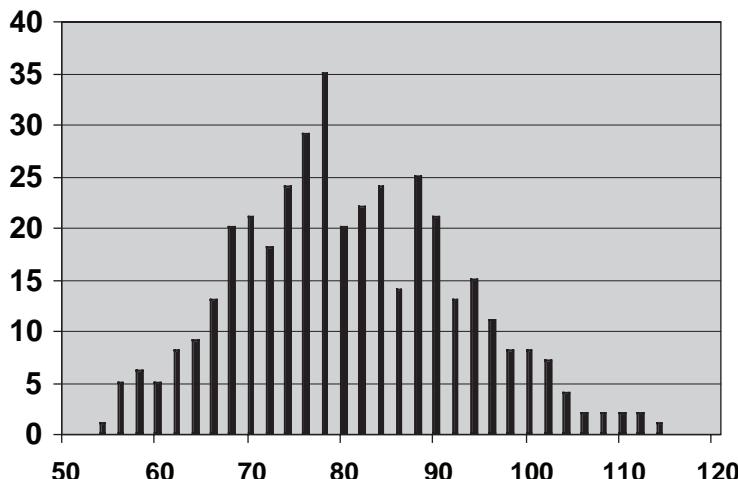


Figure 39.7 Distribution of QRS duration (ms) in Lead I in the learning set

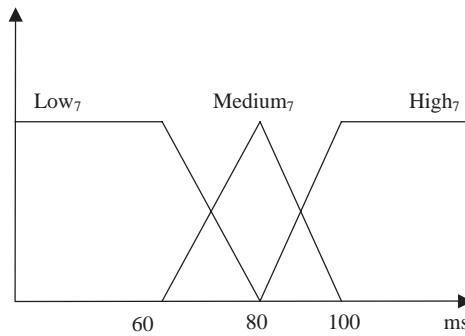


Figure 39.8 Linguistic terms low, medium, high for the feature QRS duration of Figure 39.7

These nodes perform an abstraction of the crisp measurements, by means of the level of a compatibility measure with the linguistic concepts A_i in $\{Low_i, Medium_i, High_i\}$.

The resulting architecture is reported in Figure 39.9, which is characterized by 96 input nodes and 288 linguistic nodes. Like in the previous architecture, the evaluation process has been performed considering the error rate in the classification of the four diagnostic classes. This hybrid neural architecture was trained with set of 404 ECG records. The trained neural architecture shows a total accuracy of 82.2% and a partial accuracy of 94.5% in the test set. The sensitivity and specificity of the four diagnostic classes are reported in Table 39.6. This architecture has a good classification property for IMI (94.5% specificity and 88.0% sensitivity). The composite class ‘others’ has a good accuracy. In addition, AMI and MIX show low sensitivities, and this kind of performance agrees with previous studies with standard neural network architecture and with classical classifier [20, 28].

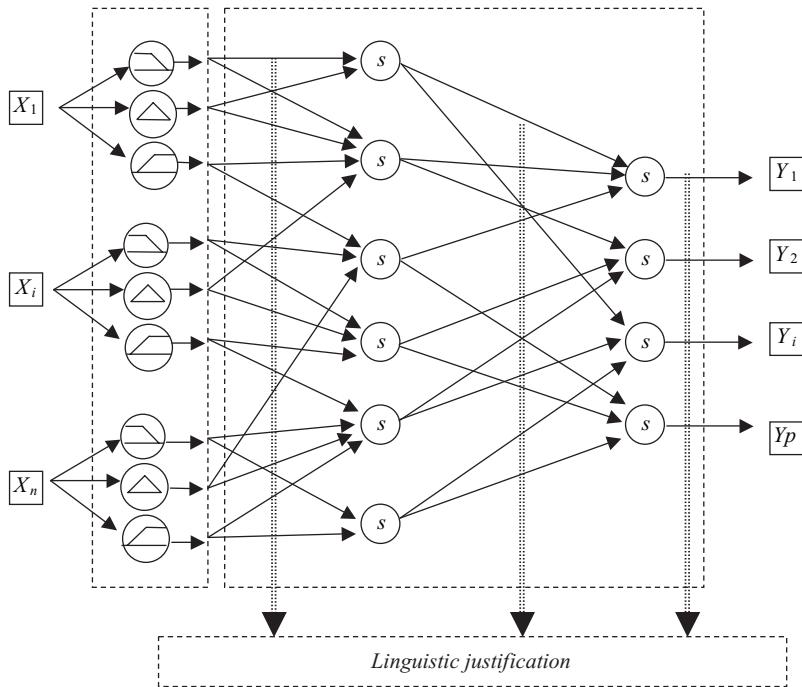


Figure 39.9 The architecture of linguistic processing with the possibility of a linguistic justification

Table 39.6 Sensitivity and specificity of AMI, IMI, MIX, and others with the architecture of linguistic preprocessing

Class	Specificity (%)	sensitivity (%)
AMI	92.7	72.0
IMI	94.5	88.0
MIX	95.4	60.0
Others	93.3	93.3

In addition, this architecture has been used for a simplified linguistic justification phase (Figure 39.9). A specific procedure analyzes the state of the network, considering inputs, outputs, and internal weights of the entire network in order to characterize the most relevant inputs for a specific classification [27]. Once obtained this information, their symbolic meaning are recovered and a rule of the following form is obtained:

IF (MF₁ is LL₁) **AND**
 (MF₂ is LL₂) **AND** ...
THEN (Class J is CF)

where (MF_i is LL_i) are the couples (morphologic_feature – linguistic label) which characterize the selected inputs, and (class j is CF) characterizes the jth output diagnostic class by the certainty factor CF. In this way, the ECG record under examination feeds the network, producing a diagnostic classification, and the state of the network permits the possibility to extract a linguistic justification of the specific classification. Considering the linguistic justification with highest certainty factors (CF = certain), we can consider the set of couples (MF_i is LL_i) as a linguistic description that caused/produced the diagnostic classification.

An example of linguistic justification from a subject in the test set classified as IMI is the following (considering the five most significant sentences):

(R duration) in V₂ is medium
 (T amplitude) in III is low
 (R amplitude) in AVR is medium
 (Q duration) in aVF is high
 (Q duration) in III is high

and an example of AMI justification is the following:

(R duration) in aVF is medium
 (Q duration) in II is low
 (R duration) in aVR is low
 (QRS duration) in aVR is low
 (R duration) in V1 is high

Then, the architecture with a linguistic description of input features possess either a good classification ability or a simplified linguistic justification, which can support the human expert.

39.5 Conclusion

In this chapter, different levels of data abstraction in the process of diagnostic classification in medical field have been studied. These levels were analyzed and discussed in the framework of information granular computing, and the following two main blocks were described in detail: the descriptive model and the predictive model. Three representative examples of information granulation have been considered: the

SOM model, the RBF model, and the linguistic model. A real complex problem of diagnostic classification was considered for testing and validating the proposed approaches: the computerized ECG classification with two validated databases. The description, the analysis, and the discussion of the three methods have shown that the individualization of different abstraction levels and the use of information granule in different way have improved the generalization capability, the accuracy, and the transparency of the classification task.

References

- [1] L.A. Zadeh. Toward a generalized theory of uncertainty (GTU) – an outline. *Inf. Sci.* 172 (2005) 1–40.
- [2] A.F. Gobi and W. Pedrycz. The potential of fuzzy neural networks in the realization of approximate reasoning engines. *Fuzzy Sets Syst.* 157 (2006) 2954–2973.
- [3] E.T. Kim, M.K. Park, S.H. Ji, and M. Park. A new approach to fuzzy modeling. *IEEE Trans. Fuzzy Sets* 5 (1997) 328–337.
- [4] W. Pedrycz and F. Gomide. *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press, Cambridge, MA, 1998.
- [5] W. Pedrycz and V. Vasilakos. Linguistic models and linguistic modeling. *IEEE Trans. Syst. Man Cybern.* 29 (1999) 745–757.
- [6] K. Cios, W. Pedrycz, and R. Swiniarski. *Data Mining Techniques*. Kluwer Academic Publisher, Boston, MA, 1998.
- [7] T.Y. Lin. Introduction to special issues on data mining and granular computing (Editorial). *Int. J. Approx. Reason.* 40 (2005) 1–2.
- [8] P. Piatetsky-Shapiro and W. Frawley (eds). *Knowledge Discovery in Databases*. MIT/AAAI Press, Cambridge, MA, 1991.
- [9] W. Pedrycz and K.C. Kwak. Boosting of granular models. *Fuzzy Sets Syst.* 157 (2006) 2934–2953.
- [10] L.A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.
- [11] G. Bortolan and W. Pedrycz. Hyperbox classifier for arrhythmia classification. *Kybernetes* 36 (2007) 531–547.
- [12] L.A. Zadeh. Fuzzy sets and information granularity. In: M. Gupta, R. Ragade, and R. Yager (eds), *Advances in Fuzzy Set Theory and Applications*, North-Holland, Amsterdam, 1979, pp. 3–18.
- [13] M. Dash and H. Liu. Feature selection for classification. *Intell. Data Anal.* 1 (1997) 131–156.
- [14] R. Jensen and Q. Shen. Fuzzy-rough data reduction with ant colony optimization. *Fuzzy Sets Syst.* 149 (2005) 5–20.
- [15] I. Koch and K. Naito. Dimension selection for feature selection and dimension reduction with principal and independent component analysis. *Neural Comput.* 19 (2007) 513–545.
- [16] H.L. Wei and S.A. Billings. Feature subset selection and ranking for data dimensionality reduction. *IEEE Trans Pattern Anal Mach Intell* 29 (2007) 162–166.
- [17] C.R. Brohet, C. Derwael, A. Robert, and R. Fesler. Methodology of ECG interpretation in the Louvain program. *Method Inf. Med.* 29 (1990) 403–409.
- [18] R. Degani and G. Bortolan. Methodology of ECG interpretation in the Padova Program. *Method Inf. Med.* 29 (1990) 386–392.
- [19] G. Bortolan, C. Brohet, and S. Fusaro. Possibilities of using neural networks for diagnostic ECG classification. *J. Electrocardiol.* 29 (1996) 10–16.
- [20] G. Bortolan and J.L. Willemans. Diagnostic ECG classification based on neural networks. *J. Electrocardiol.* 26 (1994) 75–79.
- [21] J.L. Willemans, E. Lesaffre, and J. Pardaens. Comparison of the classification ability of the electrocardiogram and vectorcardiogram. *Am. J. Cardiol.* 59 (1987) 119–124.
- [22] T. Kohonen. *Self Organization and Associative Memory*. Springer-Verlag, Berlin, 1989.
- [23] T. Kohonen. The self-organizing map. *Proc. IEEE* 78 (1990) 1464–1480.
- [24] G. Bortolan and W. Pedrycz. An interactive framework for an analysis of ECG signals. *Artif. Intell. Med.* 24 (2002) 109–132.
- [25] R. Silipo, G. Bortolan, and C. Marchesi. Supervised and unsupervised learning for diagnostic ECG classification. In: *18th Annual International Conference of the IEEE-EMBS*. IEEE Publishing Services. New York, Amsterdam, The Netherlands, 1996. (CD-ROM)

- [26] R. Silipo, G. Bortolan, and C. Marchesi. Design of hybrid architectures based on neural classifier and RBF pre-processing for ECG analysis. *Int. J. Approx. Reason.* 21 (1999) 177–196.
- [27] P. Bozzola, G. Bortolan, C. Combi, F. Pinciroli, and C. Brohet. A hybrid neuro-fuzzy system for ECG classification of myocardial infarction. In: A. Murray and R. Arzbaecher (eds), *Computers in Cardiology 96*. IEEE Computer Society, Los Alamitos, CA, 1996, pp. 241–244.
- [28] J.L. Willems, E. Lesaffre, J. Pardaens, and D. De Schreye. Multigroup logistic classification of the standard 12-lead and 3-lead ECG. In: J.L. Willems, J.H. van Bemmel, and C. Zywietz (eds), *Computer ECG Analysis: Towards Standardization*, North-Holland, Amsterdam, 1986, pp. 203–210.

40

Eigen Fuzzy Sets and Image Information Retrieval

Ferdinando Di Martino, Salvatore Sessa, and Hajime Nobuhara

40.1 Introduction

It is natural to interpret a monochromatic image of size $n \times n$ (pixels) to be a fuzzy relation R whose entries $R(x, y)$ are obtained by normalizing the intensity $P(x, y)$ of each pixel with respect to (for short, w.r.t.) the length L of the scale; that is, $R(x, y) = P(x, y)/L$. In literature, the usage of fuzzy relation calculus to image processing is well known: see, e.g., [1, 2] for applications to pattern recognition, [3] for image restoration, and [4–6] for compression/decompression image procedures. Here we use the greatest eigen fuzzy set (for short, GEFS) A of R w.r.t. the max–min composition [7–12] and the smallest eigen fuzzy set (for short, SEFS) B of R w.r.t. the min–max composition for applications to problems of image information retrieval. The membership functions of GEFS and SEFS contain values of the assigned fuzzy relation and the pair (A, B) is considered as information granule of R [13–15]. Indeed, we find the information granules of the original image R and of the retrieved images. Based on these pairs, a similarity measure is also introduced in order to compare R with the retrieved images. The tests are made on the images extracted from ‘View Sphere Database’ (<http://www.prima.inrialpes.fr>), in which an object is photographed from various directions by using a camera placed on a semisphere whose center is the object itself.

40.2 Eigen Fuzzy Sets

Let R be a fuzzy relation defined on a referential set X and A be a fuzzy set of X , that is, $R \in F(X \times X) = \{S: X \times X \rightarrow [0, 1]\}$ and $A \in F(X) = \{B: X \rightarrow [0, 1]\}$, such that

$$A = R \circ A, \quad (1)$$

where ‘ \circ ’ stands for the well-known max–min composition. In terms of membership functions, the equation (1) is read as

$$A(y) = \max_{x \in X} \{\min(A(x), R(x, y))\} \quad (2)$$

for all $x, y \in X$ and A is defined to be an eigen fuzzy set of R w.r.t. the max–min composition. Let $A_i \in F(X)$, $i = 1, 2, \dots$, be defined iteratively by

$$A_1(z) = \max_{x \in X} R(x, z) \quad \text{for all } z \in X, \quad A_2 = R \circ A_1, \dots, A_{n+1} = R \circ A_n, \dots \quad (3)$$

It is known in literature [7, 11, 12] that there exists an integer $p \in \{1, \dots, \text{card } X\}$, such that $A_p = R \circ A_p = A_{p+1}$ is the GEFS of R w.r.t. the max–min composition. In the sequel we also consider the dual equation of (2), that is, the following equation:

$$A = R \bullet A, \quad (4)$$

where ‘•’ denotes the min–max composition. In terms of membership functions, the equation (4) is read as

$$A(y) = \min_{x \in X} \{\max(A(x), R(x, y))\} \quad (5)$$

for all $x, y \in X$ and A is also said to be an eigen fuzzy set of R w.r.t. the min–max composition. Let $B_i \in F(X)$, $i = 1, 2, \dots$, be defined iteratively by

$$B_1(z) = \min_{x \in X} R(x, z) \quad \text{for all } z \in X, \quad B_2 = R \bullet B_1, \dots, B_{n+1} = R \bullet B_n, \dots \quad (6)$$

Similarly it can be easily proved that there exists some $q \in \{1, \dots, \text{card } X\}$ such that $B_q = R \bullet B_q = B_{q+1}$ is the SEFS of R w.r.t. the min–max composition (5). For example, we consider the following fuzzy relation for illustrating the above concepts, and for simplicity of presentation, we assume $\text{card } X = 6$:

$$R = \begin{pmatrix} 0.6 & 0.2 & 0.5 & 0.7 & 1.0 & 0.9 \\ 0.8 & 0.1 & 0.3 & 0.3 & 0.6 & 0.7 \\ 0.7 & 0.4 & 0.4 & 0.6 & 0.9 & 0.5 \\ 0.5 & 0.6 & 0.2 & 0.5 & 0.8 & 0.7 \\ 0.4 & 0.5 & 0.4 & 0.3 & 0.7 & 0.4 \\ 0.3 & 0.2 & 0.3 & 0.4 & 0.6 & 0.2 \end{pmatrix}.$$

By using the iterations (3), we have that $A_1 = (0.8, 0.6, 0.5, 0.7, 1.0, 0.9)$, $A_2 = R \circ A_1 = (0.6, 0.6, 0.5, 0.7, 0.8, 0.8)$, and $A_3 = R \circ A_2 = (0.6, 0.6, 0.5, 0.6, 0.7, 0.6)$; hence, $A_4 = R \circ A_3 = A_3$; that is, A_3 is the GEFS of R w.r.t. the max–min composition (2). Now with the iterations (6), we get $B_1 = (0.3, 0.1, 0.2, 0.3, 0.6, 0.2)$, $B_2 = R \bullet B_1 = (0.3, 0.1, 0.3, 0.3, 0.6, 0.2)$, and $B_3 = R \bullet B_2 = B_2$; that is, B_2 is the SEFS of R w.r.t. the min–max composition (5).

In our tests, the GEFS A_p w.r.t. the max–min composition (2) is calculated by using an algorithm, based on formulas (3), which finds an integer $p \in \{1, \dots, \text{card } X\}$, such that $A_p = R \circ A_p = A_{p+1}$ holds. For further details about an optimization of this algorithm we remind the interested reader to the specific literature (see, e.g., [7, 12]). In similar way, another algorithm, based on formulas (6), finds an integer $q \in \{1, \dots, \text{card } X\}$, such that the equality $B_q = R \bullet B_q = B_{q+1}$ is satisfied and thus the SEFS B_q of R w.r.t. the min–max composition (5) is calculated as well.

40.3 A Granular View of the Images

Let X be our referential set and $Y \subseteq F(X \times X)$ be a finite set of color images of sizes (pixels) $N \times N$, $N = \text{card } X$, in the RGB space. In each of the three bands of this space we calculate the information granules of an image. More generally, if we consider a multiband space with M bands, we define the family of

information granules $G(R_i) = \{(A_{i1}, B_{i1}), \dots, (A_{iM}, B_{iM})\}$, where $A_{ik} \in F(X)$ (resp., $B_{ik} \in F(X)$) is the GEFS (resp., SEFS) w.r.t. the max-min (resp., min-max) composition of the image R_i in the k th band, $k = 1, \dots, M$. Strictly speaking, we can define a multivalued mapping $G: R_i \in Y \rightarrow ((A_{i1}, B_{i1}), \dots, (A_{iM}, B_{iM})) \in (F(X) \times F(X))^M$, and based on these pairs, we now define a similarity operator between two images R_i and R_j , $i, j \in \{1, 2, \dots, \text{card } Y\}$ by setting

$$d(R_i, R_j) = \frac{1}{M} \sum_{k=1}^M \sum_{x \in X} (A_{ik}(x) - A_{jk}(x))^2 + (B_{ik}(x) - B_{jk}(x))^2. \quad (7)$$

For a complete comparison between two images $R_i, R_j \in F(X) \times F(X)$ of sizes $N \times N$, $N = \text{card } X$, in M bands we have to use a ‘fine-grained’ view in dimension $N \times N \times M$. Indeed, by using the information granules $(A_{i1}, B_{i1}), \dots, (A_{iM}, B_{iM})$ in the formula (7), we can derive, for image retrieval application, a similarity information in a ‘coarser-grained’ view in dimension $2N \times M$, of course N being the dimension of both GEFS A_{ik} and SEFS B_{ik} , $k = 1, \dots, M$, in a monocromatic image. In the next section are described the results of our experiments by using these information granules in order to find that retrieved image which minimizes the similarity measure (7) with respect to a sample image. Other similarity measures used in image processing can be found in [1] (for further details, see e.g., [16]).

40.4 Image Retrieval

We have used in our experiments the color test images of size 256×256 (pixels) extracted from View Sphere Database concerning two objects: an eraser and a pen. The object is considered to be the center of a semisphere on which a camera is placed in 41 different directions, each determined from two angles θ ($0^\circ < \theta < 90^\circ$) and F ($-180^\circ < \phi < 180^\circ$) as illustrated in Figure 40.1. The camera establishes an image of the object for each direction, which can be identified from the two above-mentioned angles. An image R_1 (with prefixed angles $\theta = 11^\circ$, $F = 36^\circ$ for the eraser, and $\theta = 10^\circ$, $F = 54^\circ$ for the pen) is assumed as sample image and it must be compared with another image R_2 chosen among the remaining

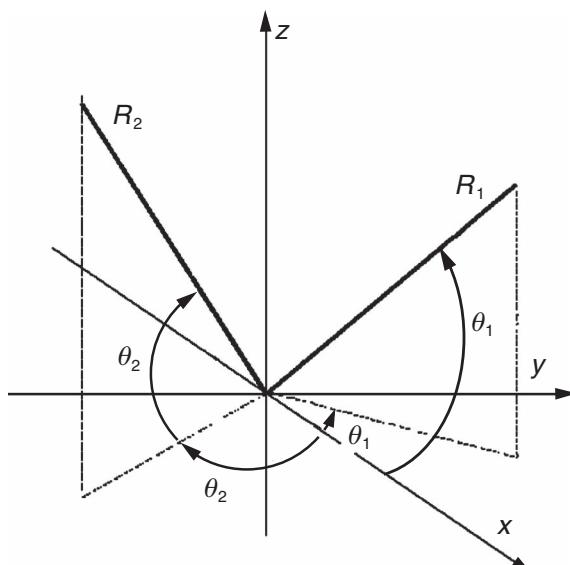


Figure 40.1 The angles for R_1 and R_2 . (The object is in the origin.)

Table 40.1 Forty tests on the eraser at $\theta = 11^\circ$ and $\phi = 36^\circ$

θ	ϕ	$d_R(R_1, R_2)$	$d_G(R_1, R_2)$	$d_B(R_1, R_2)$	$d(R_1, R_2)$
10	54	7.2543	20.4322	15.1914	14.2926
11	-36	18.4459	30.1343	25.7560	24.7787
25	37	16.4410	35.3923	24.2910	25.3748
10	89	18.7165	32.1656	25.8345	25.5722
10	-54	17.3107	34.4895	25.8311	25.8771
25	-37	18.6635	40.0964	29.4798	29.4132
10	-89	18.4712	39.5442	31.3288	29.7814
53	-09	20.8855	39.8762	28.7621	29.8412
49	08	20.0679	39.3050	31.6754	30.3494
19	72	26.4956	34.0048	33.2890	31.2631
49	-08	23.8688	41.9027	31.8200	32.5305
53	09	23.4420	42.9204	31.2708	32.5444
68	13	30.2911	40.5672	33.9494	34.9359
10	108	28.6132	41.8015	37.5753	35.9967
68	-13	30.3181	41.7512	36.2359	36.1017
11	108	28.7340	41.9619	38.5458	36.4139
05	63	26.7522	45.3335	39.7992	37.2950
19	-72	26.3711	45.9441	41.6426	37.9859
05	-63	29.2269	46.4113	39.8831	38.5071
05	81	33.7586	43.0861	41.3066	39.3837
10	-108	29.0343	54.5360	41.4635	41.6780
11	-108	30.6790	54.6601	41.3710	42.2367
81	107	22.8538	62.3589	41.9106	42.3744
05	-81	35.7017	45.6723	47.1439	42.8393
68	84	22.4891	64.5719	42.0165	43.0258
81	34	23.1417	68.3752	43.6347	45.0505
68	-84	24.8346	68.1872	43.0392	45.3537
81	-34	23.9752	70.6055	42.9990	45.8599
81	-107	23.9138	69.2505	45.2260	46.1301
68	59	25.2859	68.3162	46.2928	46.6316
68	131	27.9327	68.6098	48.8383	48.4602
68	-59	27.8207	70.9821	46.9860	48.5963
05	135	28.3233	82.2147	47.5461	52.6947
68	156	28.6810	84.7417	54.0792	55.8340
19	144	33.5144	86.9232	58.5220	59.6532
05	-135	36.3067	89.4515	61.5936	62.4506
19	-144	36.6148	92.8826	61.4792	63.6589
68	-131	35.9501	93.4193	64.3219	64.5637
68	-156	37.6012	96.4486	62.0581	65.3693

40 images (directions); that is, we suppose to have a set Y with 41 images. We use the formula (7) for defining a similarity measure between R_1 and R_2 , where $X = \{1, 2, \dots, 256\}$. Since we are faced with color images, we evaluate the above eigen fuzzy sets in the three components $R (= 1)$, $G (= 2)$, $B (= 3)$ of each image in the RGB space, for which we must assume $M = 3$ in the formula (7), which we rewrite as

$$d(R_1, R_2) = \frac{1}{3} [d_R(R_1, R_2) + d_G(R_1, R_2) + d_B(R_1, R_2)], \quad (8)$$

where

$$d_R(R_1, R_2) = \sum_{x=1}^{256} (A_{11}(x) - A_{21}(x))^2 + (B_{11}(x) - B_{21}(x))^2, \quad (9)$$

$$d_G(R_1, R_2) = \sum_{x=1}^{256} (A_{12}(x) - A_{22}(x))^2 + (B_{12}(x) - B_{22}(x))^2, \quad (10)$$

and

$$d_B(R_1, R_2) = \sum_{x=1}^{256} (A_{13}(x) - A_{23}(x))^2 + (B_{13}(x) - B_{23}(x))^2 \quad (11)$$

being (A_{i1}, B_{i1}) , (A_{i2}, B_{i2}) , and (A_{i3}, B_{i3}) the information granules of each image R_i , $i = 1, 2, \dots$, in the bands R, G, B, respectively. For scopes of image information retrieval, we are of course interested in the image R_2 of the object itself chosen among other 40 images available in the View Sphere Database (each identified with the two above-mentioned angles θ and ϕ), which makes minimum the similarity measure (8), that is, R_2 is such that $d(R_1, R_2) = \min\{d(R_1, R) : R \in Y\}$.

40.5 Results of Tests

Concerning the eraser, we consider the image obtained from the camera in the direction with angles $\theta = 11^\circ$ and $\phi = 36^\circ$ as sample image R_1 . The results are obtained by considering the remaining set of 40 images, of which we evaluate the quantities (9–11) and the global similarity measure (8), given in Table 40.1.

Generally speaking, the similarity measure (8) decreases for directions whose angles θ and ϕ have a small difference from those ones prefixed for the sample image represented in Figure 40.2 (left): the minimum value of $d(R_1, R_2)$ is obtained for the retrieved image of Figure 40.2 (right) obtained under the direction with angles $\theta = 10^\circ$ and $\phi = 54^\circ$. We also point out the shapes of the membership functions of the eigen fuzzy sets involved only in the band R of the RGB space since analogous situations happen in the bands G and B. Indeed, the GEFS (w.r.t. the max–min composition) of the fuzzy relations representing the sample and retrieved images of Figure 40.2, respectively, have very similar shapes as shown in Figures 40.3 and 40.4.

The SEFS (w.r.t. the min–max composition) of the fuzzy relations representing the sample and retrieved images of Figure 40.2, respectively, have slightly different shapes as shown in Figures 40.5 and 40.6.

In order to have a further confirmation of our approach, we have considered a second object, a pen, contained in the View Sphere Database whose sample image R_1 is obtained from the camera in the direction with angles $\theta = 10^\circ$ and $\phi = 54^\circ$. We also limit the problem to a data set of 40 test images of which we evaluate the quantities (9–11) and the global similarity measure (8), reported in Table 40.2.

Figure 40.7 (left) contains the sample image R_1 and Figure 40.7 (right) gives the retrieved image R_2 having minimum value $d(R_1, R_2)$, obtained in the direction with angles $\theta = 10^\circ$ and $\phi = 18^\circ$.

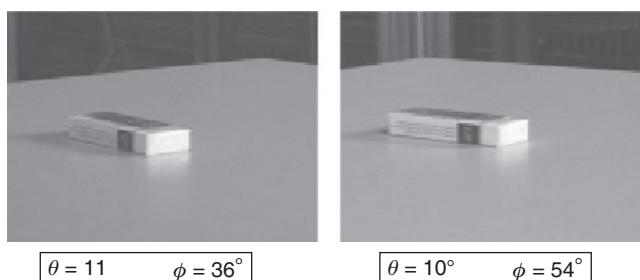


Figure 40.2 Eraser: Sample image (left) and retrieved image (right)

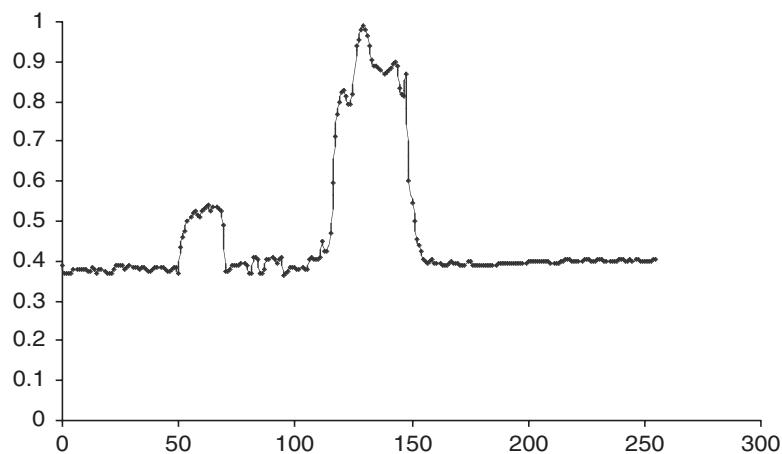


Figure 40.3 GEFS in the band R at $\theta = 11^\circ$ and $\phi = 36^\circ$

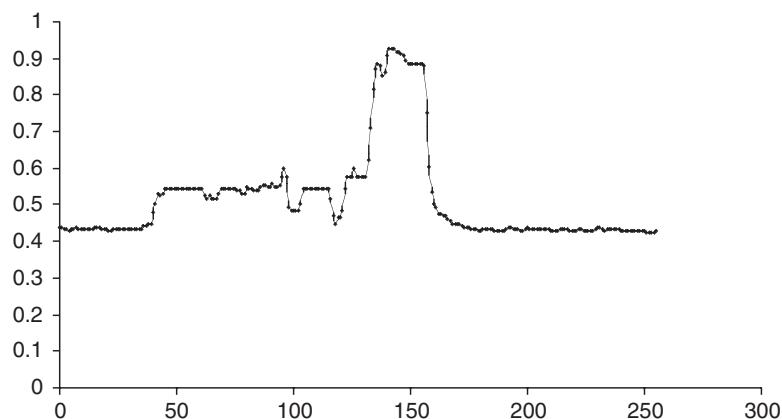


Figure 40.4 GEFS in the band R at $\theta = 10^\circ$ and $\phi = 54^\circ$

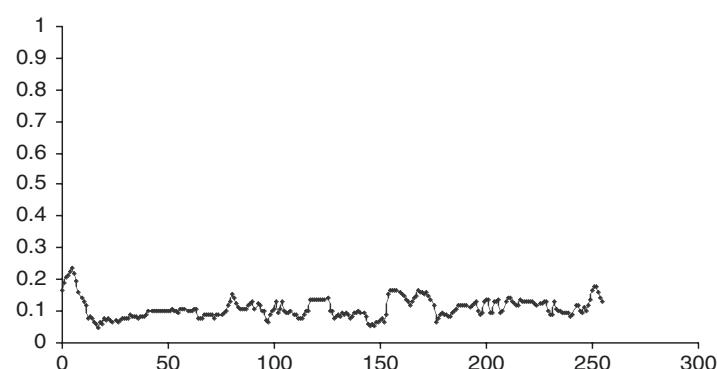


Figure 40.5 SEFS in the band R at $\theta = 11^\circ$ and $\phi = 36^\circ$

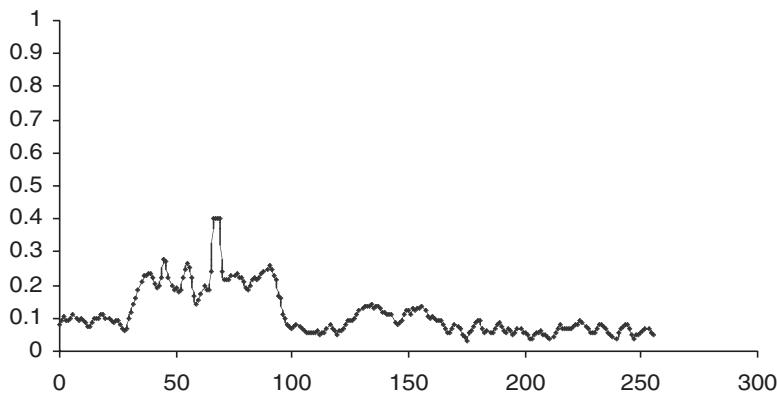


Figure 40.6 SEFS in the band R at $\theta = 10^\circ$ and $\phi = 54^\circ$

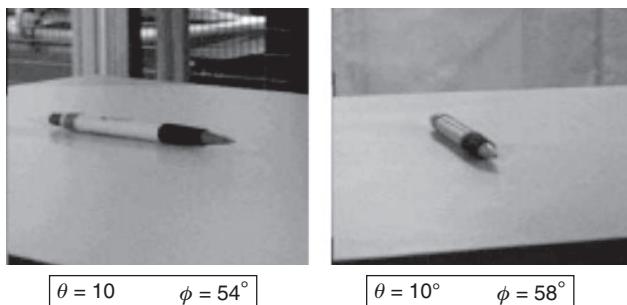


Figure 40.7 Pen: Sample image (left) and retrieved image (right)

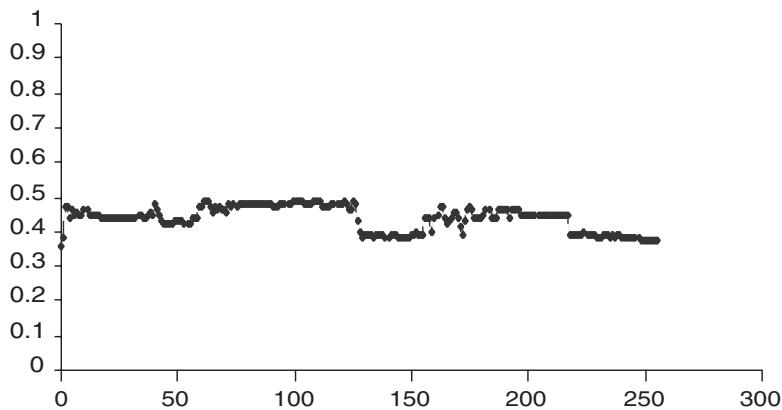


Figure 40.8 GEFS in the band R at $\theta = 10^\circ$ and $\phi = 54^\circ$

As in the first experiment, we also point out the shapes of the membership functions of the eigen fuzzy sets involved only in the band R of the RGB space. Figures 40.8 and 40.9 show the GEFS w.r.t. the max-min composition of the fuzzy relations representing the sample and retrieved image of Figure 40.7, respectively. We also note here slightly different shapes of the membership functions. Same conclusion holds for both SEFS, given in Figures 40.10 and 40.11, w.r.t. the min-max composition of the fuzzy relations related to the images of Figure 40.7, respectively.

Table 40.2 Forty tests on the pen at $\theta = 10^\circ$ and $\phi = 54^\circ$

θ	ϕ	$d_R(R_1, R_2)$	$d_G(R_1, R_2)$	$d_B(R_1, R_2)$	$d(R_1, R_2)$
10	18	0.8064	0.4495	0.9232	0.7264
10	-18	1.2654	1.0980	7.2903	3.2179
68	84	2.7435	1.7468	5.8812	3.4572
11	36	2.1035	5.4634	3.3769	3.6479
10	-54	2.5394	2.0005	9.2138	4.5845
68	-84	5.8812	7.9886	4.2964	6.0554
27	-46	5.8044	3.9859	12.9321	7.5742
27	46	5.8013	4.2496	12.7533	7.6014
11	-36	5.6878	9.1868	8.2856	7.7201
36	25	6.1038	4.6583	13.4581	8.0734
19	72	6.3045	4.9884	13.1659	8.1529
19	-72	6.7958	6.3561	12.5927	8.5815
36	-25	7.2973	6.4115	12.6759	8.7949
53	37	8.9293	7.2379	10.8677	9.0116
53	-37	8.9293	7.2379	10.8677	9.0116
36	47	8.6518	6.7293	11.9654	9.1155
81	-34	9.1170	7.7926	12.1062	9.6719
36	-47	8.6518	7.6186	12.8390	9.7031
81	34	9.4044	7.3458	13.3615	10.0372
05	63	10.1318	7.4309	16.1506	11.2378
49	08	10.2599	7.1762	17.0513	11.4958
68	59	10.1766	7.8892	17.0273	11.6977
49	-08	10.6486	7.4322	17.5533	11.8780
68	-59	10.9755	7.6874	17.0933	11.9187
68	13	10.5710	8.0438	17.4920	12.0356
05	-63	11.7102	7.5643	17.0980	12.1242
68	-13	11.6390	8.9190	17.9446	12.8342
05	135	18.8210	13.7300	22.9570	18.5027
49	-137	18.8439	15.1612	21.6178	18.5410
49	137	18.3671	15.1954	22.1725	18.5783
53	108	19.1876	14.9012	21.9736	18.6875
53	-108	20.6757	17.0290	22.8642	20.1896
19	144	21.4607	18.9883	24.9176	21.7889
05	-135	21.3450	19.1930	24.8910	21.8097
19	-144	22.9764	19.6550	26.5277	23.0530
68	131	23.4155	20.9765	27.1092	23.8337
68	-131	25.6110	21.8400	28.9030	25.4513
81	180	26.1709	21.4363	30.8812	26.1628
68	156	26.7178	22.1099	30.1015	26.3097
68	-156	26.8981	21.4138	31.9070	26.7396

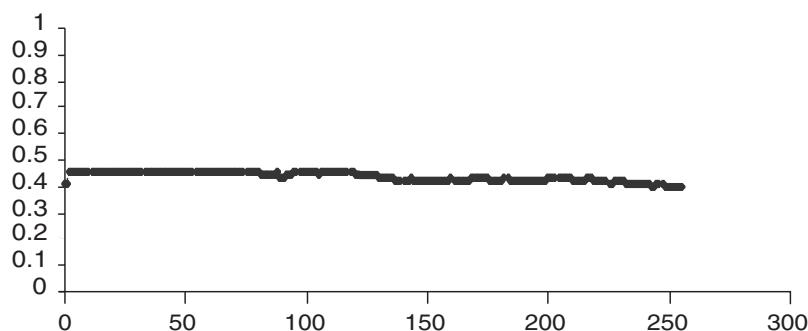


Figure 40.9 GEFS in the band R at $\theta = 10^\circ$ and $\phi = 18^\circ$

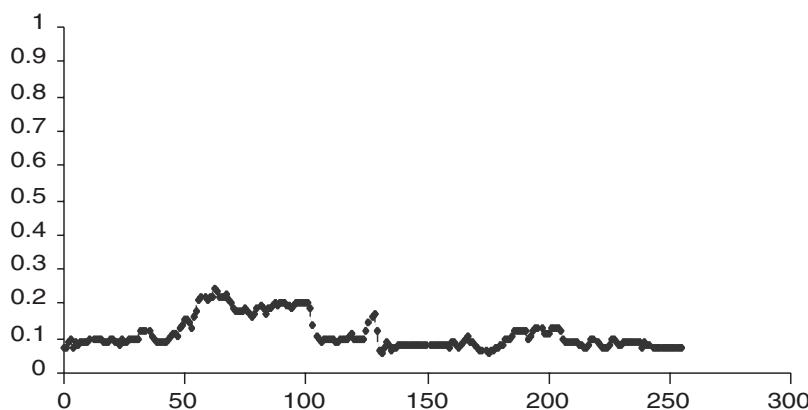


Figure 40.10 SEFS in the band R at $\theta = 10^\circ$ and $\phi = 54^\circ$

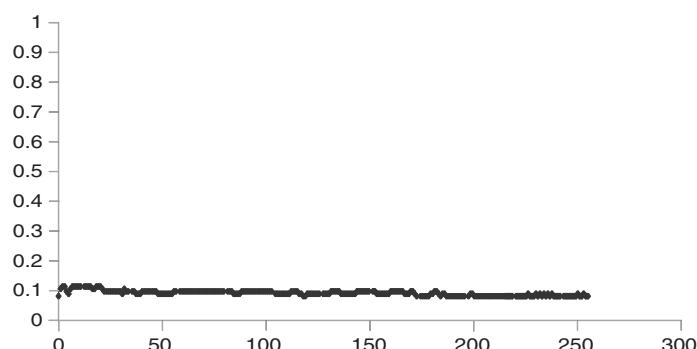


Figure 40.11 SEFS in the band R at $\theta = 10^\circ$ and $\phi = 18^\circ$

40.6 Conclusion

Two types of eigen fuzzy sets, that is, the GEFS (resp., SEFS) of a fuzzy relation w.r.t. the max–min (resp., min–max) composition, are proposed. Since an image can be interpreted as a fuzzy relation by normalizing the values of its pixels, we have proved that GEFS and SEFS of this relation are useful for image information retrieval problems. The GEFS and SEFS of a set of images in all bands are interpreted as information granules, on which a similarity measure is defined and used for comparing a sample image with other images to be retrieved. This comparison consists essentially in finding the retrieved image which minimizes the similarity measure and it is performed in two data sets of 40 test images extracted from ‘View Sphere Database’ concerning two objects: an eraser and a pen. We have executed the same above tests on data sets of images of other objects extracted from the mentioned database and not reported here for brevity of discussion. All the experiments have proved that GEFS and SEFS are good tools for retrieving image information.

Acknowledgment

The third author thanks the ‘Mizuho Foundation for the Promotion of Sciences’ for supporting this research.

References

- [1] J.C. Bezdek, J. Keller, R. Krisnapuram, and N.R. Pal. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.
- [2] I. Bloch and H. Maitre. Fuzzy mathematical morphologies: A comparative study. *Pattern Recognit.* 28 (1995) 1341–1387.
- [3] K. Arakawa. Fuzzy rule-based signal processing and its applications to image restoration. *Fuzzy Sets Syst.* 77 (1996) 3–13.
- [4] F. Di Martino, V. Loia, and S. Sessa. A method for coding/decoding images by using fuzzy relation equations. In: *Proceedings of IFSA 2003, Lecture Notes in Artificial Intelligence*, Vol. 2715. Springer, Berlin, Germany, 2003, pp. 436–441.
- [5] H. Nobuhara, K. Hirota, and W. Pedrycz. Fast solving method of fuzzy relational equations and its application to lossy image compression. *IEEE Trans. Fuzzy Syst.* 8 (3) (2000) 325–334.
- [6] S. Sessa, H. Nobuhara, W. Pedrycz, and K. Hirota. Two iterative methods of decomposition of a fuzzy relation for image compression and decompression processing. *Soft Comput. J.* 8 (2004) 698–704.
- [7] M.M. Bourke and D. Grant Fisher. Convergence, Eigen fuzzy sets and stability analysis of relation matrices. *Fuzzy Sets Syst.* 81 (1996) 227–234.
- [8] H. Nobuhara, and K. Hirota. Eigen fuzzy sets of various composition and their application to image analysis. In: *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, USA, IV–192, 2003.
- [9] H. Nobuhara and K. Hirota. A solution for eigen fuzzy sets of adjoint max-min composition and its application to image analysis. In: *Proceedings of IEEE International Symposium on Intelligent Signal Processing*, Budapest, Hungary, 2003, pp. 27–30.
- [10] H. Nobuhara, B. Bede, and K. Hirota. On various eigen fuzzy sets and their application to image reconstruction. *Inf. Sci.* 176 (20) (2006) 2988–3010.
- [11] E. Sanchez. Resolution of eigen fuzzy sets equations. *Fuzzy Sets Syst.* 1 (1978) 69–74.
- [12] E. Sanchez. Eigen fuzzy sets and fuzzy relations. *J. Math. Anal. Appl.* 81 (1981) 399–421.
- [13] A. Bargiela and W. Pedrycz. *Granular Computing*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.
- [14] Y.Y. Yao. Information granulation and rough set approximations. *Int. J. Intell. Syst.* 16 (2001) 87–104.
- [15] L.A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.
- [16] T.Y. Lin, Y.Y. Yao, and L.A. Zadeh (eds). *Data Mining, Rough Sets and Granular Computing*. Physica-Verlag, Heidelberg, Germany, 2002.

41

Rough Sets and Granular Computing in Dealing with Missing Attribute Values

Jerzy W. Grzymala-Busse

41.1 Introduction

Many real-life data sets are incomplete. In this chapter data sets are presented as *decision tables*, in which *cases (examples)* correspond to rows, while *attributes* and a *decision* correspond to columns. An example of such a table is presented in Table 41.1. In this decision table attributes are *age*, *weight*, and *gender*, and the decision is *strength*. The set of all cases with the same decision value is called a *concept*. In Table 41.1, case set $\{1, 2, 3, 4\}$ is a concept of all cases such that the value of *strength* is *small*. The data set from Table 41.1 is incomplete, since some attribute values, denoted by ‘?’ are missing.

In this chapter we will discuss mostly data mining methods handling missing attribute values. Note that in statistics dealing with missing attribute values is as important as it is in data mining. However, statisticians use frequently different methods, briefly described at the end of this section.

Some theoretical properties of data sets with missing attribute values were studied in [1–3].

We will categorize methods to handle missing attribute values as *sequential* and *parallel*. Sequential methods are applied before the main process of knowledge acquisition, e.g., rule induction or decision tree generation, while in parallel methods both processes are conducted at the same time, i.e., in parallel.

The most typical sequential methods handling missing attribute values include deleting cases with missing attribute values, replacing a missing attribute value by the most common value of that attribute, replacing a missing attribute value by the mean for numerical attributes, assigning all possible values to the missing attribute value and assigning to a missing attribute value the corresponding value taken from the closest fit case.

The parallel methods of handling missing attribute values include an approach based on rough set theory, exemplified by the MLEM2 algorithm, a modification of the LEM2 (Learning from Examples Module, version 2) rule induction algorithm in which rules are induced from the original incomplete data set. In this approach the user may make use of an additional information about the source of incompleteness. For example, the user may know whether missing attribute values were caused by incidental erasing or if these values were irrelevant to begin with; e.g., the attribute ‘Number of pregnancies’ is irrelevant for males. C4.5 [4] approach to missing attribute values is another example of a method from

Table 41.1 An incomplete decision table

Case	Attributes			Decision
	Age	Weight	Gender	
1	?	Light	Male	Small
2	Old	?	Female	Small
3	?	Light	?	Small
4	Old	Heavy	Female	Small
5	Medium	?	Male	Large
6	?	Heavy	Female	Large
7	Medium	?	Male	Large
8	Medium	Heavy	Male	Large

this group. C4.5 induces a decision tree during tree generation, splitting cases with missing attribute values into fractions and adding these fractions to new case subsets. A method of *surrogate splits* to handle missing attribute values was introduced in CART [5], yet another system to induce decision trees. Other methods of handling missing attribute values while generating decision trees were presented in [6] and [7]. Additionally, in statistics, *pairwise deletion* [8, 9] is used to evaluate statistical parameters from available information.

In this chapter we assume that the main process of knowledge acquisition is rule induction. Additionally for the rest of the chapter we will assume that all decision values are known, i.e., specified. Also, we will assume that for each case at least one attribute value is known.

41.2 Sequential Methods

In sequential methods to handle missing attribute values original incomplete data sets, with missing attribute values, are converted into complete data sets and then the main process, e.g., rule induction, is conducted.

41.2.1 Deleting Cases with Missing Attribute Values

This method is based on removing cases with missing attribute values. In statistics it is known as *listwise deletion* (or *casewise deletion*, or *complete case analysis*). All cases with missing attribute values are deleted from the original incomplete data set. This method applied to the decision table from Table 41.1 produces a new table, presented in Table 41.2.

Apparently, a lot of information is missing in Table 41.2. However, some author advocate this method as a reasonable technique [8, 9].

Table 41.2 A data set with deleted cases with missing attribute values

Case	Attributes			Decision
	Age	Weight	Gender	
4	Old	Heavy	Female	Small
8	Medium	Heavy	Male	Large

Table 41.3 Data set with missing attribute values replaced by the most common values

Case	Attributes			Decision
	Age	Weight	Gender	
1	Medium	Light	Male	Small
2	Old	Heavy	Female	Small
3	Medium	Light	Male	Small
4	Old	Heavy	Female	Small
5	Medium	Heavy	Male	Large
6	Medium	Heavy	Female	Large
7	Medium	Heavy	Male	Large
8	Medium	Heavy	Male	Large

41.2.2 The Most Common Value of an Attribute

This method is one of the simplest methods to handle missing attribute values. Each missing attribute value is replaced by the most common value of this attribute. In different words, a missing attribute value is replaced by the most probable known attribute value, where such probabilities are represented by relative frequencies of corresponding attribute values. This method of handling missing attribute values is implemented, e.g., in CN2 [10]. In our example from Table 41.1, a result of using this method is presented in Table 41.3.

For case 1, the value of *Age* in Table 41.3 is *medium* since in Table 41.1 the attribute *age* has three values *medium* and two values *old*. Similarly, for case 2, the value of *weight* in Table 41.3 is *heavy* since the attribute *weight* has the value *light* twice and value *heavy* three times.

41.2.3 The Most Common Value of an Attribute Restricted to a Concept

A modification of the method of replacing missing attribute values by the most common value is a method in which the most common value of the attribute restricted to the concept is used instead of the most common value for all cases. Such a concept is the same concept that contains the case with missing attribute value.

Let us say that attribute *a* has missing attribute value for case *x* from concept *C* and that the value of *a* for *x* is missing. This missing attribute value is exchanged by the known attribute value for which the conditional probability $P(\text{known value of } a \text{ for case } x | C)$ is the largest. This method was implemented, e.g., in ASSISTANT [11]. In our example from Table 41.1, a result of using this method is presented in Table 41.4.

For example, in Table 44.1, case 1 belongs to the concept $\{1, 2, 3, 4\}$, all known values of *Age*, restricted to $\{1, 2, 3, 3\}$, are *old*, so the missing attribute value for case 1 and attribute *Age* is replaced by *old*. On the other hand, in Table 41.1, case 2 belongs to the same concept $\{1, 2, 3, 4\}$, and the value of *weight* is missing. The known values of *weight*, restricted to $\{1, 2, 3, 4\}$, are *light* (twice) and *heavy* (once), so the missing attribute value is exchanged by *light*.

41.2.4 Assigning All Possible Attribute Values to a Missing Attribute Value

This approach to missing attribute values was presented for the first time in [12] and implemented in LERS. Every case with missing attribute values is replaced by the set of cases in which every missing attribute value is replaced by all possible known values. In the example from Table 41.1, a result of using this method is presented in Table 41.5.

Table 41.4 Data set with missing attribute values replaced by the most common value of the attribute restricted to a concept

Case	Attributes			Decision
	Age	Weight	Gender	
1	Old	Light	Male	Small
2	Old	Light	Female	Small
3	Old	Light	Female	Small
4	Old	Heavy	Female	Small
5	Medium	Heavy	Male	Large
6	Medium	Heavy	Female	Large
7	Medium	Heavy	Male	Large
8	Medium	Heavy	Male	Large

In the example of Table 41.1, the first case from Table 41.1, with the missing attribute value for attribute *Age*, is replaced by two cases, 1^i and 1^{ii} , where case 1^i has value *medium* for attribute *Age*, and case 1^{ii} has values *old* for the same attribute, since attribute *Age* has two possible known values, *medium* and *old*. Case 2 from Table 41.1, with the missing attribute value for the attribute *weight*, is replaced by two cases, 2^i and 2^{ii} , with values *light* and *heavy*, since the attribute *weight* has two possible known values, *light* and *heavy*, respectively. Note that due to this method the new table, such as Table 41.5, may be inconsistent. In Table 41.5, case 1^{ii} conflicts with case 7^i , case 3^{ii} conflicts with case 7^i , etc. However, rule sets may be induced from inconsistent data sets using standard rough-set techniques, see, e.g., [13–17].

41.2.5 Assigning All Possible Attribute Values Restricted to a Concept

This method was described, e.g., in [18]. Here, every case with missing attribute values is replaced by the set of cases in which every attribute *a* with the missing attribute value has its every possible known

Table 41.5 Data set in which all possible values are assigned to missing attribute values

Case	Attributes			Decision
	Age	Weight	Gender	
1^i	Medium	Light	Male	Small
1^{ii}	Old	Light	Male	Small
2^i	Old	Light	Female	Small
2^{ii}	Old	Heavy	Female	Small
3^i	Medium	Light	Female	Small
3^{ii}	Medium	Light	Male	Small
3^{iii}	Old	Light	Female	Small
3^{iv}	Old	Light	Male	Small
4	Old	Heavy	Female	Small
5^i	Medium	Light	Male	Large
5^{ii}	Medium	Heavy	Male	Large
6^i	Medium	Heavy	Male	Large
6^{ii}	Old	Heavy	Female	Large
7^i	Medium	Light	Male	Large
7^{ii}	Medium	Heavy	Male	Large
8	Medium	Heavy	Male	Large

Table 41.6 Data set in which all possible values, restricted to the concept, are assigned to missing attribute values

Case	Attributes			Decision
	Age	Weight	Gender	
1	Old	Light	Male	Small
2^i	Old	Light	Female	Small
2^{ii}	Old	Heavy	Female	Small
3^i	Old	Light	Female	Small
3^{ii}	Old	Light	Male	Small
4	Old	Heavy	Female	Small
5	Medium	Heavy	Male	Large
6	Medium	Heavy	Female	Large
7	Medium	Heavy	Male	Large
8	Medium	Heavy	Male	Large

value restricted to the concept to which the case belongs. In the example from Table 41.1, a result of using this method is presented in Table 41.6.

In the example of Table 41.1, the first case from Table 41.1, with the missing attribute value for attribute *Age*, is replaced by one with value *old* for attribute *age*, since attribute *Age*, restricted to the concept $\{1, 2, 3, 4\}$ has one possible known value, *old*. Case 2 from Table 41.1, with the missing attribute value for the attribute *weight*, is replaced by two cases, 2^i and 2^{ii} , with values *light* and *heavy*, since the attribute *weight*, restricted to the concept $\{5, 6, 7, 8\}$, has two possible known values, *light* and *heavy*, respectively. Again, due to this method the new table, such as Table 41.6, may be inconsistent.

41.2.6 Replacing Missing Attribute Values by the Attribute Mean

This method is used for data sets with numerical attributes. An example of such a data set is presented in Table 41.7.

In this method, every missing attribute value for a numerical attribute is replaced by the arithmetic mean of known attribute values. In Table 41.7, the mean of known attribute values for the attribute *Age* is 43.6, hence all missing attribute values for *age* should be replaced by 43.6. The table with missing attribute values replaced by the mean is presented in Table 41.8. For the symbolic attribute *gender*, missing attribute values were replaced using the most common value of the attribute.

Table 41.7 An example of a data set with numerical attributes

Case	Attributes			Decision
	Age	Weight	Gender	
1	?	96	Male	Small
2	51	?	Female	Small
3	?	168	?	Small
4	62	186	Female	Small
5	36	?	Male	Large
6	?	171	Female	Large
7	31	?	Male	Large
8	38	205	Male	Large

Table 41.8 Data set in which missing attribute values are replaced by the attribute mean and the most common value

Case	Attributes			Decision
	Age	Weight	Gender	
1	43.6	96	Male	Small
2	51	165.2	Female	Small
3	43.6	168	Male	Small
4	62	186	Female	Small
5	36	165.2	Male	Large
6	43.6	171	Female	Large
7	31	165.2	Male	Large
8	38	205	Male	Large

41.2.7 Replacing Missing Attribute Values by the Attribute Mean Restricted to a Concept

Similarly as the previous method, this method is restricted to numerical attributes. A missing attribute value of a numerical attribute is replaced by the arithmetic mean of all known values of the attribute restricted to the concept. For example from Table 41.7, case 1 has missing attribute value for *age*. Case 1 belongs to the concept {1, 2, 3, 4}. The arithmetic mean of known values of *Age* restricted to the concept, i.e., values 51 and 62, is equal to 56.5, so the missing attribute value is replaced by 56.5. Case 2 belongs to the same concept {1, 2, 3, 4}, the arithmetic mean of 96, 168, and 186 is 150, so the missing attribute value for case is replaced by 150. The table with missing attribute values replaced by the mean restricted to the concept is presented in Table 41.9. For symbolic attributes *gender* missing attribute values were replaced using the most common value of the attribute restricted to the concept.

41.2.8 Global Closest Fit

The global closes fit method [19] is based on replacing a missing attribute value by the known value in another case that resembles as much as possible the case with the missing attribute value. In searching for the closest fit case we compare two vectors of attribute values, one vector corresponds to the case with

Table 41.9 Data set in which missing attribute values are replaced by the attribute mean and the most common value, both restricted to the concept

Case	Attributes			Decision
	Age	Weight	Gender	
1	56.5	96	Male	Small
2	51	150	Female	Small
3	56.5	168	Female	Small
4	62	186	Female	Small
5	36	188	Male	Large
6	35	171	Female	Large
7	31	188	Male	Large
8	38	205	Male	Large

Table 41.10 Distance (1, x)

$d(1, 2)$	$d(1, 3)$	$d(1, 4)$	$d(1, 5)$	$d(1, 6)$	$d(1, 7)$	$d(1, 8)$
3.00	2.66	2.83	2.00	2.69	2.00	2.00

a missing attribute value, the other vector is a candidate for the closest fit. The search is conducted for all cases, hence the name global closest fit. For each case a distance is computed, the case for which the distance is the smallest is the closest fitting case that is used to determine the missing attribute value. Let x and y be two cases. The distance between cases $x = (x_1, x_2, \dots, x_n)$ and $y = (y_1, y_2, \dots, y_n)$ is computed as follows

$$\text{distance}(x, y) = \sum_{i=1}^n \text{distance}(x_i, y_i),$$

where

$$\text{distance}(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i, \\ 1 & \text{if } x \text{ and } y \text{ are symbolic and } x_i \neq y_i, \\ & \text{or } x_i = ? \text{ or } y_i = ? \text{ or } x_i = ? \text{ and } y_i = ?, \\ \frac{|x_i - y_i|}{r} & \text{if } x_i \text{ and } y_i \text{ are numbers and } x_i \neq y_i, \end{cases}$$

where r is the difference between the maximum and minimum of the known values of the numerical attribute with a missing value. If there is a tie for two cases with the same distance, a kind of heuristics is necessary, for example, select the first case. In general, using the global closest fit method may result in data sets in which some missing attribute values are not replaced by known values. Additional iterations of using this method may reduce the number of missing attribute values, but may not end up with all missing attribute values being replaced by known attribute values. Note that in statistics a similar method is called a hot deck imputation.

For the data set in Table 41.7, distances between case 1 and all remaining cases are presented in Table 41.10. For example, the distance $d(1, 3) = 1 + \frac{|168-96|}{|205-96|} + 1 = 2.66$. For case 1, the missing attribute value (for attribute *Age*) should be the value of *Age* for case 5, i.e., 36, since for this case the distance is the smallest. However, the value of *Age* for case 3 is still missing. The table with missing attribute values replaced by the value computed on the basis of the global closest fit is presented in Table 41.11. Some missing attribute values are still present in this table. In such cases it is recommended to use another method of handling missing attribute value to replace all missing attribute values by known attribute values.

Table 41.11 Data set processed by the global closest fit method

Case	Attributes			Decision
	Age	Weight	Gender	
1	36	96	Male	Small
2	51	186	Female	Small
3	?	168	Female	Small
4	62	186	Female	Small
5	36	205	Male	Large
6	62	171	Female	Large
7	31	?	Male	Large
8	38	205	Male	Large

Table 41.12 Data set restricted to the concept {1, 2, 3, 4}

Case	Attributes			Decision
	Age	Weight	Gender	
1	?	96	Male	Small
2	51	?	Female	Small
3	?	168	?	Small
4	62	186	Female	Small

41.2.9 Concept Closest Fit

This method is similar to the global closest fit method. The difference is that the original data set, containing missing attribute values, is first split into smaller data sets, each smaller data set corresponds to a concept from the original data set. More precisely, every smaller data set is constructed from one of the original concepts, by restricting cases to the concept. For the data set from Table 41.7, two smaller data sets are created, presented in Tables 41.12 and 41.13.

Following the data set split, the same global closest fit method is applied to both tables separately. Eventually, both tables, processed by the global fit method, are merged into the same table. In our example from Table 41.7, the final, merged table is presented in Table 41.14.

41.2.10 Other Methods

An *event-covering method* [20, 21], based on an interdependency between known and missing attribute values, is another method handling missing attribute values. The interdependency is computed from contingency tables. The outcome of this method is not necessarily a complete data set (with all attribute values known), just like in the case of closest fit methods.

Another method of handling missing attribute values, called D^3RJ was discussed in [22, 23]. In this method a data set is decomposed into complete data subsets, rule sets are induced from such data subsets, and finally these rule sets are merged.

Yet another method of handling missing attribute values was referred to as Shapiro's method in [24], where for each attribute with missing attribute values a new data set is created, such attributes take place of the decision and vice versa, the decision becomes one of the attributes. From such a table missing attribute values are learned using either a rule set or decision tree techniques. This method, identified as a *chase* algorithm, was also discussed in [25, 26]. In statistics there exist similar methods based on regression.

Learning missing attribute values from summary constraints was reported in [27, 28].

Yet another approach to handling missing attribute values was presented in [29].

Table 41.13 Data set restricted to the concept {5, 6, 7, 8}

Case	Attributes			Decision
	Age	Weight	Gender	
5	36	?	Male	Large
6	?	171	Female	Large
7	31	?	Male	Large
8	38	205	Male	Large

Table 41.14 Data set processed by the concept closest fit method

Case	Attributes			Decision
	Age	Weight	Gender	
1	?	96	Male	Small
2	51	186	Female	Small
3	62	168	Female	Small
4	62	186	Female	Small
5	36	205	Male	Large
6	38	171	Female	Large
7	31	?	Male	Large
8	38	205	Male	Large

There is a number of statistical methods of handling missing attribute values, usually known under the name of *imputation* [8, 9, 30], such as maximum likelihood and the expectation maximization algorithm. Recently *multiple imputation* gained popularity. It is a Monte Carlo method of handling missing attribute values in which missing attribute values are replaced by many plausible values, then many complete data sets are analyzed and the results are combined.

41.3 Parallel Methods

In this section we will concentrate on handling missing attribute values in parallel with rule induction.

We will distinguish three main reasons for missing attribute values. First, some attribute values were not recorded or were mistakenly erased. These attribute values, relevant but missing, will be called *lost*. Secondly, some attribute values were not recorded because they were irrelevant. For example, a doctor was able to diagnose a patient without some medical tests, or a home owner was asked to evaluate the quality of air conditioning while the home was not equipped with an air conditioner. Such missing attribute values will be called '*do not care*' conditions. Additionally, a 'do not care' condition may be replaced by any attribute value limited to the same concept. For example [31], if a patient was diagnosed as not affected by a disease, we may want to replace the missing test (attribute) value by any typical value for that attribute but restricted to patients in the same class (concept), i.e., for other patients not affected by the disease. Such missing attribute value will be called *attribute-concept value*. This type of missing attribute values was introduced in [32].

First we will introduce some useful ideas, such as blocks of attribute-value pairs, characteristic sets, characteristic relations, lower and upper approximations. Later we will explain how to induce rules using the same blocks of attribute-value pairs that were used to compute lower and upper approximations. Input data sets are not preprocessed the same way as in sequential methods, instead, the rule learning algorithm is modified to learn rules directly from the original, incomplete data sets.

41.3.1 Blocks Of Attribute-Value Pairs and Characteristic Sets

In this subsection we will quote some basic ideas of the rough set theory. In this section we will assume that lost values will be denoted by '?', 'do not care' conditions will be denoted by '*', and attribute-concept values will be denoted by '-'.

Let (a, v) be an attribute-value pair. For complete decision tables, a block of (a, v) , denoted by $[(a, v)]$, is the set of all cases x for which $\rho(x, a) = v$. For incomplete decision tables the definition of a block of an attribute-value pair is modified.

- If for an attribute a there exists a case x such that $\rho(x, a) = ?$, i.e., the corresponding value is lost, then the case x should not be included in any blocks $[(a, v)]$ for all values v of attribute a .
- If for an attribute a there exists a case x such that the corresponding value is a ‘do not care’ condition, i.e., $\rho(x, a) = *$, then the case x should be included in blocks $[(a, v)]$ for all specified values v of attribute a .
- If for an attribute a there exists a case x such that the corresponding value is an attribute-concept value, i.e., $\rho(x, a) = -$, then the corresponding case x should be included in blocks $[(a, v)]$ for all specified values $v \in V(x, a)$ of attribute a , where

$$V(x, a) = \{\rho(y, a) \mid \rho(y, a) \text{ is specified}, y \in U, \rho(y, d) = \rho(x, d)\}.$$

This modification of the attribute-value pair block definition is consistent with the interpretation of missing attribute values, lost and ‘do not care’ conditions. For Table 41.15, $V(\text{Age}, 3) = \{\text{old}\}$ and $V(\text{Weight}, 7) = \{\text{heavy}\}$. Thus, for Table 41.1

$$\begin{aligned} [(\text{Age}, \text{old})] &= \{2, 3, 4, 6\}, \\ [(\text{Age}, \text{medium})] &= \{5, 6, 7, 8\}, \\ [(\text{Weight}, \text{light})] &= \{1, 2, 3\}, \\ [(\text{Weight}, \text{heavy})] &= \{2, 4, 6, 7, 8\}, \\ [(\text{Gender}, \text{male})] &= \{1, 3, 5, 7, 8\}, \\ [(\text{Gender}, \text{female})] &= \{2, 3, 4, 6\}. \end{aligned}$$

For a case $x \in U$ the characteristic set $K_B(x)$ is defined as the intersection of the sets $K(x, a)$, for all $a \in B$, where the set $K(x, a)$ is defined in the following way:

- If $\rho(x, a)$ is specified, then $K(x, a)$ is the block $[(a, \rho(x, a))]$ of attribute a and its value $\rho(x, a)$,
- If $\rho(x, a) = ?$ or $\rho(x, a) = *$ then the set $K(x, a) = U$,
- If $\rho(x, a) = -$, then the corresponding set $K(x, a)$ is equal to the union of all blocks of attribute-value pairs (a, v) , where $v \in V(x, a)$ if $V(x, a)$ is nonempty. If $V(x, a)$ is empty, $K(x, a) = U$.

For Table 1 and $B = A$,

$$\begin{aligned} K_A(1) &= U \cap \{1, 2, 3\} \cap \{1, 3, 5, 7, 8\} = \{1, 3\}, \\ K_A(2) &= \{2, 3, 4, 6\} \cap U \cap \{2, 3, 4, 6\} = \{2, 3, 4, 6\}, \\ K_A(3) &= \{2, 3, 4, 6\} \cap \{1, 2, 3\} \cap U = \{2, 3\}, \\ K_A(4) &= \{2, 3, 4, 6\} \cap \{2, 4, 6, 7, 8\} \cap \{2, 3, 4, 6\} = \{2, 4, 6\}, \\ K_A(5) &= \{5, 6, 7, 8\} \cap U \cap \{1, 3, 5, 7, 8\} = \{5, 7, 8\}, \end{aligned}$$

Table 41.15 An example of a data set with lost values, do not care conditions, and attribute-concept values

Case	Attributes			Decision
	Age	Weight	Gender	
1	?	Light	Male	Small
2	Old	*	Female	Small
3	-	Light	*	Small
4	Old	Heavy	Female	Small
5	Medium	?	Male	Large
6	*	Heavy	Female	Large
7	Medium	-	Male	Large
8	Medium	Heavy	Male	Large

$$\begin{aligned}K_A(6) &= U \cap \{2, 4, 6, 7, 8\} \cap \{2, 3, 4, 6\} = \{2, 4, 6\}, \\K_A(7) &= \{5, 6, 7, 8\} \cap \{2, 4, 6, 7, 8\} \cap \{1, 3, 5, 7, 8\} = \{7, 8\}, \text{ and} \\K_A(8) &= \{5, 6, 7, 8\} \cap \{2, 4, 6, 7, 8\} \cap \{1, 3, 5, 7, 8\} = \{7, 8\}.\end{aligned}$$

The characteristic set $K_B(x)$ may be interpreted as the smallest set of cases that are indistinguishable from x using all attributes from B , using a given interpretation of missing attribute values. Thus, $K_A(x)$ is the set of all cases that cannot be distinguished from x using all attributes. For further properties of characteristic sets see [31, 33–37]. Incomplete decision tables in which all attribute values are lost, from the viewpoint of rough set theory, were studied for the first time in [38], where two algorithms for rule induction, modified to handle lost attribute values, were presented. This approach was studied later in [39–41].

Incomplete decision tables in which all missing attribute values are ‘do not care’ conditions, from the view point of rough set theory, were studied for the first time in [12], where a method for rule induction was introduced in which each missing attribute value was replaced by all values from the domain of the attribute. Originally such values were replaced by all values from the entire domain of the attribute, later, by attribute values restricted to the same concept to which a case with a missing attribute value belongs. Such incomplete decision tables, with all missing attribute values being ‘do not care conditions,’ were also studied in [42, 43]. Both approaches to missing attribute values were generalized in [31, 33–36].

41.3.2 Definability

For completely specified decision tables, any union of elementary sets of B is called a B -definable set [44–46]. Definability for completely specified decision tables should be modified to fit into incomplete decision tables. For incomplete decision tables, a union of some intersections of attribute-value pair blocks, where such attributes are members of B and are distinct, will be called B -locally definable sets. A union of characteristic sets $K_B(x)$, where $x \in X \subseteq U$ will be called a B -globally definable set. Any set X that is B -globally definable is B -locally definable, the converse is not true. For example, the set $\{6, 7, 8\}$ is A -locally definable since $\{6, 7, 8\} = [(Age, medium)] \cap [(Weight, heavy)]$. However, the set $\{6, 7, 8\}$ is not A -globally definable. Obviously, if a set is not B -locally definable then it cannot be expressed by rule sets using attributes from B . This is why it is so important to distinguish between B -locally definable sets and those that are not B -locally definable.

41.3.3 Lower and Upper Approximations

For completely specified decision tables lower and upper approximations are defined on the basis of the indiscernibility relation. Let X be any subset of the set U of all cases. The set X is called a *concept* and is usually defined as the set of all cases defined by a specific value of the decision. In general, X is not a B -definable set. However, set X may be approximated by two B -definable sets, the first one is called a B -lower approximation of X , denoted by $\underline{B}X$ and defined as follows

$$\{x \in U \mid [x]_B \subseteq X\}.$$

The second set is called a B -upper approximation of X , denoted by $\overline{B}X$ and defined as follows

$$\{x \in U \mid [x]_B \cap X \neq \emptyset\}.$$

The above shown way of computing lower and upper approximations, by constructing these approximations from singletons x , will be called the *first method*. The B -lower approximation of X is the greatest B -definable set, contained in X . The B -upper approximation of X is the smallest B -definable set containing X .

As it was observed in [45], for complete decision tables we may use a *second method* to define the B -lower approximation of X , by the following formula

$$\cup\{[x]_B \mid x \in U, [x]_B \subseteq X\},$$

and the B -upper approximation of x may be defined, using the second method, by

$$\cup\{[x]_B \mid x \in U, [x]_B \cap X \neq \emptyset\}.$$

Obviously, for complete decision tables both methods result in the same respective sets; i.e., corresponding lower approximations are identical and so are upper approximations.

For incomplete decision tables lower and upper approximations may be defined in a few different ways. In this chapter we suggest three different definitions of lower and upper approximations for incomplete decision tables. Again, let X be a concept, let B be a subset of the set A of all attributes, and let $R(B)$ be the characteristic relation of the incomplete decision table with characteristic sets $K(x)$, where $x \in U$. Our first definition uses a similar idea as in the previous articles on incomplete decision tables [40–43]; i.e., lower and upper approximations are sets of singletons from the universe U satisfying some properties. Thus, lower and upper approximations are defined by analogy with the above first method, by constructing both sets from singletons. We will call these approximations *singleton*. A singleton B -lower approximation of X is defined as follows:

$$\underline{B}X = \{x \in U \mid K_B(x) \subseteq X\}.$$

A singleton B -upper approximation of X is

$$\overline{B}X = \{x \in U \mid K_B(x) \cap X \neq \emptyset\}.$$

In our example of the decision table presented in Table 41.1 let us say that $B = A$. Then the singleton A -lower and A -upper approximations of the two concepts: $\{1, 2, 4, 8\}$ and $\{3, 5, 6, 7\}$ are

$$\begin{aligned}\underline{A}\{1, 2, 3, 4\} &= \{1, 3\}, \\ \underline{A}\{5, 6, 7, 8\} &= \{5, 7, 8\}, \\ \overline{A}\{1, 2, 3, 4\} &= \{1, 2, 3, 4, 6\}, \\ \overline{A}\{5, 6, 7, 8\} &= \{2, 4, 5, 6, 7, 8\}.\end{aligned}$$

As it was observed in, e.g., [31, 33–35], singleton approximations should not be used, in general, for data mining and, in particular, for rule induction.

The second method of defining lower and upper approximations for complete decision tables uses another idea: lower and upper approximations are unions of elementary sets, subsets of U . Therefore we may define lower and upper approximations for incomplete decision tables by analogy with the second method, using characteristic sets instead of elementary sets. There are two ways to do this. Using the first way, a *subset* B -lower approximation of X is defined as follows:

$$\underline{B}X = \cup\{K_B(x) \mid x \in U, K_B(x) \subseteq X\}.$$

A *subset* B -upper approximation of X is

$$\overline{B}X = \cup\{K_B(x) \mid x \in U, K_B(x) \cap X \neq \emptyset\}.$$

Since any characteristic relation $R(B)$ is reflexive, for any concept X , singleton B -lower and B -upper approximations of X are subsets of the subset B -lower and B -upper approximations of X , respectively. For the same decision table, presented in Table 41.1, the subset A -lower and A -upper approximations are

$$\begin{aligned}\underline{A}\{1, 2, 3, 4\} &= \{1, 2, 3\}, \\ \underline{A}\{5, 6, 7, 8\} &= \{5, 7, 8\}, \\ \overline{A}\{1, 2, 3, 4\} &= \{1, 2, 3, 4, 6\}, \\ \overline{A}\{5, 6, 7, 8\} &= \{2, 3, 4, 5, 6, 7, 8\}.\end{aligned}$$

The second possibility is to modify the subset definition of lower and upper approximation by replacing the universe U from the subset definition by a concept X . A *concept B*-lower approximation of the concept X is defined as follows:

$$\underline{B}X = \cup\{K_B(x) \mid x \in X, K_B(x) \subseteq X\}.$$

Obviously, the subset B -lower approximation of X is the same set as the concept B -lower approximation of X . A concept B -upper approximation of the concept X is defined as follows:

$$\begin{aligned}\overline{B}X &= \cup\{K_B(x) \mid x \in X, K_B(x) \cap X \neq \emptyset\} \\ &= \cup\{K_B(x) \mid x \in X\}.\end{aligned}$$

The concept upper approximations were defined in [47] and [48] as well. The concept B -upper approximation of X is a subset of the subset B -upper approximation of X . Besides, the concept B -upper approximations are truly the smallest B -definable sets containing X . For the decision table presented in Table 41.1, the concept A -upper approximations are

$$\begin{aligned}\overline{A}\{1, 2, 3, 4\} &= \{1, 2, 3, 4, 6\}, \\ \overline{A}\{5, 6, 7, 8\} &= \{2, 4, 5, 6, 7, 8\}.\end{aligned}$$

Note that for complete decision tables, all three definitions of lower approximations, singleton, subset and concept, coalesce to the same definition. Also, for complete decision tables, all three definitions of upper approximations coalesce to the same definition. This is not true for incomplete decision tables, as our example shows.

41.3.4 Rule Induction from Incomplete Data – MLEM2

The MLEM2 rule induction algorithm is a modified version of the algorithm LEM2. Rules induced from the lower approximation of the concept *certainly* describe the concept, so they are called *certain*.

On the other hand, rules induced from the upper approximation of the concept describe the concept only *possibly* (or *plausibly*), so they are called *possible* [13]. MLEM2 may induce both certain and possible rules from a decision table with some missing attribute values being lost, some missing attribute values being ‘do not care’ conditions, and some being attribute-concept values, while some attributes may be numerical. For rule induction from decision tables with numerical attributes see [34]. MLEM2 handles missing attribute values by computing (in a different way than in LEM2) blocks of attribute-value pairs, and then characteristic sets and lower and upper approximations. All these definitions are modified according to the two previous subsections, the algorithm itself remains the same.

Rule sets in the LERS format (every rule is equipped with three numbers, the total number of attribute-value pairs on the left-hand side of the rule, the total number of examples correctly classified by the rule during training, and the total number of training cases matching the left-hand side of the rule), induced from the decision table presented in Table 41.1 are:

certain rule set:

1, 3, 3
(Weight, light) -> (Strength, small)

2, 3, 3
(Age, medium) & (Gender, male) -> (Strength, large)
and possible rule set:

1, 3, 4
(Age, old) -> (Strength, small)
1, 3, 3
(Weight, light) -> (Strength, small)
1, 3, 5

(Weight, heavy) -> (Strength, large)
 1, 1, 4
 (Age, old) -> (Strength, large)
 1, 4, 4
 (Age, medium) -> (Strength, large)

41.4 Conclusion

Research on comparison methods dealing with missing attribute values shows that there is no universally best approach [18, 24, 37, 49]. Experiments conducted on many real-life data sets from the UCI Machine Learning Repository (<http://www.ics.uci.edu/mlearn/MLRepository.html>) show that the best strategy of handling missing attribute values depends on a data set [50, 51]. For a specific data set the best method of handling missing attribute values should be chosen individually, using as the criterion of optimality the arithmetic mean of many multifold cross-validation experiments [52]. Thus, methods based on granular computing, such as rough-set approaches to the problem, truly enhance the methodology of dealing with incomplete data.

References

- [1] T. Imielinski and W. Lipski, Jr. Incomplete information in relational databases. *J. ACM* 31 (1984) 761–791.
- [2] W. Lipski, Jr. On semantic issues connected with incomplete information databases. *ACM Trans. Database Syst.* 4 (1979) 262–296.
- [3] W. Lipski, Jr. On databases with incomplete information. *J. ACM* 28 (1981) 41–70.
- [4] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [5] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth & Brooks, Monterey, CA, 1984.
- [6] P. Brazdil and I. Brúha. Processing unknown attribute values by ID3. In: *Proceedings of 4th International Conference on Computing and Information*, Toronto, 1992, pp. 227–230.
- [7] I. Brúha. Meta-learner for unknown attribute values processing: Dealing with inconsistency of meta-databases. *J. Intell. Inf. Syst.* 22 (2004) 71–87.
- [8] P.D. Allison. *Missing Data*. Sage Publications, Thousand Oaks, CA, 2002.
- [9] R.J.A. Little and D.B. Rubin. *Statistical Analysis with Missing Data*, 2nd ed. J. Wiley & Sons, Inc., New York, 2002.
- [10] P. Clark and T. Niblett. The CN2 induction algorithm. *Mach. Learn.* 3 (1989) 261–283.
- [11] I. Kononenko, I. Bratko, and E. Roskar. *Experiments in Automatic Learning of Medical Diagnostic Rules*. Technical Report. Jozef Stefan Institute, Ljubljana, Yugoslavia, 1984.
- [12] J.W. Grzymala-Busse. On the unknown attribute values in learning from examples. In: *Proceedings of the ISMIS-91, 6th International Symposium on Methodologies for Intelligent Systems, 1991. Lecture Notes in Artificial Intelligence*, Vol. 542. Springer-Verlag, Berlin, Heidelberg, New York, 1991, pp. 368–377.
- [13] J.W. Grzymala-Busse. Knowledge acquisition under uncertainty – A rough set approach. *J. Intell. Robotic Syst.* 1 (1988) 3–16.
- [14] J.W. Grzymala-Busse. LERS – A system for learning from examples based on rough sets. In: R. Slowinski (ed.), *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer Academic Publishers, Dordrecht, Boston, London, 1992, pp. 3–18.
- [15] J.W. Grzymala-Busse. A new version of the rule induction system LERS. *Fundam. Inf.* 31 (1997) 27–39.
- [16] J.W. Grzymala-Busse. MLEM2: A new algorithm for rule induction from imperfect data. In: *Proceedings of 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2002, pp. 243–250.
- [17] L. Polkowski and A. Skowron (eds.) *Rough Sets in Knowledge Discovery, 2, Applications, Case Studies and Software Systems, Appendix 2: Software Systems*. Physica Verlag, Heidelberg New York, 1998, pp. 551–601.
- [18] J.W. Grzymala-Busse and M. Hu. A comparison of several approaches to missing attribute values in data mining. In: *Proceedings of 2-nd International Conference on Rough Sets and Current Trends in Computing*, LNAI Series 2005. Springer-Verlag, Heidelberg, Germany, 2000, pp. 340–347.

- [19] J.W. Grzymala-Busse, W.J. Grzymala-Busse, and L.K. Goodwin. A comparison of three closest fit approaches to missing attribute values in preterm birth data. *Int. J. Intell. Syst.* 17 (2002) 125–134.
- [20] D.K. Chiu and A.K.C. Wong. Synthesizing knowledge: A cluster analysis approach using event-covering. *IEEE Trans. Syst. Man Cybern. SMC-16* (1986) 251–259.
- [21] K.C. Wong and K.Y. Chiu. Synthesizing statistical knowledge for incomplete mixed-mode data. *IEEE Trans. Pattern Anal. Mach. Intell.* 9 (1987) 796–805.
- [22] R. Latkowski. On decomposition for incomplete data. *Fundam. Inf.* 54 (2003) 1–16.
- [23] R. Latkowski and M. Mikolajczyk. Data decomposition and decision rule joining for classification of data with missing values. In: *Proceedings of 4-th International Conference on Rough Sets and Current Trends in Computing*, 2004. Lecture Notes in Artificial Intelligence 3066. Springer-Verlag, Heidelberg, Berlin, 2004, pp. 254–263.
- [24] J.R. Quinlan. Unknown attribute values in induction. In: *Proceedings of 6-th International Workshop on Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1989, pp. 164–168.
- [25] A. Dardzinska and Z.W. Ras. Chasing unknown values in incomplete information systems. In: *Proceedings of Workshop on Foundations and New Directions in Data Mining, in conjunction with 3-rd IEEE International Conference on Data Mining*, Melbourne, FL, November 19–22, 2003, pp. 24–30.
- [26] A. Dardzinska and Z.W. Ras. On rule discovery from incomplete information systems. In: *Proceedings of Workshop on Foundations and New Directions in Data Mining, in conjunction with 3-rd IEEE International Conference on Data Mining*, Melbourne, FL, November 19–22, 2003, pp. 31–35.
- [27] X. Wu and D. Barbara. Learning missing values from summary constraints. *ACM SIGKDD Explor. Newslett.* 4 (2002) 21–30.
- [28] X. Wu and D. Barbara. Modeling and imputation of large incomplete multidimensional datasets. In: *Proceedings of 4-th International Conference on Data Warehousing and Knowledge Discovery*, LNCS 2454. Springer-Verlag, Heidelberg, Germany, 2002, pp. 286–295.
- [29] S. Greco, B. Matarazzo, and R. Slowinski. Dealing with missing data in rough set analysis of multi-attribute and multi-criteria decision problems. In: S.H. Zanakis, G. Doukidis, and Z. Zopounidis (eds), *Decision Making: Recent developments and Worldwide Applications*. Kluwer Academic Publishers, Dordrecht, Boston, London, 2000, pp. 295–316.
- [30] J.L. Schafer. *Analysis of Incomplete Multivariate Data*. Chapman and Hall, London, 1997.
- [31] J.W. Grzymala-Busse. Incomplete data and generalization of indiscernibility relation, definability, and approximations. In: *Proceedings of 10-th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*. Springer-Verlag, New York, 2005, pp. 244–253.
- [32] J.W. Grzymala-Busse. Three approaches to missing attribute valuesA rough set perspective. In: *Proceedings Workshop on Foundation of Data Mining, in conjunction with the 4-th IEEE International Conference on Data Mining*, 2004, pp. 55–62.
- [33] J.W. Grzymala-Busse. Rough set strategies to data with missing attribute values. In: *Proceedings of Workshop on Foundations and New Directions in Data Mining, in conjunction with 3-rd IEEE International Conference on Data Mining*, Melbourne, FL, November 19–22, 2003, pp. 56–63.
- [34] J.W. Grzymala-Busse. Data with missing attribute values: Generalization of indiscernibility relation and rule induction. *Trans. Rough Sets Lect. Notes Comput. Sci. J. Subline.* 1 (2004) 78–95.
- [35] J.W. Grzymala-Busse. Characteristic relations for incomplete data: A generalization of the indiscernibility relation. In: *Proceedings of 4-th International Conference on Rough Sets and Current Trends in Computing*, Lecture Notes in Artificial Intelligence 3066. Springer-Verlag, New York, 2004, pp. 244–253.
- [36] J.W. Grzymala-Busse. Rough set approach to incomplete data. In: *Proceedings of 7-th International Conference on Artificial Intelligence and Soft Computing*, 2004, Lecture Notes in Artificial Intelligence 3070. Springer-Verlag, New York, 2004, pp. 50–55.
- [37] J.W. Grzymala-Busse and S. Siddhaye. Rough set approaches to rule induction from incomplete data. In: *Proceedings of 10-th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2004, Vol. 2, pp. 923–930.
- [38] J.W. Grzymala-Busse and A.Y. Wang. Modified algorithms LEM1 and LEM2 for rule induction from data with missing attribute values. In: *Proceedings of 5-th International Workshop on Rough Sets and Soft Computing in conjunction with the 3-rd Joint Conference on Information Sciences*, 1997, pp. 69–72.
- [39] J. Stefanowski. *Algorithms of Decision Rule Induction in Data Mining*. Poznan University of Technology Press, Poznan, Poland, 2001.
- [40] J. Stefanowski and A. Tsoukias. On the extension of rough sets under incomplete information. In: *Proceedings of 7-th International Workshop on New Directions in Rough Sets, Data Mining, and Granular-Soft Computing*, LNCS 1711. Springer-Verlag, Heidelberg, Germany, 1999, pp. 73–81.

- [41] J. Stefanowski and A. Tsoukias. Incomplete information tables and rough classification. *Comput. Intell.* 17 (2001) 545–566.
- [42] M. Kryszkiewicz. Rough set approach to incomplete information systems. In: *Proceedings of 2-nd Annual Joint Conference on Information Sciences*, 1995, pp. 194–197.
- [43] M. Kryszkiewicz. Rules in incomplete information systems. *Inf. Sci.* 113 (1999) 271–292.
- [44] Z. Pawlak. Rough sets. *Int. J. Comput. Inf. Sci.* 11 (1982) 341–356.
- [45] Z. Pawlak. *Rough Sets. Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, Boston, London, 1991.
- [46] Z. Pawlak, J.W. Grzymala-Busse, R. Slowinski, and W. Ziarko. Rough sets. *Commun. ACM* 38 (1955) 88–95.
- [47] T.Y. Lin. Topological and fuzzy rough sets. In: R. Slowinski (ed), *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer Academic Publishers, Dordrecht, Boston, London, 1992, pp. 287–304.
- [48] R. Slowinski and D. Vanderpooten. A generalized definition of rough approximations based on similarity. *IEEE Trans. Knowl. Data Eng.* 12 (2000) 331–336.
- [49] K. Lakshminarayanan, S.A. Harp, and T. Samad. Imputation of missing data in industrial databases. *Appl. Intell.* 11 (1999) 259–275.
- [50] J.W. Grzymala-Busse. Experiments on mining incomplete data – A rough set approach. In: *Proceedings of 11-th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2006, pp. 2586–2593.
- [51] J.W. Grzymala-Busse and S. Santoso. Experiments on data with three interpretations of missing attribute values. A rough set approach. In: *Proceedings of International Conference on Intelligent Information Systems, New Trends in Intelligent Information Processing and WEB Mining*. Springer-Verlag, Heidelberg, Germany, 2006, pp. 143–152.
- [52] S. Weiss and C.A. Kulikowski. How to estimate the true performance of a learning system. In: *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1991, pp. 17–49.

42

Granular Computing in Machine Learning and Data Mining

Eyke Hüllermeier

42.1 Introduction

While aspects of knowledge representation and reasoning have dominated early research in artificial intelligence (AI), problems of automated learning and knowledge acquisition have more and more come to the fore in recent years. This is not very surprising in view of the fact that the ‘knowledge acquisition bottleneck’ seems to remain one of the key problems in the design of intelligent and knowledge-based systems. Indeed, experience has shown that a purely knowledge-driven approach, which aims at formalizing problem-relevant human expert knowledge, is difficult, intricate, tedious, and, more often than not, does not even lead to fully satisfactory results. Consequently, a kind of *data-driven* adaptation or ‘tuning’ of intelligent systems is often worthwhile. In fact, the latter even suggests itself in many applications where data are readily available. Indeed, recent research has shown that the knowledge-driven approach can be complemented or, in the extreme, even replaced by a data-driven one in a reasonable way.

The problem of inducing models by generalizing beyond observed data has been studied intensively in the field of *machine learning* for more than 25 years. Even though the goals of machine learning are quite similar to those in (inductive) statistics, a much older research area, the methods developed in these fields can be seen as complementary. As a response to the progress in digital data acquisition and storage technology, along with the limited human capabilities in analyzing and exploiting large amounts of data, another research discipline has recently received a great deal of attention in diverse research communities. This discipline, which is closely related to both statistics and machine learning, is often referred to as *knowledge discovery in databases* (KDD). According to a widely accepted definition, KDD refers to the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable structure in data [1]. The central step within the overall KDD process is *data mining*, the application of computational techniques to the task of finding patterns and models in data.

The aim of this chapter is to show that ideas and concepts from *granular computing* (GrC) play an important role in machine learning, data mining, and related fields. After a brief introduction to these fields (Section 42.2), some concrete and well-known learning and mining methods are outlined in Section 42.3. The main part of this chapter is Section 42.4, which elaborates on the role of GrC in machine learning and data mining. The chapter concludes with a summary in Section 42.5.

42.2 Machine Learning, Data Mining, and Related Fields

The automated learning of models from empirical data is a central theme in several research disciplines, ranging from classical (inferential) statistics to more recent fields such as machine learning. Model induction may serve different purposes, such as accurate *prediction* of future observations or intelligible *description* of dependencies between variables in the domain under investigation, among other things. Typically, a model induction process involves the following steps:

- data acquisition,
- data preparation (cleaning, transforming, selecting, scaling, . . .),
- model induction,
- model interpretation and validation,
- model application.

A common distinction of performance tasks in empirical¹ machine learning is supervised learning (e.g., classification and regression), unsupervised learning (e.g., clustering), and reinforcement learning. Throughout the chapter, we shall focus on the first two performance tasks that have attracted much more attention in the GrC community than the latter one.

In unsupervised learning, the learning algorithm is simply provided with a set of data. Typically, a single observation z is characterized in terms of a *feature vector*

$$z = (z_1, z_2 \dots z_m) \in \mathcal{Z} = Z_1 \times Z_2 \times \dots \times Z_m,$$

that is, a point in the *feature space* \mathcal{Z} . This feature-based representation assumes a predefined set of m attributes with (categorical or numerical) domains Z_i . However, the data to be analyzed can also be of more general nature. For example, the analysis of complex objects such as sequences, trees, or graphs, which cannot be directly represented as a feature vector of fixed length, has recently received a lot of attention [2].

Roughly speaking, the goal in unsupervised learning is to discover any kind of structure in the data, such as properties of the distribution, relationships between data entities, or dependencies between attributes. This includes, e.g., non-parametric features such as modes, gaps, or clusters in the data, as well as interesting patterns like those discovered in association analysis.

The setting of supervised learning proceeds from a predefined division of the data space into an input space \mathcal{X} and an output space \mathcal{Y} . Assuming a dependency between the input attributes and the output, the former is considered as the *predictive* part of an instance description (like the regressor variables in regression analysis), whereas the latter corresponds to the target to be predicted (e.g., the dependent variable in regression). The learning algorithm is provided with a set of labeled examples $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Again, the inputs x are typically feature vectors. A distinction between different types of performance tasks is made according to the structure of the output space \mathcal{Y} . Even though problems involving output spaces of a richer structure have been considered recently (e.g., so-called ranking problems [3, 4]), \mathcal{Y} is typically a one-dimensional space. In particular, the output is a categorical attribute (i.e., \mathcal{Y} is a nominal scale) in *classification*. Here, the goal is to generalize beyond the examples given by inducing a model that represents a complete mapping from the input space to the output space (a hypothetical classification function). The model itself can be represented by means of different formalisms such as threshold concepts or logical conjunctions. In *regression*, the output is a numerical variable; hence, the goal is to induce a real-valued mapping $\mathcal{X} \rightarrow \mathcal{Y}$ that approximates an underlying (functional or probabilistic) relation between \mathcal{X} and \mathcal{Y} well in a specific sense. So-called *ordinal regression* is in-between regression and classification: the output is measured on an ordinal (ordered categorical) scale.

¹ Here, *empirical* learning is used as an antonym to *analytical* learning. Roughly speaking, analytical learning systems do not require external inputs, whereas such inputs are essential for empirical learning systems. An example of analytical learning is speedup learning.

As can be seen, supervised machine learning puts special emphasis on *induction* as a performance task. Moreover, apart from the *efficiency* of the induced model, the *predictive accuracy* of that model is the most important quality criterion. The latter refers to the ability to make accurate predictions of outputs for so far unseen inputs. The predictive accuracy of a model $h : \mathcal{X} \longrightarrow \mathcal{Y}$ is typically measured in terms of the *expected loss*, i.e., the expected value of $\ell(y, h(x))$, where $\ell(\cdot)$ is a loss function $\mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$ (and (x, y) an example drawn at random according to an underlying probability measure over $\mathcal{X} \times \mathcal{Y}$).²

Data mining has a somewhat different focus.³ Here, other aspects such as the *understandability* gain in importance. In fact, the goal in data mining is not necessarily to induce *global* models of the system under consideration (e.g., in the form of a functional relation between input and output variables) or to recover some underlying data generating process, but rather to discover *local* patterns of interest, e.g., very frequent (hence typical) or very rare (hence atypical) events. Data mining is of a more explanatory nature, and patterns discovered in a data set are usually of a *descriptive* rather than of a *predictive* nature. Data mining also puts special emphasis on the analysis of very large data sets and, hence, on aspects of scalability and efficiency.

Despite these slightly different goals, the typical KDD process has much in common with the process of inductive reasoning as outlined above, except for the fact that the former can be (and indeed often is) circular in the sense that the data mining results will retroact on the acquisition, selection, and preparation of the data, possibly initiating a repeated pass with modified data, analysis tools, or queries. A typical KDD process may comprise the following steps:

- data cleaning,
- data integration (combination of multiple sources),
- data selection,
- data transformation (into a form suitable for the analysis),
- data mining,
- evaluation of patterns,
- knowledge presentation.

Recently, the interest in data mining has shifted from the analysis of large but homogeneous data sets (relational tables) to the analysis of more complex and heterogeneous information sources such as texts, images, audio, and video data, and the term *information mining* has been coined to describe a KDD process focused on this type of information sources [5].

There are several other fields that are closely related to machine learning and data mining, such as classical statistics and various forms of data analysis (distinguished by adjectives like multivariate, exploratory, Bayesian, intelligent, ...). Needless to say, it is impossible to set a clear boundary between these fields. Subsequently, we shall simply subsume them under the heading ‘machine learning and data mining’ (ML&DM), understood in a wide sense as the application of computational methods and algorithms for extracting models and patterns from potentially very large data sets.

42.3 Exemplary Methods

In this section, we briefly outline some well-known ML&DM methods to which we shall occasionally refer to in later sections. The methods themselves are relatively simple and, moreover, will be introduced in their basic form. In fact, our intention is not to provide a state-of-the-art review of machine learning

² Since this measure is normally unknown, the expected loss is approximated by the *empirical* loss in practice, i.e., the average loss on a test data set.

³ Our distinction between machine learning and data mining can roughly be seen as a ‘modern’ or extended distinction between descriptive and inductive statistics. We note, however, that this view is not an *opinio communis*. For example, some people have an even more general view of data mining that also subsumes machine learning methods.

and data mining techniques, but rather to illustrate and emphasize the role of granular computing within these fields.

Subsequently, we restrict ourselves to the basic setting of (un-)supervised learning in which examples are represented in terms of feature vectors. Thus, let \mathcal{X} denote an instance space, where an instance corresponds to the attribute-value description x of an object. In this case, $\mathcal{X} = X_1 \times X_2 \times \cdots \times X_m$, with X_i the domain of the i th attribute, and an instance is represented as a vector $x = (x^1 \dots x^m) \in \mathcal{X}$. In the case of supervised learning, we will focus on the problem task of classification, so the output space is given by a finite set of labels (classes) $\mathcal{Y} = \{\lambda_1 \dots \lambda_c\}$. Training data shall be given in the form of a set $\mathcal{T} \subseteq \mathcal{X} \times \mathcal{Y}$ of examples.

42.3.1 Nearest Neighbor Classification

In k -nearest neighbor (k -NN) classification [6], the label y_0^{est} (hypothetically) assigned to a query x_0 is given by the label that is most frequent among x_0 's k nearest neighbors, where nearness is measured in terms of a similarity or distance function, typically the Euclidean metric. In weighted k -NN, the neighbors are moreover weighted by their distance [7]:

$$y_0^{est} \stackrel{\text{df}}{=} \arg \max_{j=1\dots c} \sum_{i=1}^k \omega_i \mathbb{I}(\lambda_j = y_i), \quad (1)$$

where x_i is the i th nearest neighbor of x_0 in the training set \mathcal{T} ; y_i and ω_i are, respectively, the label and the weight of x_i , and $\mathbb{I}(\cdot)$ is the standard $\{\text{true, false}\} \rightarrow \{0, 1\}$ mapping. A simple definition of the weights is $\omega_i = 1 - d_i \cdot (\sum_{j=1}^k d_j)^{-1}$, where the d_i are the corresponding distances.

NN estimation provides the basis of the class of *case-based* or *instance-based* learning methods (also known as memory-based [8] or exemplar-based [9]). As opposed to conventional inductive learning methods, case-based algorithms do not perform inductive inference in the sense of replacing the original data by a hypothetical model. In fact, instead of inducing a model, these methods learn by simply storing (some of) the observed examples. They defer the processing of these inputs until a prediction (or some other type of query) is actually requested (which qualifies them as *lazy* learning methods [10]). Predictions are then derived by combining the information provided by the stored examples in some way or other. After the query has been answered, the prediction itself and any intermediate results are discarded. Lazy learning methods are quite efficient in the training phase, since training basically comes down to storing new cases or, more generally, to maintaining a proper case base. At classification (prediction) time, lazy methods are comparably expensive, as the prediction step always involves a nearest neighbor search (the complexity of which is at least logarithmic in the number of stored cases [11]).

From a granular computing point of view, it is interesting to mention that the simple 1-NN classifier associates a Voronoi diagram with a training set \mathcal{T} : Every instance x_i in \mathcal{T} defines a ‘granule’ in the form of a ‘spheres of influence,’ namely, the subset of instances $x \in \mathcal{X}$ for which x_i is the nearest neighbor in \mathcal{T} . Hypothetically, all these instances would be assigned to the class of x_i .

42.3.2 Rule Induction

Rule induction algorithms are among the most well-known and widely applied (supervised) machine learning methods. In rule induction, a hypothetical relation between the input space \mathcal{X} and an output space \mathcal{Y} is expressed in terms of a rule base, i.e., a set or a list of rules $(r_1 \dots r_m)$. The antecedent part of a rule r_i is typically a conjunction of *selectors* on the individual attributes, where a selector is a condition on the attribute value in the form of a logical predicate. In the case of classification, the conclusion part is simply a class assignment. Roughly speaking, a rule is understood as a logical implication, the conclusion of which becomes valid if the precondition is satisfied; for example,

$$\text{IF } (x_1 = m) \text{ AND } (x_2 \leq 25) \text{ THEN } (y = 1).$$

Rule induction is an interesting learning approach for several reasons. Notably, a rule-based model is convenient from a knowledge representation point of view, since IF–THEN rules of the above type are easily understandable by human beings. This distinguishes rule induction from ‘black box’ models such as neural networks or kernel machines.

Standard rule induction methods (including, e.g., AQ-family [12, 13], CN2 [14, 15], RIPPER [16], and FOIL [17]) have been applied successfully in numerous applications. Typically, such algorithm either follow a ‘separate and conquer’ (covering) or a ‘divide and conquer’ strategy [18]. In the first case, rules are learned in succession, one by one. In each step of the iteration, one tries to find a single ‘best’ rule (by choosing corresponding selectors) that covers as many as possible examples of a single class but no, or at least comparatively few, examples of other classes; in this context, a rule is said to ‘cover’ an instance if its premise part is satisfied by this instance. The covered examples are then removed from the training set, and the same process is repeated until all examples are covered. In the divide & conquer approach, the instance space is split according to the value of an appropriately chosen attribute. This is done in a recursive way; i.e., the same procedure is applied to every part thus obtained, until the part becomes ‘pure’ enough (almost exclusively contains examples from a single class). This recursive partitioning scheme produces a set of rules with a hierarchical structure, usually called a *decision tree* [19].

Since every rule r_i is associated with a certain subset $\mathcal{R}_i \subseteq \mathcal{X}$ of the instance space (usually an axis-parallel rectangle) or, on the ‘empirical’ level, the subset of the training data that it covers, it might be considered as a ‘granule’ from a GrC point of view. In this connection, it is interesting to mention that the separate and conquer strategy, as opposed to the divide and conquer scheme, does usually not produce a rule base which is complete and consistent in the sense that every potential query instance $x \in \mathcal{X}$ is covered by one and only one rule. Instead, it may happen that an instance is not covered by any rule or that it is covered by more than one rule. Consequently, in order to evaluate a rule base, one needs both a default decision and a strategy for resolving conflicts. The default decision usually consists of predicting the majority class. To resolve conflicts, various strategies are conceivable. For example, the rules can be considered as an ordered list (decision list), which means that a rule is applied only if it is the first one among those the condition part of which is satisfied. Alternatively, a kind of voting scheme can be used, just like in nearest neighbor classification: every rule gives a vote in favor of a class, and these votes are aggregated into a final decision.

42.3.3 Association Analysis

Association analysis, a particular type of dependency analysis, is a widely applied data mining technique that has been studied intensively in recent years [20]. The goal in association analysis is to find ‘interesting’ associations in a data set, that is, dependencies between so-called *itemsets* \mathcal{A} and \mathcal{B} expressed in terms of rules of the form $\mathcal{A} \rightarrow \mathcal{B}$. To illustrate, consider the well-known example where items are products and a data record (transaction) \mathcal{I} is a shopping basket such as {butter, milk, bread}. The intended meaning of an association $\mathcal{A} \rightarrow \mathcal{B}$ is that if \mathcal{A} is present in a transaction, then \mathcal{B} is likely to be present as well. For example, the rule {butter, bread} \rightarrow {milk} suggests that a shopping basket that contains butter and bread typically also contains milk.

In the above setting, a single item can be represented in terms of a binary (0/1-valued) attribute reflecting its presence or absence in a transaction. To make association analysis applicable to data sets involving numerical attributes, such attributes are typically discretized into intervals, and each interval is considered as a new binary attribute. For example, the attribute **temperature** might be replaced by two binary attributes **cold** and **warm**, where **cold** = 1 (**warm** = 0) if the temperature is below 10° and **warm** = 1 (**cold** = 0) otherwise.

A basic problem in association analysis is to find all rules $\mathcal{A} \rightarrow \mathcal{B}$ the *support* (relative frequency of transactions \mathcal{I} with $\mathcal{A} \cup \mathcal{B} \subseteq \mathcal{I}$) and *confidence* (relative frequency of transactions \mathcal{I} with $\mathcal{B} \subseteq \mathcal{I}$ among those with $\mathcal{A} \subseteq \mathcal{I}$) of which reach user-defined thresholds **minsupp** and **minconf**, respectively. Since the number of potential rules is exponential in the number of attributes, this problem is algorithmically quite challenging. The mining of association rules heavily exploits the structure of patterns which presents

itself in the form of a generalization/specialization relation. Several efficient algorithms have been devised so far [21–23]. Typically, such algorithms perform by generating a set of candidate rules from selected itemsets which are then filtered according to several quality criteria. For example, the well-known APRIORI algorithm [21] generates rules from *frequent itemsets*. Thus, the problem of finding sufficiently supported rules reduces to the problem of finding frequent (=sufficiently supported) itemsets, which constitutes the main part of the APRIORI algorithm. Alternative techniques have been developed to avoid the costly process of candidate generation and testing (e.g. [24–26]); see [27, 28] for a comparison of different mining algorithms and [29] for a report on the performance of different frequent itemset mining implementations on selected real-world and artificial databases.

42.4 Granular Computing in Machine Learning and Data Mining

According to a commonly accepted definition, *granulation* refers to the grouping of elements based on their indistinguishability, similarity, proximity, or functionality. In the literature, different links have been identified that allow for considering a certain type of problem or application from the viewpoint of granular computing, or even for interpreting a certain method as a form of granular computation.

Firstly, referring to the purely formal representation of a granule as a subset of a reference set, every kind of set-theoretic analysis may be considered from a GrC point of view. An examples from the field of ML&DM is the *version space model* of inductive learning [30], which exploits a partial specificity ordering of hypotheses in order to provide an efficient representation of the set of all hypotheses of an underlying hypothesis space (the reference set) that are consistent with a given set of data. Other examples include formal concept analysis [31] and association analysis (cf. Section 42.3.3), where the partial order between subsets (defined by the inclusion relation) plays a fundamental role. Nevertheless, one should consider this ‘set-theoretic’ link with reservation, since equating granules with subsets on a purely syntactic level completely ignores the semantic dimension of GrC [32]. In fact, not every subset is a granule!

Secondly, many ML&DM methods produce some kind of ‘granulation’ of the data space, either during the learning process itself or as an output. An obvious example is provided by *clustering methods*. Clustering is indeed of major importance for both unsupervised learning and GrC; however, as this topic is covered in depth elsewhere in this volume [33], we shall not go into much detail in this chapter. Another example is the partitioning of an instance space induced by the *decision boundaries* of a classifier: Every subset $D_\lambda = \{x \in \mathcal{X} \mid h(x) = \lambda\} \subseteq \mathcal{X}$ of instances hypothetically labeled with class λ by a classifier function $h : \mathcal{X} \rightarrow \mathcal{Y}$ may be seen as a granule and, hence, the process of creating such a partitioning by learning a classifier as granular computation. This interpretation is especially obvious for recursive partitioning methods like decision tree induction, and indeed, such methods are often considered as instances of granular computation. Of course, instead of simply reinterpreting established concepts like clustering or partitioning in terms of GrC, one should carefully ask whether ‘information granules’ and related ideas from GrC do have a distinguished role to play, in the sense of being actively involved in the inductive reasoning or pattern discovery process. We shall give examples of corresponding methods in Sections 42.4.1 and 42.4.4 below.

Thirdly, there are several formalisms for modeling and processing uncertain and imprecise information that are naturally associated with and actually provide the formal foundation of granular computation, notably rough set theory [34, 35] (see also [36–38] for recent surveys) and fuzzy set theory [39]. We shall recall basic ideas from rough set-based data analysis in Sections 42.4.1 and 42.4.3. Moreover, an overview of the application of fuzzy sets in ML&DM is given in Section 42.4.2.

42.4.1 Data Preprocessing

Data preprocessing is of major importance in ML&DM, as it strongly influences the success and efficiency of the learning and analysis methods to be applied afterward. One commonly employed preprocessing step is the *discretization* of a numerical attribute, that is, the construction of a finite partition of the domain

of that attribute. This step is often necessary since many ML&DM methods can handle only discrete attributes with finite domains.

Another common preprocessing step is a reduction of the size of the input data. Assuming this data to be given in terms of an attribute-value (feature vector) representation, there are basically two possibilities: reducing the number of rows of the corresponding data table, and reducing the number of columns. The latter corresponds to *feature selection* or, more generally, *dimensionality reduction*. Feature (or attribute) selection is quite important, since irrelevant attributes may badly deteriorate the performance of a machine learning method. A large repertoire of methods for dimensionality reduction exists, such as the well-known statistical method of PCA. In Section 42.4.1, we present an approach which is based on rough set theory and, hence, is very akin to granular computation. Reducing the number of rows of a data table, i.e., the number of examples, is often important for reasons of efficiency. In fact, the complexity of most learning algorithms is superlinear in the number of examples, so applying them to huge data sets can become problematic. The simplest approach in this connection is *sampling*, that is, selecting a subset of the original training data (at random). In Section 42.4.1, we discuss an alternative approach based on ideas and concepts from GrC.

Discretization

Discretization is a topic that has received a lot of attention in the field of ML&DM in the last decade. In fact, a plethora of concrete methods is now available, ranging from the simplest ones like equiwidth partitioning (binning) to more sophisticated ones, such as entropy-based partitioning [40] or the more recently proposed CAIM algorithm [41]. A comprehensive survey of these methods is clearly beyond the scope of this chapter.

In [42], discretization methods have been distinguished along the following dimensions: global versus local, supervised versus unsupervised, dynamic versus static. Global methods, such as binning, discretize all continuous attributes independently of each other and thus produce a mesh over the entire instance space. In the case of m continuous features, this mesh thus consists of $k_1 \times \dots \times k_m$ regions, where k_i is the size of the partition of the i th attribute. As opposed to this, local methods produce separate partitions for local regions of the instance space. Thus, depending on the region of the instance space (values of the other attributes), a numerical value x of an attribute A might be grouped with different values of the same attribute. Local methods of that kind are typically used, for example, in recursive partitioning methods like decision tree induction. Unsupervised discretization methods do only take the attribute values into account but ignore the class labels of the training data. As opposed to this, supervised methods try to additionally exploit this information, that is, to find a discretization that optimally supports the design of a classifier to be learned afterward. Static methods perform a single pass over the data and determine an optimal granularity (number of intervals) for each feature independently of the other attributes. Thus, potential interdependencies between the features cannot be captured. As opposed to this, dynamic methods aim at finding the optimal granularity for all features simultaneously.

From a GrC point of view, it seems important to emphasize the advantages of using fuzzy sets in discretization. In fact, an obvious drawback of an interval-based partition is the abrupt transition between the ‘information granules’ defined by these intervals. Depending on the learning or data mining method applied to the discretized data, this may lead to undesirable effects like discontinuity or instability. For example, replacing the numerical attribute `size` by a discrete attribute that assumes values `short`, `medium`, and `tall` with associated intervals $(0, 150]$, $[150, 180]$, and $(180, 250)$, respectively, two persons whose size differs by only 1 mm may fall into completely different categories. An obvious idea to avoid such problems is to replace intervals by fuzzy sets, that is, to define a discretization as a collection of k fuzzy sets $F_1 \dots F_k$. These fuzzy sets are typically overlapping, thereby creating soft transition boundaries between information granules. The choice of the fuzzy sets is usually restricted by a number of requirements the partition should fulfill; for example, the fuzzy sets are typically assumed to form a partition of unity; that is, $F_1 + \dots + F_k \equiv 1$ [43]. From a knowledge interpretation point of view, fuzzy sets are furthermore appealing due to their linguistic interpretability. For example, given that the fuzzy sets forming a partition can be associated with reasonable linguistic terms (like `tall` in the case of `size`), the patterns discovered by a data mining method can be presented to the user in a very comprehensible way.

Feature Selection Using Rough Sets

In the context of supervised learning, the principle goal of dimensionality reduction is to embed the original input data in a space of smaller dimension while not loosing any information that might be useful for inducing a predictive model. In the case of feature selection, this space is simply obtained by removing some of the original dimensions (attributes), whereas approaches like PCA allow for more general types of transformation (linear projections).

The aforementioned goal can nicely be formalized within the framework of rough set theory (RST), and indeed, various approaches to RST-based feature selection have been developed (see e.g. [44]). Consider an information system $S = \langle \Omega, \mathcal{A}, \mathcal{X}_\mathcal{A}, \varphi \rangle$, where Ω is a finite set of objects and \mathcal{A} is a finite set of attributes; every attribute $a \in \mathcal{A}$ has an associated domain \mathcal{X}_a , and $\mathcal{X}_\mathcal{A} \stackrel{\text{df}}{=} \bigcup_{a \in \mathcal{A}} \mathcal{X}_a$. The information function φ is an $\Omega \times \mathcal{A} \rightarrow \mathcal{X}_\mathcal{A}$ mapping such that $\varphi(\omega, a) \in \mathcal{X}_a$; thus, every object $\omega \in \Omega$ can be mapped to its representation in terms of a feature vector:

$$\varphi(\omega) = (\varphi(\omega, a_1) \dots \varphi(\omega, a_m)).$$

Given an equivalence relation Π on Ω , the equivalence classes

$$\Pi(\omega) = \{\omega' \in \Omega \mid (\omega, \omega') \in \Pi\}$$

are also referred to as *information granules*. The idea is that an equivalence relation reflects the availability of information about the objects in Ω . If this information is not detailed enough, it might be impossible to distinguish two objects which hence appear to be equivalent. In other words, an information granule is a collection of objects indistinguishable by the information at hand. A tuple (Ω, Π) is often called a *granular space*.

Of particular interest in the context of information systems is a special type of equivalence relation: An *indiscernibility relation* $\mathcal{I}_\mathcal{B}$ is induced by a subset of attributes $\mathcal{B} \subseteq \mathcal{A}$ and defined by

$$(\omega, \omega') \in \mathcal{I}_\mathcal{B} \stackrel{\text{df}}{\iff} \forall a \in \mathcal{B} : \varphi(\omega, a) = \varphi(\omega', a).$$

In plain words, $(\omega, \omega') \in \mathcal{I}_\mathcal{B}$ means that two objects ω and ω' cannot be distinguished in terms of the attribute set \mathcal{B} , as they share the same value for every attribute $a \in \mathcal{B}$.

Consider the problem to approximate an ordinary set $U \subseteq \Omega$ at the level of detail dictated by an equivalence relation Π , that is, in terms of a union of corresponding information granules. Of course, depending on the granularity of the quotient space Ω/Π , a precise characterization of U will not always be possible. Instead, in rough set theory, U is approximated in terms of a pair $(\underline{U}_\Pi, \overline{U}^\Pi)$, called the lower and upper approximation, respectively:

$$\begin{aligned} \underline{U}_\Pi &\stackrel{\text{df}}{=} \{\omega \in \Omega \mid \Pi(\omega) \subseteq U\} \\ \overline{U}^\Pi &\stackrel{\text{df}}{=} \{\omega \in \Omega \mid \Pi(\omega) \cap U \neq \emptyset\} \end{aligned}$$

Given the information at hand, one can say that every $\omega \in \underline{U}_\Pi$ *certainly* belongs to U , since even all those objects it cannot be distinguished from do belong to U . Every $\omega \in \overline{U}^\Pi$ *possibly* belongs to U , as it is indistinguishable from at least one element of U .

Now, coming back to the setting of supervised learning, let the attribute set \mathcal{A} be given by $\mathcal{A} = \mathcal{A}_{in} \cup \{y\}$, where y is a distinguished output (decision, class) attribute with domain $\mathcal{Y} = \mathcal{X}_y = \{\lambda_1 \dots \lambda_c\}$ and \mathcal{A}_{in} corresponds to the input attributes. The indistinguishability relation $\mathcal{I}_{\{y\}}$ induced by the decision attribute partitions the object set Ω into the class representatives. Thus, for every class y_i ($i = 1 \dots c$),

$$D_i \stackrel{\text{df}}{=} \{\omega \in \Omega \mid \varphi(\omega, y) = \lambda_i\}$$

is an equivalence class namely, the objects with class label λ_i .

The *quality of approximation* that can be achieved by a subset $\mathcal{B} \subseteq \mathcal{A}_{in}$ of attributes is defined by

$$\gamma_{\mathcal{B}}(\Omega, y) \stackrel{\text{df}}{=} \frac{1}{\text{card}(\Omega)} \sum_{i=1}^c \text{card}\left(\underline{C}_{i_{\mathcal{I}_{\mathcal{B}}}}\right).$$

Note that, given the information provided by the attributes \mathcal{B} , $\underline{C}_{i_{\mathcal{I}_{\mathcal{B}}}}$ corresponds to the set of objects that can *certainly* be assigned to the class λ_i since, by definition, none of these objects is indistinguishable from any other object having an other class. Obviously, $\gamma_{\mathcal{B}}(\Omega, y)$ assumes values between 0 and 1. A *reduct* is any (non-empty) subset of attributes $\mathcal{B} \subseteq \mathcal{A}_{in}$ such that $\gamma_{\mathcal{B}}(\Omega, y) = \gamma_{\mathcal{A}_{in}}(\Omega, y)$, that is, an attribute set having the same approximation quality as the original set \mathcal{A}_{in} . Or, stated differently, \mathcal{B} is a reduct if the granulation of Ω induced by the relation $\mathcal{I}_{\mathcal{B}}$ is a *refinement* of the granulation induced by $\mathcal{I}_{\{y\}}$.

Using rough set terminology, the original goal of a lossless though as effective as possible feature reduction can now be formulated in a concise way as follows: Find a *minimal* reduct \mathcal{B} , that is, a reduct \mathcal{B} such that $\text{card}(\mathcal{B}) \leq \text{card}(\mathcal{B}')$ for all (non-empty) reducts \mathcal{B}' . Please note that there will usually not exist a unique minimal reduct.

In principle, the problem of finding a minimal reduct can be solved by searching the space of all feature subsets of \mathcal{A}_{in} in a systematic way. This, however, becomes intractable with a growing number of features, and indeed, the minimal reduct problem itself is NP-hard [45].⁴ A possible way out is to use heuristic search method, thereby gaining efficiency at the cost of optimality. A relatively straightforward idea, for example, is to implement a forward selection procedure, i.e., to start with the empty attribute set and to add single attributes in a greedy manner. The selection of the next attribute to be added to the current feature subset \mathcal{B} can be made, e.g., on the basis of the *significance*. For an attribute a , the latter is defined by $\gamma_{\mathcal{B}}(\Omega, y) - \gamma_{\mathcal{B} \cup \{a\}}(\Omega, y)$, where $\mathcal{B}' = \mathcal{B} \cup \{a\}$. Thus, the idea is to successively add attributes of highest significance until no further attribute with positive significance exists [47].

It deserves mentioning that, from a machine learning point of view, insisting on reducts with $\gamma_{\mathcal{B}}(\Omega, y) = 1$ (or $\gamma_{\mathcal{B}}(\Omega, y) = \gamma_{\mathcal{A}_{in}}(\Omega, y)$ in the case where $\gamma_{\mathcal{A}_{in}}(\Omega, y) < 1$) is not necessarily useful. In fact, it is well known that in model induction a careful distinction must be made between reproducing the training data and having high classification performance on new, so far unseen (test) data. Roughly speaking, reproducing the training data in too exact a manner often comes along with the problem of overfitting, that is, inducing a model with high accuracy on the training set but low predictive performance on new data. A simple approach to alleviate this problem is to terminate the above forward selection algorithm as soon as the approximation quality reaches a predefined threshold $t < 1$ [48]. However, more sophisticated approaches to avoiding overfitting and dealing with noisy data have also been developed, e.g., based on the concept of so-called *dynamic reducts* [49].

The concepts of *vagueness* and *indiscernibility* as modeled, respectively, by fuzzy sets and rough sets, are related but distinct and in fact complementary [50]. This motivates a combination of the two approaches, giving rise to fuzzy–rough sets and rough–fuzzy sets [51]. Amongst other advantages, fuzzy–rough sets allow for handling both categorical and numerical attributes simultaneously in connection with feature selection [48].

Data Reduction

As mentioned previously, a simple approach to reducing the number of training examples is to sample the original data set.⁵ From a granular computing point of view, an alternative and rather obvious idea is to replace the original data set by a smaller number of ‘granular examples,’ where a granular example refers to a kind of representative or prototype for a subset of (similar) original data points. Thus, the original training examples are compressed into a smaller number of granular examples.

⁴ Likewise, finding all minimal reducts has exponential complexity [46].

⁵ Sampling or, more precisely, resampling techniques are not only used for this but also for other purposes, e.g., for creating diversity in ensemble methods.

To achieve such a compression, the authors in [52] introduce a so-called admission function. A function of this type is defined for every granule and serves as a kind of filter. For example, if granules correspond to intervals or, more generally, (hyper-)rectangles, the admission function might map every data point inside the rectangle to the center point of that rectangle and ignore (filter out) all others. Thus, the compressed data set, given by the union of the outputs of all admission functions, would be given by the center points of those rectangles that cover at least one of the original data points.

In the above approach, the set of granules is assumed to be predefined. Besides, the approach is restricted in the sense that a granular example is again an element of the original data space \mathcal{Z} . In fact, what the method actually implements is a kind of vector quantization.

In this connection, one might of course think of more general alternatives. First, it might be interesting to implement a method which is *data driven* in the sense that the granules are determined by the original training data $\mathcal{T} \subseteq \mathcal{Z}$ itself. Moreover, a granular example need not necessarily be an element of \mathcal{Z} but could be a more complex entity. This might be reasonable in order to preserve as much information about \mathcal{T} as possible.

To give an example, the data set \mathcal{T} might first be partitioned into clusters of similar (closely neighbored) examples; this could be done using standard clustering techniques. Then, a granular example might be defined for every cluster $C \subseteq \mathcal{T}$, namely, by the smallest rectangle $G \subseteq \mathcal{Z}$ such that $C \subseteq G$. Similarly, subsets of \mathcal{T} might be replaced by fuzzy sets instead of rectangles; see [53] for a related approach to creating ‘fuzzy summaries’ of the data. In any case, a granular example can be considered as a compact, approximate representation of a collection of conventional examples. Note that it might be useful to associate a *weight* with every granular example which reflects the number of original examples it represents. We also like to point out that a transformation of a data set into a set of granular examples should not be confused with what is typically understood by *discretization* (of numerical data). In particular, granular examples might well be overlapping, and their union does not necessarily cover the data space. (Besides, a discretization does not change the number of examples but only turns numerical into discrete attributes.)

Of course, as a consequence of this more general approach, the preprocessing step transforms the original data (e.g., a set of feature vectors) into a set of more complex entities (e.g., rectangles) which constitutes the input for the learning method applied afterward. Thus, the latter must be able to handle the corresponding type of input, which necessitates an extension of conventional learning methods. Even though an extension of this kind will not be possible in every case, or may at least be cumbersome, it can be realized quite easily for several learning methods; an example will be given in Section 42.4.4.

42.4.2 Fuzzy Sets in Learning and Data Mining

The field of machine learning and data mining has received a great deal of attention in the fuzzy sets community in recent years [54]. Among the various contributions that fuzzy methods can make to ML&DM, let us highlight the following points:

- **Graduality:** The ability to represent gradual concepts and fuzzy properties in a thorough way is one of the key features of fuzzy sets, and this aspect is also of primary importance in the context of ML&DM. In data mining, for example, the patterns of interest are often vague and have boundaries that are non-sharp in the sense of fuzzy set theory.
- **Interpretability:** Fuzzy sets have the capability to interface quantitative patterns with qualitative knowledge structures expressed in terms of natural language. This makes the application of fuzzy technology very appealing from a knowledge representational point of view.
- **Robustness:** ML&DM methods using fuzzy sets instead of intervals for representing (granular) data, patterns, and models are potentially more robust, e.g., with respect to slight variations of the data, as they avoid undesirable boundary effects.
- **Uncertainty:** In ML&DM, like in other fields, fuzzy sets and related uncertainty formalisms can complement probability theory in a reasonable way, because not all types of uncertainty relevant to machine learning are of a probabilistic nature.

- Background knowledge: Fuzzy-set-based modeling techniques provide a convenient tool for making expert knowledge accessible to computational methods and, hence, to incorporate background knowledge in the learning process.

In the following, we briefly outline some typical applications of fuzzy set theory in ML&DM, emphasizing motivations and differences to conventional (non-fuzzy) approaches; see [54] for a more thorough discussion.

Fuzzy Cluster Analysis

In conventional clustering, every object is assigned to one cluster in an unequivocal way. Consequently, the individual clusters are separated by sharp boundaries. In practice, such boundaries are often not very natural or even counterintuitive. Instead, the boundary of single clusters and the transition between different clusters are usually ‘smooth’ rather than abrupt. This is the main motivation underlying fuzzy extensions to clustering algorithms [55]. In fuzzy clustering an object may belong to different clusters at the same time, at least to some extent, and the degree to which it belongs to a particular cluster is expressed in terms of a fuzzy membership. The membership functions of the different clusters (defined on the set of observed points) is usually assumed to form a partition of unity. This version, often called *probabilistic clustering*, can be generalized further by weakening this constraint: In *possibilistic clustering*, the sum of membership degrees is constrained to be *at least* one [56]. Fuzzy clustering has proved to be extremely useful in practice and is now routinely applied also outside the fuzzy community (e.g., in recent bioinformatics applications [57]). We refer to [33] in this volume for a more detailed discussion of (fuzzy) clustering.

Learning Fuzzy Rule Bases

The most frequent application of FST in machine learning is the induction or the adaptation of rule-based models. This is hardly astonishing, since rule-based models have always been a cornerstone of fuzzy systems and a central aspect of research in the whole field. Fuzzy rule bases can represent both classification and regression functions, and different types of fuzzy models have been used for these purposes. In order to realize a regression function, a fuzzy system is usually wrapped in a ‘fuzzifier’ and a ‘defuzzifier’: The former maps a crisp input to a fuzzy one, which is then processed by the fuzzy system, and the latter maps the (fuzzy) output of the system back to a crisp value.

In the case of classification learning, the consequent of single rules is usually a class assignment (i.e. a singleton fuzzy set).⁶ Evaluating a rule base (à la Mamdani-Assilan) thus becomes trivial and simply amounts to ‘maximum matching,’ that is, searching the maximally supporting rule for each class. Thus, much of the appealing interpolation and approximation properties of fuzzy inference gets lost, and fuzziness only means that rules can be activated to a certain degree. There are, however, alternative methods which combine the predictions of several rules into a classification of the query [58].

A plethora of strategies has been developed for inducing a fuzzy rule base from the data given, and we refrain from a detailed exposition here. Especially important in the field of fuzzy rule learning are hybrid methods that combine FST with other methodologies, notably, evolutionary algorithms and neural networks. For example, evolutionary algorithms are often used in order to optimize (‘tune’) a fuzzy rule base or for searching the space of potential rule bases in a (more or less) systematic way [59]. Quite interesting are also *neuro-fuzzy* methods [60]. For example, one idea is to encode a fuzzy system as a neural network and to apply standard methods (like backpropagation) in order train such a network. This way, neuro-fuzzy systems combine the representational advantages of fuzzy systems with the flexibility and adaptivity of neural networks.

Fuzzy Decision Tree Induction

Fuzzy variants of decision tree induction have been developed for quite a while (e.g. [61, 62]) and seem to remain a topic of interest even today (see [63] for a recent approach and a comprehensive overview

⁶ More generally, a rule consequent can suggest different classes with different degrees of certainty.

of research in this field). In fact, these approaches provide a typical example for the ‘fuzzification’ of standard machine learning methods. In the case of decision trees, it is primarily the ‘crisp’ thresholds used for defining splitting predicates (constraints), such as $\text{size} \leq 181$, at inner nodes that have been criticized: Such thresholds lead to hard decision boundaries in the input space, which means that a slight variation of an attribute (e.g., $\text{size} = 182$ instead of $\text{size} = 181$) can entail a completely different classification of an object (e.g., of a person characterized by size, weight, gender, ...). Moreover, the learning process becomes unstable in the sense that a slight variation of the training examples can change the induced decision tree drastically.

In order to make the decision boundaries ‘soft,’ an obvious idea is to apply fuzzy predicates at the inner nodes of a decision tree, such as $\text{size} \in \text{TALL}$, where TALL is a fuzzy set (rather than an interval). In other words, a fuzzy partition instead of a crisp one is used for the splitting attribute (here size) at an inner node. Since an example can satisfy a fuzzy predicate to a certain degree, the examples are partitioned in a fuzzy manner as well. That is, an object is not assigned to exactly one successor node in a unique way, but perhaps to several successors with a certain degree. For example, a person whose size is 181 cm could be an element of the TALL -group to the degree, say, 0.7 and of the complementary group to the degree 0.3.

The above idea of ‘soft recursive partitioning’ has been realized in different ways. Moreover, the problems entailed by corresponding fuzzy extensions have been investigated. For example, how can information-theoretic splitting measures like information gain, originally defined for ordinary sets of examples, be extended to fuzzy sets of examples [64]? Or, how can a new object be classified by a fuzzy decision tree?

Fuzzy Association Analysis

The use of fuzzy sets in connection with association analysis has been proposed by numerous authors (see [65, 66] for recent overviews), with motivations closely resembling those in the case of rule learning and decision tree induction. Again, by allowing for ‘soft’ rather than crisp boundaries of intervals, fuzzy sets can avoid certain undesirable threshold effects [67], this time concerning the quality measures of association rules (like support and confidence) rather than the classification of objects. Moreover, identifying fuzzy sets with linguistic terms allows for a comprehensible and user-friendly presentation of rules discovered in a database. For example, provided a proper modeling of the fuzzy concepts involved, a rule such as $\{\text{middleAged}, \text{multilingual}\} \rightarrow \{\text{HighIncome}\}$ discovered in an employee database can be presented as ‘middle-aged, multilingual employees typically have high incomes.’

Fuzzy Methods in Instance-Based Learning

Several fuzzy-set-based extensions and generalizations of instance-based learning methods, including nearest neighbor estimation [68] and case-based reasoning [69], have been proposed in literature. This is hardly astonishing, given that the concept of similarity, which lies at the heart of instance-based learning, is also one of the main semantics of fuzzy membership degrees [70, 71]. Among the potential advantages of fuzzy approaches, let us mention the following ones: Firstly, by formalizing case-based inference in terms of fuzzy-set-based approximate reasoning, the former can be considered as a special case of the latter and, hence, becomes amenable to various extensions, notably the combination of case-based and rule-based inference [72]. Secondly, by exploiting the close connection between fuzzy sets and possibility theory, the latter can be used for expressing the uncertainty related to nearest neighbor predictions. In particular, possibility theory is able to represent *partial ignorance*, which is a point of critical importance in instance-based learning [73].

42.4.3 Rule Induction in the Framework of RST

Even though the theory of rough sets can support machine learning and data mining methods of different type [74–77], it seems to be particularly useful for rule induction and indeed has been applied quite extensively for this purpose (e.g., [78–81]). As outlined in Section 42.3.2, the key problem in rule-based classification is to represent each class in terms of a set of rules. Every single rule should be

representative of its class in the sense that it covers as many as possible examples of this class and as few as possible examples of the other classes. Theoretical issues related to this kind of covering strategy can nicely be formalized in terms of concepts from RST, notably those of lower and upper approximation.

In fact, the sets to be approximated in the context of rule learning are the class representatives, that is, the decision classes $D_1 \dots D_c$ induced by the decision variable y . Here, D_i denotes the set of instances in the training set \mathcal{T} with class label λ_i . Given a set of attributes \mathcal{B} , a key question is whether these classes are *B-definable*. If this is the case, it is possible to characterize the training data exactly in terms of a corresponding set of rules. (Note that, in the case of discrete attributes, every input vector $(x^1 \dots x^b)$ can be identified with a rule premise, and several such examples can be merged into more general premise parts.)

More generally, the union of the \mathcal{B} -lower approximations of the D_i is called the \mathcal{B} -positive region of classification, $\text{Pos}_{\mathcal{B}}(y)$. The lower approximation of a class gives rise to *certain* rules, whereas the upper approximation leads to *approximate* rules. More generally, a potential rule can be evaluated in terms of several quality measures such as *strength* (number of covered training examples) and *accuracy* (fraction of correctly classified examples among the covered ones).

As already mentioned before, *predictive models* having good generalization performance usually ought to be as simple as possible. To solve the corresponding problem of inducing a *minimal* model from the training data, that is, an as small as possible set of rules covering the training data, most algorithms based on rough sets refer to techniques which are quite comparable to the general covering strategies outlined in Section 42.3.2; for an overview of such induction methods see, e.g., [79].

Apart from finding minimal models, researchers in the field of RST have also considered the problem to extract an *exhaustive set of rules*, that is, the set of all decision rules satisfying certain quality requirements. Whereas the *minimal rule set* problem is typical of inductive inference in machine learning, the latter task is more akin to data mining and can be seen as a typical pattern discovery problem. In fact, finding an exhaustive set of rules is very similar to association rule mining as discussed in Section 42.3.3; correspondingly, similar algorithms are used for solving these problems [82].

42.4.4 Learning with Granular Examples

In the previous subsections, we have surveyed some applications in ML&DM which are related to granular computing via fuzzy and rough set theory. The aim of this subsection is to give a more concrete example of the use of ideas and concepts from GrC in machine learning. As we mentioned in Section 42.4.1, preprocessing the original data set by compressing it into a smaller set of ‘granular examples’ calls for an extension of a learning method which is applied subsequent to this preprocessing step. In the following, we will consider the simple nearest neighbor classification method as an example and show that, for this particular case, a corresponding extension can be realized in a relatively simple way.

Roughly speaking, the necessary extension of the original NN classifier comes down to replacing the original distance metric, defined as an $\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ mapping, by a distance function defined on $\mathcal{X} \times \mathcal{G}$, where \mathcal{G} denotes the space of granular examples. Thus, the generalized distance measure must be able to compute the distance between a query input x_0 and a granular example $g \in \mathcal{G}$. Given a measure of that kind, NN classification can in principle be realized in the same way as before, namely, by retrieving the k nearest granular examples and combining the class labels associated with these examples into a final prediction, e.g., by means of majority voting.

The RISE (rule induction from a set of examples) algorithm, originally introduced in [83] with the intention to combine case-based and rule-based learning, in principle realizes the above idea. More specifically, the idea of RISE is to unify the concepts of a case and a rule, considering individual examples (x_i, y_i) as maximally specific rules. Assuming an attribute-value representation of inputs (i.e., an input is specified in terms of an assignment of values to a fixed number of m attributes $A_1 \dots A_m$), such a rule is specified as follows:

$$\text{IF } (A_1 = x_i^1) \text{ and } \dots \text{ and } (A_m = x_i^m) \text{ THEN } (Y = y_i).$$

Specific rules can then be generalized (in a minimal way) so as to cover the nearest example of the same class. This is accomplished by weakening some of the conditions in the rule antecedent. Generalizations of this kind are realized in an iterative way as long as the performance of the overall system can be improved.

In the original approach, RISE defines the distance between two input patterns $x_i = (x_i^1 \dots x_i^m)$ and $x_j = (x_j^1 \dots x_j^m)$ as

$$\delta(x_i, x_j) = \frac{1}{m} \sum_{k=1}^m \delta_k(x_i^k, x_j^k),$$

where $\delta_k(x_i^k, x_j^k)$ denotes the Euclidean distance for numerical attributes (normalized to the range [0, 1]) and the following simplified version of the value difference metric for nominal attributes:

$$\delta_k(x_i^k, x_j^k) = \sum_{\ell=1}^c |\mathbb{P}(\lambda_\ell | x_i^k) - \mathbb{P}(\lambda_\ell | x_j^k)|,$$

where c is the number of classes and $\mathbb{P}(\lambda_\ell | x_i^k)$ is the conditional probability of the ℓ th class λ_ℓ given the value x_i^k for the k th attribute. (These probabilities are estimated by the corresponding relative frequencies in the data.) The distance between a regular example and a rule is defined by the minimal distance between that example and a point covered by the rule. For more technical details we refer to [83, 84].

Regarding the classification of new examples, RISE refers to the nearest neighbor principle underlying case-based learning methods: In order to classify a query instance, it simply looks for the rule with the minimal distance (see Figure 42.1 for a simple illustration). If there are several rules having the same (minimal) distance, the one with the highest quality is chosen. The quality of a rule is quantified in terms of the well-known Laplace measure $(p+1)/(p+n+c)$, where p is the number of positive examples covered by the rule (i.e., examples the class of which corresponds to the class associated with the rule), n is the number of negative examples, and c is the number of classes.

From a GrC point of view, the RISE algorithm can be seen as a strategy for creating a set of granular examples from a set of training data T so as to maximize classification accuracy when using these examples in ‘granular’ NN classification as outlined above. Thus, the preprocessing of data (reduction of the number of examples) is not a separate step but is actually integrated in the learning algorithm. RISE is hence comparable to so-called editing strategies for NN classification [85], with the difference that it is not restricted to selecting a subset of regular examples but may instead compress several such examples into a granular example.

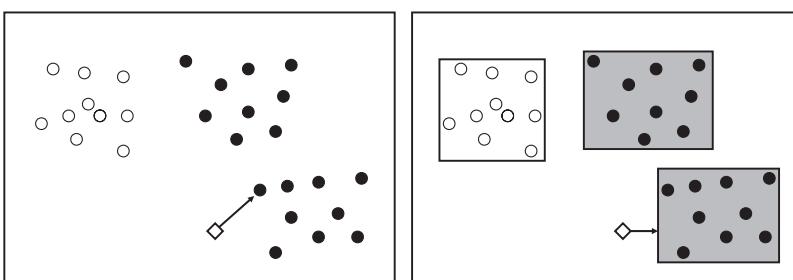


Figure 42.1 Left: Simple NN classification assigns the query (diamond) to the class of the nearest neighbor, which is black. Right: Granular examples (rectangles) are summaries of subsets of original examples, the query is assigned to the class of the closest granular example

42.5 Conclusion

This chapter has revealed the existence of many links between GrC and fields like machine learning and data mining. In fact, GrC has an important role to play in these research areas, since many of the key problems in machine learning and data mining, like the compression of data, the extraction of patterns from data, and the abstraction and generalization beyond given examples, are intimately related to ideas and concepts from GrC. Besides, several of the main GrC methodologies, notably, rough set and fuzzy set theory, have been applied successfully in machine learning and data mining since many years.

Nevertheless, as a relatively recent research paradigm, GrC still has to establish itself as an integral part of machine learning and data mining and to prove that it can contribute to these fields in a substantial way. A key challenge for future work is a unified methodology of ‘granular ML&DM’ which provides a theoretical foundation of machine learning and data mining built on the GrC paradigm as a main conceptual framework.

References

- [1] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In: *Advances in Knowledge Discovery and Data Mining*. MIT Press, Cambridge, MA, 1996, pp. 1–34.
- [2] J. Shawe-Taylor and N. Christianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK, 2004.
- [3] S. Har-Peled, D. Roth, and D. Zimak. Constraint classification: A new approach to multiclass classification. In: *Proceedings 13th International Conference on Algorithmic Learning Theory*, Lübeck, Germany, 2002. Springer, pp. 365–379.
- [4] J. Fürnkranz and E. Hüllermeier. Pairwise preference learning and ranking. In: *Proceedings of ECML–2003, 13th European Conference on Machine Learning*, Cavtat-Dubrovnik, Croatia, September 2003.
- [5] R. Kruse and C. Borgelt. Information mining: Editorial. *Int. J. Approx. Reason.* 32 (2003) 63–65.
- [6] B.V. Dasarathy (ed.) *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, CA, 1991.
- [7] D.R. Wilson. *Advances in Instance-Based Learning Algorithms*. Ph.D. thesis, Department of Computer Science, Brigham Young University, 1997.
- [8] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Commun. ACM* 29(2) (1986) 1213–1228.
- [9] S. Salzberg. A nearest hyperrectangle learning method. *Mach. Learn.* 6 (1991) 251–276.
- [10] D.W. Aha (ed.) *Lazy Learning*. Kluwer Academic, Dordrecht, 1997.
- [11] A.N. Papadopoulos and Y. Manolopoulos. *Nearest Neighbor Search: A Database Perspective*. Series in Computer Science. Springer-Verlag, Berlin, Heidelberg, 2005.
- [12] R.S. Michalski. On the quasi-minimal solution of the general covering problem. In: *Proceedings of 5th International Symposium on Information Processing*, Vol. A3, Bled, Yugoslavia, 1969, pp. 125–128.
- [13] R.S. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The multi-purpose incremental learning system AQ15 and its testing application on three medical domains. In: *Proceedings of 5th National Conference on Artificial Intelligence*, Philadelphia, PA, 1986, pp. 1041–1047.
- [14] P. Clark and T. Niblett. The CN2 induction algorithm. *Mach. Learn.* 3 (1989) 261–283.
- [15] P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In: *Proceedings of the 5th European Working Session of Learning*, Porto, Portugal, 1991, pp. 151–163.
- [16] W.W. Cohen. Fast effective rule induction. In: *Proceedings of 12th International Conference on Machine Learning*, Tahoe City, CA, 1995. Morgan Kaufmann.
- [17] J.R. Quinlan. Learning logical definitions from relations. *Mach. Learn.* 5 (1990) 239–266.
- [18] J. Fürnkranz. Separate-and-conquer rule learning. *Artif. Intell. Rev.* 13(1) (1999) 3–54.
- [19] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.
- [20] D. Dubois, E. Hüllermeier, and H. Prade. A systematic approach to the assessment of fuzzy association rules. *Data Min. Knowl. Discovery* 13(2) (2006) 167–192.
- [21] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In: *Proceedings of the 20th Conference on VLDB*, Santiago, Chile, 1994, pp. 487–499.
- [22] J.S. Park, M.S. Chen, and P.S. Yu. An efficient hash-based algorithm for mining association rules. In: *Proceedings ACM SIGMOD International Conference on Management of Data*, San Jose, CA, 1995, pp. 175–186.

- [23] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In: *VLDB-95, Proceedings of 21th International Conference on Very Large Data Bases*, Zurich, September 1995, pp. 432–444.
- [24] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation. *Data Min. Knowl. Discovery* 8 (2004) 53–87.
- [25] F. Coenen, G. Goulinboune, and P. Leng. Tree structures for mining association rules. *Data Min. Knowl. Discovery* 8 (2004) 25–51.
- [26] F. Coenen, P. Leng, and S. Ahmed. Data structures for association rule mining: T-trees and P-trees. *IEEE Trans. Knowledge and Data Eng.* 16(6) (2004) 774–778.
- [27] J. Hipp, U. Güntzer, and G. Nakhaeizadeh. Algorithms for association rule mining – a general survey and comparison. *Newslett. Spec. Interest Group Knowl. Discovery Data Min.* 2(1) (2000) 58–64.
- [28] B. Goethals and M.J. Zaki. Advances in frequent itemset mining implementations. In: B. Goethals and M.J. Zaki (eds), *Proceedings of IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, Melbourne, FL, 2003.
- [29] B. Goethals and M.J. Zaki. Advances in frequent itemset mining implementations: Report on FIMI'03. *SIGKDD Explor.* 6(1) (2004) 109–117.
- [30] T.M. Mitchell. Version spaces: A candidate elimination approach to rule learning. In: *Proceedings IJCAI-77*, Cambridge, MA, 1977, pp. 305–310.
- [31] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, Heidelberg, Berlin, 1999.
- [32] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer Academic Publishers, Boston, Dordrecht, London, 2005.
- [33] F. Höppner and F. Klawonn. Systems of information granules. In: W. Pedrycz, A. Skowron, and V. Kreinovich (eds), *Handbook on Granular Computing*. John Wiley and Sons, Hoboken, NJ, 2007.
- [34] Z. Pawlak. Rough sets. *Int. J. Comput. Inf. Sci.* 11 (1982) 341–356.
- [35] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, 1992.
- [36] Z. Pawlak and A. Skowron. Rudiments of rough sets. *Inf. Sci.* 177(1) (2007) 3–27.
- [37] Z. Pawlak and A. Skowron. Rough sets: Some extensions. *Inf. Sci.* 177(1) (2007) 28–40.
- [38] Z. Pawlak and A. Skowron. Rough sets and Boolean reasoning. *Inf. Sci.* 177(1) (2007) 41–73.
- [39] L.A. Zadeh. Fuzzy sets. *Inf. Control* 8(3) (1965) 338–353.
- [40] U. Fayyad and K.B. Irani. Multi-interval discretization of continuos attributes as preprocessing for classification learning. In: *Proceedings of the 13th international Joint Conference on Artificial Intelligence*. Morgan Kaufmann, San Fransisco, CA, 1993, pp. 1022–1029.
- [41] L.A. Kurgan and J. Cios. CAIM discretization algorithm. *IEEE Trans. Data Knowl. Eng.* 16(2) (2004) 145–153.
- [42] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In: A. Priditidis and S. Russell (eds), *Machine Learning: Proceedings of the 12th International Conference*. Morgan Kaufmann, San Fransisco, CA, 1995, pp. 194–202.
- [43] E.H. Ruspini. A new approach to clustering. *Inf. Control* 15 (1969) 22–32.
- [44] A. Skowron and R. Swiniarski. Rough set methods in feature selection and recognition. *Pattern Recognit. Lett.* 24(6) (2003) 833–849.
- [45] C. Rauszer. Reducts in information systems. *Fundam. Inf.* 15 (1991) 1–12.
- [46] A. Skowron and C. Rauszer. The discernibility matrices and functions in information systems. In: R. Slowinski (ed.) *Intelligent Decision Support. Handbook of Applications and Advances of Rough Sets Theory*. Kluwer, Dordrecht, 1992, pp. 331–362.
- [47] Q. Shen and A. Chouchoulas. Rough set-based dimensionality reduction for supervised and unsupervised learning. *Int. J. Appl. Math. Comput. Sci.* 11(3) (2001) 583–601.
- [48] R. Jensen and Q. Shen. Fuzzy-rough attribute reduction with application to web categorization. *Fuzzy Sets Syst.* 141(3) (2004) 469–485.
- [49] J. Bazan. A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables. In: L. Polkowski and A. Skowron (eds), *Rough Sets in Knowledge Discovery*. Physica-Verlag, Heidelberg, 1998, pp. 321–365.
- [50] L.A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90(2) (1997) 111–127.
- [51] D. Dubois and H. Prade. Rough fuzzy sets and fuzzy rough sets. *Int. J. Gen. Syst.* 17 (1990) 191–209.
- [52] J. Mill and A. Inoue. Granularization of machine learning. In: *Proceedings of IPMU-06, International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Paris, 2006, pp. 1907–1915.

- [53] A. Laurent. Generating fuzzy summaries: A new approach based on fuzzy multidimensional databases. *Intell. Data Anal. J.* 7(2) (2003) 155–177.
- [54] E. Hüllermeier. Fuzzy sets in machine learning and data mining: Status and prospects. *Fuzzy Sets Syst.* 156(3) (2005) 387–406.
- [55] F. Höppner, F. Klawonn, F. Kruse, and T. Runkler. *Fuzzy Cluster Analysis*. Wiley, Chichester, 1999.
- [56] R. Krishnapuram and J.M. Keller. A possibilistic approach to clustering. *IEEE Trans. Fuzzy Syst.* 1(2) (1993) 98–110.
- [57] A.P. Gasch and M.B. Eisen. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biol.* 3(11) (2002) 1–22.
- [58] O. Cordon, MJ. del Jesus, and F. Herrera. Analyzing the reasoning mechanisms in fuzzy rule based classification systems. *Mathware Soft Comput.* 5 (1998) 321–332.
- [59] O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena. Ten years of genetic fuzzy systems: Current framework and new trends. *Fuzzy Sets Syst.* 141(1) (2004) 5–31.
- [60] D. Nauck, F. Klawonn, and R. Kruse. *Foundations of Neuro-Fuzzy Systems*. Wiley and Sons, Chichester, 1997.
- [61] R. Weber. Fuzzy-ID3: A class of methods for automatic knowledge acquisition. In: *IIZUKA-92, Proceedings of the 2nd International Conference on Fuzzy Logic*, Vol. 1, Iizuka, Japan 1992, pp. 265–268.
- [62] C.Z. Janikow. Fuzzy decision trees: Issues and methods. *IEEE Trans. Syst. Man Cybern.* 28(1) (1998) 1–14.
- [63] C. Olaru and L. Wehenkel. A complete fuzzy decision tree technique. *Fuzzy Sets Syst.* 138(2) (2003).
- [64] T.H. Dang, B. Bouchon-Meunier, and C. Marsala. Measures of information for inductive learning. In: *Proceedings of IPMU-2004*, Perugia, Italy, 2004.
- [65] G. Chen, Q. Wei, E. Kerre, and G. Wets. Overview of fuzzy associations mining. In: *Proceedings of ISIS-2003, 4th International Symposium on Advanced Intelligent Systems*, Jeju, Korea, September 2003.
- [66] M. Delgado, N. Marin, D. Sanchez, and M.A. Vila. Fuzzy association rules: General model and applications. *IEEE Trans. Fuzzy Syst.* 11(2) (2003) 214–225.
- [67] T. Sudkamp. Examples, counterexamples, and measuring fuzzy associations. *Fuzzy Sets Syst.* 149(1) (2005) 57–71.
- [68] J.M. Keller, M.R. Gray, and J.A. Givens. A fuzzy k-nearest neighbor algorithm. *IEEE Trans. Syst. Man Cybern. SMC-15(4)* (1985) 580–584.
- [69] D. Dubois, F. Esteva, P. Garcia, L. Godo, R. Lopez de Mantaras, and H. Prade. Fuzzy set modelling in case-based reasoning. *Int. J. Intell. Syst.* 13(4) (1998) 345–373.
- [70] E.H. Ruspini. Possibility as similarity: The semantics of fuzzy logic. In: P.P. Bonissone, H. Henrion, L.N. Kanal, and J.F. Lemmer (eds), *Uncertainty In Artificial Intelligence 6*. Elsevier Science Publisher, Amsterdam, 1990, pp. 271–280.
- [71] T. Sudkamp. Similarity as a foundation for possibility. In: *Proceedings of 9th IEEE International Conference on Fuzzy Systems*, San Antonio, 2000, pp. 735–740.
- [72] D. Dubois, E. Hüllermeier, and H. Prade. Fuzzy set-based methods in instance-based reasoning. *IEEE Trans. Fuzzy Syst.* 10(3) (2002) 322–332.
- [73] E. Hüllermeier. Possibilistic instance-based learning. *Artif. Intell.* 148(1–2) (2003) 335–383.
- [74] TY. Lin and N. Cercone (eds). *Rough Sets and Data Mining*. Kluwer Academic Publishers, Dordrecht, 1997.
- [75] Y. Yao and N. Zhong. Potential applications of granular computing in knowledge discovery and data mining. In: *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics*, Tokyo, Japan, 1999, pp. 573–580.
- [76] A. Skowron, J. Stepaniuk, J. Peters, and R. Swiniarski. Calculi of approximation spaces. *Fundam. Inf.* 72(1) (2006) 363–378.
- [77] J. Bazan, A. Skowron, and R. Swiniarski. Rough sets and vague concept approximation: From sample approximation to adaptive learning. In: *Transactions on Rough Sets V*, number 4100 in LNCS. Springer-Verlag, Berlin, Heidelberg, 2006, pp. 39–62.
- [78] J. Dong, N. Zhong, and S. Ohsuga. GDT-RS: A probabilistic rough induction approach. In: *Proceedings of DS-98, First International Conference on Discovery Science*, Fukuoka, Japan, 1998, pp. 425–426.
- [79] J. Stefanowski. On rough based approaches to induction of decision rules. In: L. Polkowski and A. Skowron (eds), *Rough Sets and Knowledge Discovery*, Vol. 1. Physika-Verlag, Heidelberg, 1998, pp. 500–529.
- [80] J. Grzymala-Busse and X. Zou. Classification strategies using certain and possible rules. In: L. Polkowski and A. Skowron (eds), *Rough Sets and current Trends in Computing*, number 1424 in LNAI. Springer-Verlag, Berlin, Heidelberg, 1998, pp. 37–44.
- [81] W. Ziarko and N. Shan. A method for computing all maximally general rules in attribute-value systems. *Comput. Intell.* 2 (1993) 2–13.

- [82] D. Delic, H.J. Lenz, and M. Neiling. Improving the quality of association rule mining by means of rough sets. In: J. Kacprzyk, P. Grzegorzewski, and O. Hryniwicz (eds), *Soft Computing in Probability and Statistics*. Springer-Verlag, Berlin, Heidelberg, 2002, pp. 281–288.
- [83] P. Domingos. Unifying instance-based and rule-based induction. *Mach. Learn.* 24 (1996) 141–168.
- [84] P. Domingos. Rule induction and instance-based learning: A unified approach. In: C.S. Mellish (ed.), *Proceedings IJCAI-95, 14th International Joint Conference on Artificial Intelligence*, Montreal, 1995. Morgan Kaufmann, pp. 1226–1232.
- [85] E. McKenna and B. Smyth. Competence-guided editing methods for lazy learning. In: *Proceedings ECAI-2000, 14th European Conference on Artificial Intelligence*, Berlin, 2000, pp. 60–64.

43

On Group Decision Making, Consensus Reaching, Voting, and Voting Paradoxes under Fuzzy Preferences and a Fuzzy Majority: A Survey and a Granulation Perspective

Janusz Kacprzyk, Sławomir Zadrożny, Mario Fedrizzi, and Hannu Nurmi

43.1 Introduction

Decision making is one of the most crucial and omnipresent human activities. Its essence is to find a best alternative (option, variant, ...) from among some feasible (relevant, available, ...) ones. An universal relevance of decision making has clearly triggered an intensive research, in many fields of science, and from many diverse perspectives: behavioral, psychological, cognitive, social, mathematical, economic, etc. This chapter belongs to a formal, mathematical direction aimed at a mathematical formalization of the human rational behavior and how decisions are made.

Decision making in real world usually proceeds under multiple criteria, decision makers, stages, etc. and we consider the case of multiperson decision making, more specifically of a group type, practically from the perspective of social choice and voting, under some fuzzification of preferences and majority. We assume that there is a set of individuals who provide their testimonies assumed to be *preferences* over the set of alternatives. The problem is to find a *solution*, i.e., an alternative (or a set of alternatives) which is best acceptable by the group of individuals as a whole. For a different point of departure, involving choice sets or utility functions, we may refer the interested reader to, e.g., Kim [1], Salles [2], etc.

Group decision making has been plagued since its inception by negative results in that no ‘rational’ choice procedure satisfies all ‘natural,’ or plausible, requirements; by far the best known negative result is the so-called Arrow’s impossibility theorem (cf. Arrow [3] or Kelly [4]), negative results due to Gibbard and Satterthwaite, McKelvey, Schofield, etc. – cf. Nurmi [5]. Their essence is that no matter which group choice procedure is employed, it would satisfy some plausible conditions but not other equally plausible

ones, and this pertains to all possible choice procedures. A promising assumption to overcome these difficulties might be to modify some basic assumptions underlying the group decision making process, and this is also basically assumed here.

An important research direction is here based on the introduction of an *individual* and *social fuzzy preference relation*. Suppose that we have a set of $n \geq 2$ alternatives, $S = \{s_1, \dots, s_n\}$, and a set of $m \geq 2$ individuals, $E = \{1, \dots, m\}$. Then, an individual's $k \in E$ individual fuzzy preference relation in $S \times S$ assigns a value in the unit interval for the preference of one alternative over another.

Normally, there are also some conditions to be satisfied, as, e.g., reflexivity, connectivity, (max-min) transitivity, etc. One should however note that it is not clear which of these ‘natural’ properties of preference relations should be assumed. We will briefly discuss this issue in Section 43.3, but the interested reader should consult, e.g., Salles [2]. Moreover, a deep discussion is given in, e.g., Fodor and Roubens' [6], and also in De Baets *et al.* [7–9].

We assume the individual and social fuzzy preference relations to be defined in $S \times S$, i.e., to each pair of alternatives a strength of preference of one over another is assigned as a value from $[0, 1]$. An important direction is to assume the values of the strength of preferences to belong to some ordered set (normally a set of linguistic values). This implies some non-standard notions of soft preferences, orderings, etc. The best source for information on these and other related topics is Salles [2], and among the new approaches, e.g., Herrera *et al.* [10–15]. We will not follow this direction because not all solutions concepts, properties, etc. dealt with here have been extended using that representation of preferences.

The concept of a *majority* is another basic element of virtually all decision-making models with multiple decision makers because, intuitively, a solution is to be an alternative (or alternatives) best acceptable by the entire group, that is, by (at least!) *most* of its members.

Some of the negative results with group decision making are related to too strict a representation of majority (e.g., at least 50%, at least 2/3, ...). One can try to make that strict concept of majority closer to its real human perception by softening it, and this argument is supported by many real-life statements, for instance in a biological context as (cf. Loewer and Laddaga [16]):

... It can correctly be said that there is a consensus among biologists that Darwinian natural selection is an important cause of evolution though there is currently no consensus concerning Gould's hypothesis of speciation. This means that there is a widespread agreement among biologists concerning the first matter but disagreement concerning the second ...

and it is clear that a rigid majority as, e.g., more than 75% would evidently not reflect the essence of the above statement. Among obvious situations when a strict majority is necessary, political elections are the best example.

A natural manifestations of such a ‘soft’ majority are the so-called *linguistic quantifiers* as, e.g., most, almost all, much more than a half, etc. Such linguistic quantifiers can be, fortunately enough, dealt with by fuzzy-logic-based calculi of linguistically quantified statements as proposed by Zadeh [17]. Moreover, Yager's [18] ordered weighted averaging (OWA) operators can be used for this purpose (cf. Yager and Kacprzyk [19]), and also some other tools as, e.g., the Choquet integral.

In this chapter we will present how fuzzy preference relations and fuzzy majorities can be employed for deriving solution of group decision making and of degrees of consensus. We also mention some approaches to the alleviation of some voting paradoxes.

What is important that our exposition will be in the spirit of granular computing. Namely, one can clearly view the fuzzification of preferences and majority as an example of granulation. Namely, on the one extreme we have traditional binary preferences, with the two (or three) possible values 0 or 1 (supplemented possibly by ‘-’ or ‘0.5’ for ‘don't care or doesn't apply’). On the other extreme we have linguistic preferences, with the intensity of preferences being some linguistic terms from an ordered set (e.g., ‘high,’ ‘medium,’ ‘low’); these terms may be represented semantically by fuzzy sets in $[0, 1]$. In between there is the case of fuzzy preferences in which values of the intensity of preference between alternatives are real numbers from $[0, 1]$.

A crucial question related to the granulation of information is what a granulation employed provides, what new qualities it gives, etc. In our particular case we can subsume this issue as follows. For the one extreme case of granulation, i.e., for traditional binary preference relations, with the intensity of

preferences being either 0 or one 1 (we neglect here for simplicity ‘-’), there are strong theoretical results but the models are often too rigid, and results obtained may often not be intuitively appealing. For the case of granulation assumed in this chapter, i.e., with the traditional fuzzy preference relations and a fuzzy majority, we have more human consistent and more intuitively appealing representation of preferences and majority than in the case of binary preferences and a strict majority. This is maybe not as human consistent and intuitively appealing as the other extreme case of linguistic preferences, but we have here much more strong and constructive results. One can clearly see that the level of a proper granulation is a compromise between many aspects, to be chosen as needed.

To maintain uniformity with many source papers cited, and to make easier for the reader to consult those papers for more detail, the notations should be rather considered as valid for a particular part of the paper rather than for the entire paper. This may imply the use of the same symbols to denote different entities used in different parts of the paper.

43.2 Fuzzy Linguistic Quantifiers and the Ordered Weighted Averaging Operators for a Linguistic Quantifier-Driven Aggregation

An important element of our analysis are fuzzy linguistic quantifiers that are used for the representation of a fuzzy majority, one of key elements of some of our models to be presented.

A fuzzy set A in $X = \{x\}$ will be characterized and equated with its membership function $\mu_A : X \rightarrow [0, 1]$ such that $\mu_A(x) \in [0, 1]$ is the grade of membership of $x \in X$ in A , from full membership to full nonmembership, through all intermediate values. For a finite $X = \{x_1, \dots, x_n\}$, we write $A = \mu_A(x_1)/x_1 + \dots + \mu_A(x_n)/x_n$. Moreover, we denote $a \wedge b = \min(a, b)$ and $a \vee b = \max(a, b)$. Other, more specific notation will be introduced when needed.

A *linguistically quantified statement*, e.g., ‘most experts are convinced,’ may be generally written as

$$Qy's \text{ are } F, \quad (1)$$

where Q is a linguistic quantifier (e.g., most), $Y = \{y\}$ is a set of objects (e.g., experts), and F is a property (e.g., convinced).

Adding a different importance (relevance, competence, ...), B , to the particular y 's (objects), we obtain a *linguistically quantified statement with importance qualification*, generally written as

$$QBy's \text{ are } F, \quad (2)$$

which may be exemplified by ‘most (Q) of the important (B) experts (y 's) are convinced (F).’

The problem is to find the truth of such linguistically quantified statements, i.e., $\text{truth}(Qy's \text{ are } F)$ or $\text{truth}(QBy's \text{ are } F)$ knowing $\text{truth}(y \text{ is } F)$, for each $y \in Y$. One can use different calculi but we will consider Zadeh's [17] and Yager's [18] OWA-operators-based calculi only.

43.2.1 A Fuzzy-Logic-Based Calculus of Linguistically Quantified Statements

In Zadeh's [17] method, a (proportional) fuzzy linguistic quantifier Q is assumed to be a fuzzy set defined in $[0, 1]$ exemplified by $Q = \text{'most'}$

$$\mu_Q(x) = \begin{cases} 1 & \text{for } x \geq 0.8 \\ 2x - 0.6 & \text{for } 0.3 < x < 0.8 \\ 0 & \text{for } x \leq 0.3 \end{cases} \quad (3)$$

to be meant as that if at least 80% of some elements satisfy a property, then *most* of them certainly (to degree 1) satisfy it, when less than 30% of them satisfy it, then *most* of them certainly do not satisfy it (to degree 0), and between 30% and 80% – the more of them satisfy it the higher the degree of satisfaction by *most* of the elements.

Property F is defined as a fuzzy set in Y . For instance, if $Y = \{X, W, Z\}$, is the set of experts and F is ‘convinced,’ then $F = \text{‘convinced’} = 0.1/X + 0.6/W + 0.8/Z$ means that expert X is convinced to degree 0.1, W to degree 0.6 and Z to degree 0.8. If $Y = \{y_1, \dots, y_p\}$, then $\text{truth}(y_i \text{ is } F) = \mu_F(y_i)$, $i = 1, \dots, p$.

Then, we calculate

$$r = \frac{1}{p} \sum_{i=1}^p \mu_F(y_i) \quad (4)$$

$$\text{truth}(Qy\text{'s are } F) = \mu_Q(r). \quad (5)$$

In the case of importance qualification, B is a fuzzy set in Y , and $\mu_B(y_i) \in [0, 1]$ is a degree of importance of y_i ; from 1 for definitely important to 0 for definitely unimportant, through all intermediate values. We rewrite first ‘ $QBy\text{'s are } F$ ’ as ‘ $Q(B \text{ and } F)y\text{'s are } B$ ’ which leads to the following counterparts of (4) and (5):

$$r' = \frac{\sum_{i=1}^p [\mu_B(y_i) \wedge \mu_F(y_i)]}{\sum_{i=1}^p \mu_B(y_i)} \quad (6)$$

$$\text{truth}(QBY\text{'s are } F) = \mu_Q(r') \quad (7)$$

Example 1. Let $Y = \text{‘experts’} = \{X, Y, Z\}$, $F = \text{‘convinced’} = 0.1/X + 0.6/Y + 0.8/Z$, $Q = \text{‘most’}$ be given by (3), $B = \text{‘important’} = 0.2/X + 0.5/Y + 0.6/Z$. Then, $r = 0.5$ and $r' = 0.92$, and $\text{truth}(\text{‘most experts are convinced’}) = 0.4$ and $\text{truth}(\text{‘most of the important experts are convinced’}) = 1$.

The method presented is simple and efficient and has proved to be useful in a multitude of cases, also in this chapter.

Ordered Weighted Averaging Operators

Yager [18] (see also Yager and Kacprzyk’s [19]) has proposed a special class of aggregation operators, called the *ordered weighted averaging* (or OWA, for short) operators, which can simply and uniformly model a large class of fuzzy linguistic quantifiers, and hence of linguistic-quantifier-driven aggregation behaviors.

An OWA operator of dimension p is a mapping $O : [0, 1]^p \rightarrow [0, 1]$ if associated with O is a weighting vector $W = [w_1, \dots, w_p]^T$ such that $w_i \in [0, 1]$, $w_1 + \dots + w_p = 1$, and

$$O(x_1, \dots, x_p) = w_1 b_1 + \dots + w_p b_p, \quad (8)$$

where b_i is the i th largest element among $\{x_1, \dots, x_p\}$. B is called an ordered argument vector if each $b_i \in [0, 1]$, and $j > i$ implies $b_i \geq b_j$, $i = 1, \dots, p$.

Then

$$O(x_1, \dots, x_p) = W^T B \quad (9)$$

Example 2. Let $W^T = [0.2, 0.3, 0.1, 0.4]$ and calculate $F(0.6, 1.0, 0.3, 0.5)$. Thus, $B^T = [1.0, 0.6, 0.5, 0.3]$, and $O(0.6, 1.0, 0.3, 0.5) = W^T B = 0.55$; and $O(0.0, 0.7, 0.1, 0.2) = 0.21$.

It is not obvious how the OWA weights are found from the membership function of a fuzzy linguistic quantifier Q ; an early approach given in Yager [18] is often used:

$$w_k = \mu_Q(k/p) - \mu_Q((k-1)/p) \quad \text{for } k = 1, \dots, p. \quad (10)$$

Some examples of the weights w_i ’s associated with the particular quantifiers are:

- If $w_p = 1$, and $w_i = 0$, for each $i \neq p$, then this corresponds to $Q = \text{‘all’}$;

- If $w_i = 1$ for $i = 1$, and $w_i = 0$, for each $i \neq 1$, then this corresponds to $Q = \text{'at least one'}$;
- If $w_i = 1/p$, for each $i = 1, 2, \dots, p$, then this corresponds to the arithmetic mean, and the intermediate cases as, e.g., a half, most, much more than 75%, a few, almost all, etc., may be obtained by a suitable choice of the w_i 's between the above two extremes.

Thus, we will write

$$\text{truth}(Q\text{'s are } F) = O_Q(\text{truth}(y_i \text{ is } F)) = W^T B. \quad (11)$$

An important problem is the OWA operators with importance qualification. Suppose that we have a vector of data $A = [a_1, \dots, a_n]$, and a vector of importances $V = [v_1, \dots, v_n]$ such that $v_i \in [0, 1]$ is the importance of a_i , $i = 1, \dots, n$, ($v_1 + \dots + v_n \neq 1$, in general), and the OWA weights $W = [w_1, \dots, w_n]^T$ corresponding to Q is determined via (10).

In the popular Yager's [18] approach to be used here, the problem boils down to some redefinition of the OWA's weights w_i into \bar{w}_i . Then, (8) becomes

$$O_V(a_1, \dots, a_n) = \bar{W}^T \cdot B = \sum_{j=1}^n \bar{w}_j b_j \quad (12)$$

We order first the pieces of evidence a_i , $i = 1, \dots, n$, in descending order to obtain B such that b_j is the j th largest element of $\{a_1, \dots, a_n\}$. Next, we denote by u_j the importance of b_j , i.e., of the a_i which is the j th largest; $i, j = 1, \dots, n$. Finally, the new weights \bar{W} are defined as

$$\bar{w}_j = \mu_Q \left(\frac{\sum_{k=1}^j u_k}{\sum_{k=1}^n u_k} \right) - \mu_Q \left(\frac{\sum_{k=1}^{j-1} u_k}{\sum_{k=1}^n u_k} \right). \quad (13)$$

Example 3. Suppose that $A = [a_1, a_2, a_3, a_4] = [0.7, 1, 0.5, 0.6]$, and $V = [u_1, u_2, u_3, u_4] = [1, 0.6, 0.5, 0.9]$. $Q = \text{'most'}$ is given by (3).

Then, $B = [b_1, b_2, b_3, b_4] = [1, 0.7, 0.6, 0.5]$, and $\bar{W} = [0.04, 0.24, 0.41, 0.31]$, and $O_V(A) = \sum_{j=1}^4 \bar{w}_j b_j = 0.067 \cdot 1 + 0.4 \cdot 0.7 + 0.333 \cdot 0.6 + 0.2 \cdot 0.5 = 0.6468$.

We have now the necessary formal means to proceed to our discussion of group decision making and consensus formation models under fuzzy preferences and a fuzzy majority.

Finally, let us mention that OWA-like aggregation operators may be defined in an ordinal setting, i.e., for non-numeric data (which are only ordered), and we will refer the interested reader to, e.g., Delgado *et al.* [20] or Herrera *et al.* [11], and some other of their later papers.

The issues of fuzzy linguistic quantifiers, or maybe more generally of a linguistic quantifier driven aggregation, is very relevant in the context of granulation too in this sense that one can view a coarse granulation as corresponding to an aggregation driven by a fuzzy quantifier (e.g., most, almost all) while a low granulation – to a crisp quantifier like at least 50%, at least 2/3, etc. With the aggregation via the mean or a weighted average being in between.

43.3 Group Decision Making under Fuzzy Preferences and a Fuzzy Majority: General Remarks

Group decision making proceeds in the following setting. We have a set of $n \geq 2$ alternatives, $S = \{s_1, \dots, s_n\}$, and a set of $m \geq 2$ individuals, $E = \{1, \dots, m\}$. Each individual $k \in E$ provides his or her testimony as to the alternatives in S , assumed to be individual fuzzy preference relations defined over S (i.e., in $S \times S$). Fuzzy preference relations are employed to reflect an omnipresent fact that the preferences may not be clear-cut so that conventional non-fuzzy preference relations may not be adequate (see, e.g., many articles in Kacprzyk and Roubens [21] or Kacprzyk *et al.* [22]).

An *individual fuzzy preference relation* of individual k , R_k , is given by its membership function $\mu_{R_k} : S \times S \longrightarrow [0, 1]$ such that

$$\mu_{R_k}(s_i, s_j) = \begin{cases} 1 & \text{if } s_i \text{ is definitely preferred to } s_j \\ c \in (0.5, 1) & \text{if } s_i \text{ is slightly preferred to } s_j \\ 0.5 & \text{in the case of indifference} \\ d \in (0, 0.5) & \text{if } s_j \text{ is slightly preferred to } s_i \\ 0 & \text{if } s_j \text{ is definitely preferred to } s_i. \end{cases} \quad (14)$$

We will also use a special type of an individual fuzzy preference relation, a *fuzzy tournament*, to be defined later on.

If card S is small enough (as assumed here), an individual fuzzy preference relation of individual k , R_k , may conveniently be represented by an $n \times n$ matrix $R_k = [r_{ij}^k]$, such that $r_{ij}^k = \mu_{R_k}(s_i, s_j)$; $i, j = 1, \dots, n$; $k = 1, \dots, m$. R_k is commonly assumed (also here) to be reciprocal in that $r_{ij}^k + r_{ji}^k = 1$; moreover, it is also normally assumed that $r_{ii}^k = 0$, for all i, k ; for a different, more justified convention, cf. García-Lapresta and Llamazares [23]. Notice that we do not mention here other properties of (individual) fuzzy preference relations which are often discussed (cf. Salles [2]) but which will not be relevant to our discussion. Moreover, we will not use here a more sophisticated concept of a fuzzy preference systems proposed by De Baets *et al.* [7–9]; the reasoning is the same.

Two lines of reasoning may be followed here (cf. Kacprzyk [24–29]):

- a direct approach: $\{R_1, \dots, R_m\} \longrightarrow$ solution, in that a solution is derived directly from the set of individual fuzzy preference relations, and
- an indirect approach: $\{R_1, \dots, R_m\} \longrightarrow R \longrightarrow$ solution, in that from the set of individual fuzzy preference relations we form first a social fuzzy preference relation, R , which is then used to find a solution.

A solution is not a clear-cut notion – see, e.g., Nurmi [5, 30–33] for diverse solution concepts. We will only sketch the derivation of some more popular solution concepts to show to the reader not only the essence of the particular solution concept but how a fuzzification may be performed so that other known crisp solution concepts can also be fuzzified. We will show the derivation of some fuzzy cores and minimax sets for the direct approach, and some fuzzy consensus winners for the indirect approach, using fuzzy preference relations and a fuzzy majority represented by a linguistic quantifier as proposed by Kacprzyk [24–29].

First, we will consider the case of fuzzy preferences only and then we will add a fuzzy majority, which is a more interesting case for our purposes. This, as already indicated earlier, reflects which degree of granulation we wish to use.

43.4 Group Decision Making under Fuzzy Preferences

First, we will only assume that we have individual fuzzy preferences and a non-fuzzy majority and present some solution concepts via the direct and indirect approach.

43.4.1 Solutions Based on Individual Fuzzy Preference Relations

We first consider solution concepts that do not require any preference aggregation at all. One of the best known solution concepts is that of a core or a set of undominated alternatives. We denote a nonfuzzy majority by r (e.g., at least 50%).

An alternative $x \in S$ belongs to the *core* iff there is no other alternative $y \in S$ that defeats x by the required majority r .

We can extend this for fuzzy individual preference relations by defining *the fuzzy α -core* as follows (cf. Nurmi [30]):

An alternative $s_i \in S$ belongs to the *fuzzy α -core* S_α iff there exists no other alternative $s_j \in S$ such that $r_{ji} > \alpha$ for at least r individuals.

Notice that if the nonfuzzy core is non-empty, so is S_α for some $\alpha \in (0, 1]$. In other words, $\exists \alpha \in (0, 1]$: $\text{core} \subset S_\alpha$. Moreover, for any two values $\alpha_1, \alpha_2 \in (0, 1]$ such that $\alpha_1 < \alpha_2$, we have $S_{\alpha_1} \subseteq S_{\alpha_2}$.

Intuitively, an alternative is a member of S_α if and only if a sufficient majority of voters does not feel strongly enough against it.

Another non-fuzzy solution concept with much intuitive appeal is a minimax set which, in a non-fuzzy setting, is defined as follows.

For each $x, y \in S$, denote the number of individuals preferring x to y by $n(x, y)$. Then denote $v(x) = \max_y n(y, x)$ and $n* = \min_x v(x)$.

Now, the *minimax set* is

$$Q(n*) = \{x \mid v(x) = n*\}. \quad (15)$$

Thus, $Q(n*)$ contains those alternatives that in pairwise comparison with any other alternative are defeated by no more than $n*$ votes. Obviously, if $n* < m/2$, where m is the number of individuals, then $Q(n*)$ is a singleton and $x \in Q(n*)$ is the core if the simple majority rule (50% + 1) is applied.

Analogously, we can define the *minimax degree set* $Q(\beta)$ as follows. Given $s_i, s_j \in S$ and let, for individuals $k = 1, \dots, m$,

$$v_D^k(x_j) = \max_i r_{ij}$$

and

$$v_D(x_j) = \max_k v_D^k(x_j)$$

then, if $\min_j v_D(x_j) = \beta$:

$$Q(\beta) = \{x_j \mid v_D(x_j) = \beta\} \quad (16)$$

and for some relevant properties of the minimax degree set, cf. Nurmi [30–32].

Another concept that is analogous to the nonfuzzy minimax set is a *minimax opposition set*, $Q(v_f)$. Let n_{ij} be the number of those individuals for whom $r_{ij} > r_{ji}$ and let $v_f(x_j) = \max_i n_{ij}$. Denote by \bar{v}_f the minimum of $v_f(x_j)$ with respect to j , i.e.,

$$\bar{v}_f = \min_j v_f(x_j)$$

Then,

$$Q(v_f) = \{x_j \mid v_f(x_j) = \bar{v}_f\}. \quad (17)$$

But, clearly, $Q(v_f) = Q(n*)$ since $r_{ij} > r_{ji}$ implies that the individual prefers alternative x_i to x_j . Similarly, the preference of x_i over x_j implies that $r_{ij} > r_{ji}$. Consequently, the minimax opposition set does not take into account the intensity of preferences as expressed in the individual preference relation matrices.

A more general solution concept, the *α -minimax set* (cf. Nurmi [30]), $Q^\alpha(v_f^\alpha)$, is defined as follows. Let $n_\alpha(x_i, x_j)$ be the number of individuals for whom $r_{ij} \leq \alpha$ for some value of $\alpha \in [0, 0.5]$. We now define, for each $x_i \in S$, $v_f^\alpha(x_i) = \max_j n_\alpha(x_i, x_j)$ and $\bar{v}_f^\alpha = \min_i v_f^\alpha(x_i)$. Then

$$Q^\alpha(v_f^\alpha) = \{x_i \mid v_f^\alpha(x_i) = \bar{v}_f^\alpha\} \quad (18)$$

and it can be shown that $Q^\alpha(v_f^\alpha) \subseteq Q(n*)$ (cf. Nurmi [30]).

Solutions Based on Fuzzy Tournaments

An important reason for studying fuzzy tournaments is to overcome the difficulties inherent in the use of conventional solution concepts, namely, the fact that the latter tend to produce too large solution sets and are therefore not decisive enough. Another purpose of our discussion is to apply analogues of the nonfuzzy solutions to contexts where the opinions of individuals can be represented by more general constructs than just connected and transitive preference relations (cf., e.g., [21]).

Let us take a look at a few solution concepts of nonfuzzy tournaments, mostly those proposed by Nurmi and Kacprzyk [34].

Given the set of alternatives S , a *tournament* P on S is a complete and asymmetric relation on S .

In the context of group decision making P can be viewed as a strict preference relation. When S is of small cardinality, P can be expressed as a matrix $[p_{ij}]$, $p_{ij} \in \{0, 1\}$ so that $p_{ij} = 1$ if the alternative represented by row i is preferred to that represented by column j , and $p_{ij} = 0$ if the alternative represented by column j is preferred to that represented by row i .

Suppose that each individual has a complete, transitive and asymmetric preference relation over S , and that the number of individuals is odd. Then a tournament can be constructed through pairwise comparisons of alternatives. Alternative s_i is preferred to s_j if and only iff the number of individuals preferring the former to the latter is larger than vice versa.

Perhaps the best-known solution concept of tournaments is the Condorcet winner.

The *Condorcet winner* is an alternative which is preferred to all other alternatives by a majority.

The main problem with this solution concept is that it does not always exist.

The *Copeland winning set* UC_C consists of those alternatives that have the largest number of 1s in their corresponding rows in the tournament matrix.

In other words, the Copeland winners defeat more alternatives than any other alternatives do.

The uncovered set is defined by means of a binary relation of covering. An alternative s_i *covers* another alternative s_j iff s_i defeats s_j and everything that s_j defeats. The *uncovered set* consists of those alternatives that are covered by no alternatives.

The *Banks set* is the set of endpoints of Banks chains. Starting from any alternative s_i a *Banks chain* is constructed as follows. First one looks for an alternative that defeats s_i . Suppose that such an alternative exists and is s_j (if one does not exist, then of course s_i is the Condorcet winner). Next one looks for another alternative that defeats both s_i and s_j , etc. Eventually, no alternative can be found that would defeat all previous ones in the chain starting from s_i . The last alternative which defeats all previous ones is the endpoint of the Banks chain starting from s_i . The Banks set is then the set of all those endpoints.

The following relationships hold between the above-mentioned solutions (cf. [5]):

- All solutions converge to the Condorcet winner when one exists.
- The uncovered set includes the Copeland winning set and the Banks set.
- When S contains less than seven elements, the uncovered set and the Banks set coincide.
- When the cardinality of S exceeds 12, the Banks set and the Copeland winning set may be distinct; however, they both always belong to the uncovered set.

Given a group E of m individuals, a collective fuzzy tournament $F = [r_{ij}]$ can be obtained through pairwise comparisons of alternatives so that

$$r_{ij} = \frac{\text{card}\{k \in E \mid s_i P_k s_j\}}{m},$$

where P_k is a non-fuzzy tournament representing the preferences of individual k .

Let us now define a *strong fuzzy covering relation* $C_S \subset S \times S$ as follows

$$\forall i, j, l \in \{1, \dots, n\} : s_i C_S s_j \Leftrightarrow r_{il} \geq r_{jl} \quad \text{and} \quad r_{ij} > r_{ji}.$$

Clearly, the strong fuzzy covering relation implies the nonfuzzy covering relation, but not vice versa. The set of C_S -undominated alternatives is denoted by UC_S .

Let us first define:

A *weak fuzzy covering relation* $C_W \subset S \times S$ is defined as follows:

$$\begin{aligned} \forall s_i, s_j \in S : \\ s_i C_W s_j \Leftrightarrow r_{ij} > r_{ji} \\ \& \quad \text{card}\{s_l \in S : r_{il} > r_{jl}\} \geq \text{card}\{s_p \in S : r_{jp} > r_{ip}\}. \end{aligned}$$

Obviously, $s_i C_S s_j$ implies $s_i C_W s_j$, but not conversely. Thus, the set of C_W -undominated alternatives, UC_W , is always a subset of UC_S . Moreover, the Copeland winning set is always included in UC_S , but not necessarily in UC_W (see [34]).

If one is looking for a solution that is a plausible subset of the uncovered set, then UC_W is not appropriate since it is possible that UC_C is not always a subset of the uncovered set, let alone the Banks set.

Another solution concept, the α -uncovered set, is based on the individual fuzzy preference tournament matrices. One first defines the fuzzy domination relation D and an α -covering relation $C_\alpha \subseteq S \times S$ as follows:

$s_i D s_j$ iff at least 50% of the individuals prefer s_i to s_j to a degree of at least 0.5.

If $s_i C_\alpha s_j$, then $s_i D s_j$ and $s_i D_\alpha s_k$, for all $s_k \in S$ for which $s_j D_\alpha s_k$.

The α -uncovered set consists of those alternatives that are not α -covered by any other alternative.

An obvious candidate for a plausible solution concept for fuzzy tournaments is an α -uncovered set with the smallest value of α .

Other fuzzy solution concepts analogous to their nonfuzzy counterparts can be defined (see Nurmi and Kacprzyk [34]). For example, the α -Banks set can be constructed by imposing the restriction that the majority of voters prefer the next alternative to the previous one in the Banks chain with intensity of at least α .

43.4.2 Solutions Based on a Social Fuzzy Preference Relation

The derivation of these solution concepts requires first a derivation of a social fuzzy preference relation.

In some early approaches, Bezdek *et al.* [35, 36] discuss the problem of finding the set of undominated alternatives or other stable outcomes given a collective fuzzy preference ordering over the alternative set; see also Nurmi [30].

Here, we follow a different approach, more in line with the reasoning from Nurmi and Kacprzyk [34], and will define a couple of solution concepts with a fuzzy collective (social) preference relation.

The set S_α of α -consensus winners is defined as: $s_i \in S_\alpha$ iff $\forall s_j \neq s_i : r_{ij} \geq \alpha$, with $0.5 < \alpha \leq 1$

Whenever S_α is non-empty, it is a singleton, but it does not always exist. Thus, it may be useful to find other solution concepts that specify a nonempty alternative sets even when S_α is empty. One possible candidate is a straightforward extension of Kramer's minimax set, called a set of minimax consensus winners, by S_M :

Let $\bar{r}_j = \max_i r_{ij}$ and $\bar{r} = \min_j \max_i r_{ij}$. Then s_i belong to S_M , the set of minimax consensus winners if and only if $\bar{r}_i = \bar{r}$.

Clearly, S_M is always nonempty, but not necessarily a singleton. As a solution set it has the same interpretation as Kramer's minimax set: it consists of those alternatives which, when confronted with their toughest competitors, fare best, i.e., win by the largest score (if $\bar{r} \leq 0.5$) or lose by the smallest one (if $\bar{r} > 0.5$).

These solution concepts are based on the social preference relation matrix. Other ones can be obtained in several ways. For instance, one may start from an individual preference relation over a set of alternatives and construct the $[r_{ij}]$ matrix as follows:

$$r_{ij} = \begin{cases} \frac{1}{m} \sum_{k=1}^m a_{ij}^k & \text{for } i \neq j \\ r_{ij} = 0 & \text{for } i = j, \end{cases}$$

where $a_{ij}^k = 1$ if s_i is strictly preferred to s_j by voter k , and $a_{ij}^k = 0$ otherwise.

There is nothing ‘fuzzy’ in the above solutions. As the method of constructing the social preference relation matrix suggests, the starting point can just be the ordinary preferences as well.

To summarize the results obtained in this section, the social fuzzy preference relation may reflect, as its individual counterpart, a proper degree of granulation of preferences from the point of view of expressive power, i.e., an adequate representation, and – on the other hand – it can be a very effective and efficient granulation as in this case many solution concepts can be defined, and their relevant properties can be proved.

43.5 Group Decision Making under Fuzzy Preferences and a Fuzzy Majority

In this section we will consider some solution concepts of group decision making but when we both have fuzzy preference relations and a fuzzy majority. This will constitute a slightly more sophisticated situation with respect to granulation as in addition to fuzzy preferences, whose merits with respects to granulation have already been indicated, we also assume a more coarse granulation of majority. Namely, as opposed to a crisp majority assumed in the previous cases, we assume a fuzzy majority represented as a fuzzy linguistic quantifier.

We will also follow here the direct and indirect approach.

43.5.1 Direct Derivation of a Solution

Using the direct approach, i.e., $\{R_1, \dots, R_m\} \rightarrow$ solution, we will derive two popular solution concepts: fuzzy cores and minimax sets.

Fuzzy Cores

The core, a very popular solution concept, is defined as a set of *undominated alternatives*, i.e., those not defeated in *pairwise comparisons* by a required majority (strict!) $r \leq m$; i.e.,

$$C = \{s_j \in S : \neg \exists s_i \in S \text{ such that } r_{ij}^k > 0.5 \text{ for at least } r \text{ individuals}\} \quad (19)$$

Nurmi [30] has extended the core to the *fuzzy α -core* defined as

$$C_\alpha = \{s_j \in S : \neg \exists s_i \in S \text{ such that } r_{ij}^k > \alpha \geq 0.5 \text{ for at least } r \text{ individuals}\} \quad (20)$$

that is, as a set of alternatives not sufficiently (at least to degree α) defeated by the required (still strict!) majority $r \leq m$.

We can assume that the required majority is fuzzy, e.g., given by a fuzzy linguistic quantifier like *most*. This concept of a fuzzy majority has been proposed by Kacprzyk [24–29], and it has turned out that it can be quite useful and adequate.

To employ a fuzzy majority to extend (fuzzify) the core, we start by denoting

$$h_{ij}^k = \begin{cases} 1 & \text{if } r_{ij}^k < 0.5 \\ 0 & \text{otherwise,} \end{cases} \quad (21)$$

where, if not otherwise specified, $i, j = 1, \dots, n$ and $k = 1, \dots, m$.

Thus, h_{ij}^k just reflects if alternative s_j defeats (in pairwise comparison) alternative s_i ($h_{ij}^k = 1$) or not ($h_{ij}^k = 0$).

Then, we calculate

$$h_j^k = \frac{1}{n-1} \sum_{i=1, i \neq j}^n h_{ij}^k, \quad (22)$$

which is clearly the extent, from 0 to 1, to which individual k is not against alternative s_j , with 0 standing for definitely against to 1 standing for definitely not against, through all intermediate values.

Next, we calculate

$$h_j = \frac{1}{m} \sum_{k=1}^m h_j^k, \quad (23)$$

which expresses to what extent, from 0 to 1 as in the case of (22), *all* the individuals are not against alternative s_j .

And, finally,

$$v_Q^j = \mu_Q(h_j) \quad (24)$$

is to what extent, from 0 to 1 as before, Q (say, most) individuals are not against alternative s_j .

The *fuzzy Q-core* is now defined (cf. Kacprzyk [24–29]) as a fuzzy set

$$C_Q = v_Q^1/s_1 + \cdots + v_Q^n/s_n \quad (25)$$

i.e. as a fuzzy set of alternatives that are not defeated by Q (say, most) individuals.

In the above basic definition of a fuzzy Q -core we do not take into consideration to what degrees those defeats of one alternative by another are. They can be accounted for in a couple of plausible ways.

First and most straightforward is via a threshold in the degree of defeat in (21), for instance, by denoting

$$h_{ij}^k(\alpha) = \begin{cases} 1 & \text{if } r_{ij}^k < \alpha \leq 0.5 \\ 0 & \text{otherwise,} \end{cases} \quad (26)$$

where, again, $i, j = 1, \dots, n$ and $k = 1, \dots, m$. Thus, $h_{ij}^k(\alpha)$ just reflects if alternative s_j sufficiently (i.e., at least to degree $1 - \alpha$) defeats (in pairwise comparison) alternative s_i or not.

We can also explicitly introduce the strength of defeat into (21), for instance, by a function

$$\hat{h}_{ij}^k = \begin{cases} 2(0.5 - r_{ij}^k) & \text{if } r_{ij}^k < 0.5 \\ 0 & \text{otherwise,} \end{cases} \quad (27)$$

where, again, $i, j = 1, \dots, n$ and $k = 1, \dots, m$. Thus, \hat{h}_{ij}^k just reflects how strongly (from 0 to 1) alternative s_j defeats (in pairwise comparison) alternative s_i .

Then, by following the same steps (22)–(25), we can derive an α/Q -fuzzy core and an s/Q -fuzzy core.

Example 4. Suppose that we have four individuals, $k = 1, 2, 3, 4$, whose individual fuzzy preference relations are:

$$R_1 = \begin{array}{c|cccc} & j = 1 & 2 & 3 & 4 \\ \hline i = 1 & 0 & 0.3 & 0.7 & 0.1 \\ 2 & 0.7 & 0 & 0.6 & 0.6 \\ 3 & 0.3 & 0.4 & 0 & 0.2 \\ 4 & 0.9 & 0.4 & 0.8 & 0 \end{array} \quad R_2 = \begin{array}{c|ccccc} & j = 1 & 2 & 3 & 4 \\ \hline i = 1 & 0 & 0.4 & 0.6 & 0.2 \\ 2 & 0.6 & 0 & 0.7 & 0.4 \\ 3 & 0.4 & 0.3 & 0 & 0.1 \\ 4 & 0.8 & 0.6 & 0.9 & 0 \end{array}$$

$$R_3 = \begin{array}{c|cccc} & j=1 & 2 & 3 & 4 \\ \hline i=1 & 0 & 0.5 & 0.7 & 0.1 \\ 2 & 0.5 & 0 & 0.8 & 0.4 \\ 3 & 0.3 & 0.2 & 0 & 0.2 \\ 4 & 1 & 0.6 & 0.8 & 0 \end{array} \quad R_4 = \begin{array}{c|cccc} & j=1 & 2 & 3 & 4 \\ \hline i=1 & 0 & 0.4 & 0.7 & 0.8 \\ 2 & 0.6 & 0 & 0.4 & 0.3 \\ 3 & 0.3 & 0.6 & 0 & 0.1 \\ 4 & 0.7 & 0.7 & 0.9 & 0 \end{array}$$

Suppose now that the fuzzy linguistic quantifier is Q = ‘most’ defined by (3). Then, say,

$$\begin{aligned} C_{\cdot \text{most}} &\cong 0.06/s_1 + 0.56/s_2 + 1/s_4 \\ C_{0.3/\cdot \text{most}} &\cong 0.56/s_4 \\ C_{s/\cdot \text{most}} &\cong 0.36/s_4 \end{aligned}$$

to be meant as follows: in case of $C_{\cdot \text{most}}$, alternative s_1 belongs to the fuzzy Q -core to the extent 0.06, s_2 to the extent 0.56, and s_4 to the extent 1, and analogously for the $C_{0.3/\cdot \text{most}}$ and $C_{s/\cdot \text{most}}$. Notice that though the results obtained for the particular cores are different, for obvious reasons, s_4 is clearly the best choice which is evident if we examine the given individual fuzzy preference relations.

Clearly, the fuzzy-linguistic-quantifier-based aggregation of partial scores in the above definitions of the fuzzy Q -core, α/Q -core and s/Q -core, may be replaced by an OWA-operator-based aggregation given by (10) and (11). This was proposed by Fedrizzi *et al.* [37] and then followed by some other authors. The results obtained by using the OWA operators are similar to those for the usual fuzzy linguistic quantifiers.

Finally, let us notice that the individuals and alternatives may be assigned variable importance (competence) and relevance, respectively, and then the OWA-based aggregation with importance qualification may be used. This will not change however the essence of the fuzzy cores defined above and will not be discussed here for lack of space.

Minimax Sets

Another intuitively justified solution concept may be the minimax (opposition) set which may be defined for our purposes as follows.

Let $w(s_i, s_j) \in \{1, 2, \dots, m\}$ be the number of individuals who prefer alternative s_j to alternative s_i , i.e. for whom $r_{ij}^k < 0.5$. If now

$$v(s_i) = \max_{j=1, \dots, n} w(s_i, s_j) \quad (28)$$

and

$$v^* = \min_{i=1, \dots, n} v(s_i) \quad (29)$$

then a *minimax set* is defined as

$$M(v^*) = \{s_i \in S : v(s_i) = v^*\}, \quad (30)$$

i.e., as a (nonfuzzy) set of alternatives which in pairwise comparisons with any other alternative are defeated by no more than v^* individuals, hence by the least number of individuals.

Nurmi [30] extended the minimax set, similarly in spirit of his extension of the core (20), to the α -minimax set as follows. Let $w_\alpha(s_i, s_j) \in \{1, 2, \dots, m\}$ be the number of individuals who prefer s_j to s_i at least to degree $1 - \alpha$, i.e. for whom $r_{ij}^k < \alpha \leq 0.5$.

If now

$$v_\alpha(s_i) = \max_{j=1, \dots, n} w_\alpha(s_i, s_j) \quad (31)$$

and

$$v_\alpha^* = \min_{i=1,\dots,n} v_\alpha(s_i) \quad (32)$$

then an α -minimax set is defined as

$$M_\alpha(v_\alpha^*) = \{s_i \in S : v_\alpha(s_i) = v_\alpha^*\} \quad (33)$$

i.e., as a (nonfuzzy) set of alternatives which in pairwise comparisons with any other alternative are defeated (at least to degree $1 - \alpha$) by no more than v^* individuals, hence by the least number of individuals.

A fuzzy majority was introduced into the above definitions of minimax sets by Kacprzyk [24–28] as follows.

We start with (21), i.e.,

$$h_{ij}^k = \begin{cases} 1 & \text{if } r_{ij}^k < 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (34)$$

and

$$h_i^k = \frac{1}{n-1} \sum_{j=1, j \neq i}^n h_{ij}^k \quad (35)$$

is the extent, between 0 and 1, to which individual k is against alternative s_i .

Then

$$h_i = \frac{1}{m} \sum_{k=1}^m h_i^k \quad (36)$$

is the extent, between 0 and 1, to which all the individuals are against alternative s_i .

Next

$$t_i^Q = \mu_Q(h_i) \quad (37)$$

is the extent, from 0 to 1, to which Q (say, most) individuals are against alternative s_i , and

$$t_Q^* = \min_{i=1,\dots,n} t_i^Q \quad (38)$$

is the least defeat of any alternative by Q individuals.

Finally, a Q -minimax set is

$$M_Q(t_Q^*) = \{s_i \in S : t_i^Q = t_Q^*\}. \quad (39)$$

And analogously as for the α/Q -core and s/Q -core, we can explicitly introduce the degree of defeat $\alpha < 0.5$ and s into the definition of the Q -minimax set.

Example 5. For the same four individual fuzzy preference relations R_1, \dots, R_4 as in Example 4, we obtain, for instance,

$$\begin{aligned} M_{\cdot \text{most}}(0) &= \{s_4\} \\ M_{0.3/\cdot \text{most}}(0) &= \{s_1, s_2, s_4\} \\ M_{s/\cdot \text{most}} &= \{s_1, s_2, s_4\}. \end{aligned}$$

The OWA-based aggregation can also be employed for the derivation of fuzzy minimax sets given above. And, again, the results obtained by using the OWA-based aggregation are similar to those obtained by directly employing Zadeh's [17] calculus of linguistically quantified statements.

43.5.2 Indirect Derivation of a Solution – The Consensus Winner

Now we follow the indirect approach: $\{R_1, \dots, R_m\} \longrightarrow R \longrightarrow \text{solution}$.

It is easy to notice that the above direct derivation scheme involves in fact two problems:

- how to find a social fuzzy preference relation from the individual fuzzy preference relations, i.e.,

$$\{R_1, \dots, R_m\} \longrightarrow R;$$

- how to find a solution from the social fuzzy preference relation, i.e.,

$$R \longrightarrow \text{solution}.$$

We will not deal in more detail with the first step, i.e. $\{R_1, \dots, R_m\} \longrightarrow R$, and assume a (most) straightforward alternative that the social fuzzy preference relation $R = [r_{ij}]$ is given by

$$r_{ij} = \begin{cases} \frac{1}{m} \sum_{k=1}^m a_{ij}^k & \text{if } i \neq j \\ 0 & \text{otherwise,} \end{cases} \quad (40)$$

where

$$a_{ij}^k = \begin{cases} 1 & \text{if } r_{ij}^k > 0.5 \\ 0 & \text{otherwise.} \end{cases} \quad (41)$$

Notice that R obtained via (40) need not be reciprocal, i.e. $r_{ij} \neq 1 - r_{ji}$, but it can be shown that $r_{ij} \leq 1 - r_{ji}$, for each $i, j = 1, \dots, n$.

We will discuss now the second step, i.e., $R \longrightarrow \text{solution}$, i.e., how to determine a solution from a social fuzzy preference relation.

A solution concept of much intuitive appeal is here the consensus winner (cf. Nurmi [30]) which will be extended under a social fuzzy preference relation and a fuzzy majority.

We start with

$$g_{ij} = \begin{cases} 1 & \text{if } r_{ij} > 0.5 \\ 0 & \text{otherwise,} \end{cases} \quad (42)$$

which expresses whether alternative s_i defeats (in the whole group's opinion!) alternative s_j or not.

Next

$$g_i = \frac{1}{n-1} \sum_{j=1, j \neq i}^n g_{ij} \quad (43)$$

which is a mean degree to which alternative s_i is preferred, by the whole group, over all the other alternatives.

Then

$$z_Q^i = \mu_Q(g_i) \quad (44)$$

is the extent to which alternative s_i is preferred, by the whole group, over Q (e.g., most) other alternatives.

Finally, we define a *fuzzy Q-consensus winner* as

$$W_Q = z_Q^1/s_1 + \cdots + z_Q^n/s_n \quad (45)$$

i.e. as a fuzzy set of alternatives that are preferred, by the whole group, over Q other alternatives.

And analogously as in the case of the core, we can introduce a threshold $\alpha \geq 0.5$ and s into (42) and obtain a *fuzzy α/Q -consensus winner* and a *fuzzy s/Q -consensus winner*, respectively.

Example 6. For the same individual fuzzy preference relations as in Example 4, and using (40) and (41), we obtain the following social fuzzy preference relation

	$j = 1$	2	3	4
$i = 1$	0	0	1	0.25
$R =$				
2	0.75	0	0.75	0.25
3	0	0.25	0	0
4	1	0.75	1	0

If now the fuzzy majority is given by Q = ‘most’ defined by (3) and $\alpha = 0.8$, then we obtain

$$\begin{aligned} W_{\text{most}} &= \frac{1}{15}/s_1 + \frac{11}{15}/s_2 + 1/s_4 \\ W_{0.8/\text{most}} &= \frac{1}{15}/s_1 + \frac{11}{15}/s_4 \\ W_{s/\text{most}} &= \frac{1}{15}/s_1 + \frac{1}{15}/s_2 + 1/s_4, \end{aligned}$$

which is to be read similarly as for the fuzzy cores in Example 4. Notice that here once again alternative s_4 is clearly the best choice which is obvious by examining the social fuzzy preference relation.

One can also use here an OWA-based aggregation defined by (10) and (11) as proposed by Fedrizzi, Kacprzyk and Nurmi [37] and Kacprzyk, Fedrizzi and Nurmi [38].

This concludes our brief exposition of how to employ fuzzy linguistic quantifiers to model the fuzzy majority in group decision making, and define some more popular solution concepts. For some other solution concepts, see, e.g., Nurmi [30] or Kacprzyk [26], while for those based on fuzzy tournaments, e.g., Nurmi and Kacprzyk [34].

We will finish this section with a remark that in a number of recent papers by Kacprzyk and Zadrożny [39, 40] it has been shown that the concept of Kacprzyk’s [24, 25] fuzzy Q -core can be a general (prototypical) choice function in group decision making and voting, for instance, those of a ‘consensus solution,’ Borda’s rule, the minimax degree set, the plurality voting, the qualified plurality voting, the approval voting-like, the ‘consensus + approval voting.’ Condorcet’s rule, the Pareto rule, Copeland’s rule, Nurmi’s minimax set, Kacprzyk’s Q -minimax, the Condorcet looser, the Pareto inferior alternatives, etc. This result, as interesting as it is, is however beyond the scope of this chapter.

Finally, notice that the remarks on the granulation given in the previous section hold here too. Namely, the tools presented, i.e., fuzzy preference relations and a fuzzy majority, may represent a proper way of granulation being a good compromise between the expressive power and ease of solution.

43.6 Degrees of Consensus under Fuzzy Preferences and a Fuzzy Majority

Usually, one can find a solution concept even in a situation when the (fuzzy) preference relations of the particular individuals differ a lot; i.e., the group is far from consensus. However, the quality and usefulness of such solutions is low. A good procedure may be to try to bring first the group closer to consensus and then to try to find solutions. This calls for a degree of consensus.

Here we will show how to use fuzzy linguistic quantifiers as representations of a fuzzy majority to define a degree of consensus as proposed in Kacprzyk [28] and then advanced in Kacprzyk and Fedrizzi [41, 42] and Kacprzyk *et al.* [38, 43] (see also Kacprzyk *et al.* [22, 44] and Zadrożny [45].) This degree is meant to overcome some ‘rigidness’ of the conventional concept of consensus in which (full) consensus occurs only when ‘all the individuals agree as to all the issues.’ This may often be counterintuitive, and not consistent with a real human perception of the very essence of consensus (see, e.g., the citation from a biological context given in the beginning of the chapter).

The new degree of consensus proposed can be therefore equal to 1, which stands for full consensus, when, say, ‘most of the individuals agree as to almost all (of the relevant) issues (alternatives, options)’.

Our point of departure is again a set of individual fuzzy preference relations which are meant analogously as in Section 43.3 [see, e.g., (14)]. The degree of consensus is derived in three steps:

- First, for each pair of individuals we derive a degree of agreement as to their preferences between *all* the pairs of alternatives.
- Second, we aggregate these degrees to obtain a degree of agreement of each pair of individuals as to their preferences between Q_1 (a linguistic quantifier as, e.g., ‘most,’ ‘almost all,’ ‘much more than 50%;’ . . .) pairs of relevant alternatives.
- Third, we aggregate these degrees to obtain a degree of agreement of Q_2 (a linguistic quantifier similar to Q_1) pairs of important individuals as to their preferences between Q_1 pairs of relevant alternatives, and this is meant to be the *degree of consensus* sought.

The above derivation process of a degree of consensus may be formalized by using Zadeh’s [17] calculus of linguistically quantified statements and Yager’s [18] OWA-based aggregation.

We start with the degree of a strict agreement between individuals k_1 and k_2 as to their preferences between alternatives s_i and s_j

$$v_{ij}(k_1, k_2) = \begin{cases} 1 & \text{if } r_{ij}^{k_1} = r_{ij}^{k_2} \\ 0 & \text{otherwise,} \end{cases} \quad (46)$$

where here and later on in this section, if not otherwise specified, $k_1 = 1, \dots, m-1; k_2 = k_1+1, \dots, m$; $i = 1, \dots, n-1; j = i+1, \dots, n$.

The relevance of alternatives is assumed to be given as a fuzzy set defined in the set of alternatives S such that $\mu_B(s_i) \in [0, 1]$ is a *degree of relevance* of alternative s_i , from 0 for fully irrelevant to 1 for fully relevant, through all intermediate values.

The relevance of a pair of alternatives, $(s_i, s_j) \in S \times S$, may be defined, say, as

$$b_{ij}^B = \frac{1}{2}[\mu_B(s_i) + \mu_B(s_j)], \quad (47)$$

which is clearly the most straightforward option; evidently, $b_{ij}^B = b_{ji}^B$, and b_{ii}^B do not matter; for each i, j .

And analogously, the *importance of individuals*, I , is defined as a fuzzy set in the set of individuals such that $\mu_I(k) \in [0, 1]$ is a *degree of importance* of individual k , from 0 for fully unimportant to 1 for fully important, through all intermediate values.

Then, the importance of a pair of individuals, (k_1, k_2) , b_{k_1, k_2}^I , may be defined in various ways, e.g., analogously as (47); i.e.,

$$b_{k_1, k_2}^I = \frac{1}{2}[\mu_I(k_1) + \mu_I(k_2)]. \quad (48)$$

The degree of agreement between individuals k_1 and k_2 as to their preferences between *all* the relevant pairs of alternatives is [cf. (6)]

$$v_B(k_1, k_2) = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n [v_{ij}(k_1, k_2) \wedge b_{ij}^B]}{\sum_{i=1}^{n-1} \sum_{j=i+1}^n b_{ij}^B}. \quad (49)$$

The degree of agreement between individuals k_1 and k_2 as to their preferences between Q_1 relevant pairs of alternatives is

$$v_{Q_1}^B(k_1, k_2) = \mu_{Q_1}[v_B(k_1, k_2)]. \quad (50)$$

In turn, the degree of agreement of *all* the pairs of important individuals as to their preferences between Q_1 pairs of relevant alternatives is

$$v_{Q_1}^{I,B} = \frac{2}{m(m-1)} \frac{\sum_{k_1=1}^{m-1} \sum_{k_2=k_1+1}^m [v_{Q_1}^B(k_1, k_2) \wedge b_{k_1, k_2}^I]}{\sum_{k_1=1}^{m-1} \sum_{k_2=k_1+1}^m b_{k_1, k_2}^I} \quad (51)$$

and, finally, the degree of agreement of Q_2 pairs of important individuals as to their preferences between Q_1 pairs of relevant alternatives, called the *degree of Q1/Q2/I/B-consensus*, is

$$con(Q_1, Q_2, I, B) = \mu_{Q_2}(v_{Q_1}^{I,B}). \quad (52)$$

Since the strict agreement (46) may be viewed too rigid, we can use the degree of sufficient agreement (at least to degree $\alpha \in (0, 1]$) of individuals k_1 and k_2 as to their preferences between alternatives s_i and s_j , defined by

$$v_{ij}^\alpha(k_1, k_2) = \begin{cases} 1 & \text{if } |r_{ij}^{k_1} - r_{ij}^{k_2}| \leq 1 - \alpha \leq 1 \\ 0 & \text{otherwise,} \end{cases} \quad (53)$$

where, $k_1 = 1, \dots, m-1$; $k_2 = k_1 + 1, \dots, m$; $i = 1, \dots, n-1$; $j = i + 1, \dots, n$.

Then, following the steps (46)–(52), we obtain the degree of sufficient (at least to degree α) agreement of Q_2 pairs of important individuals as to their preferences between Q_1 relevant pairs of alternatives, called a *degree of α /Q1/Q2/I/B-consensus*, as

$$con^\alpha(Q_1, Q_2, I, B) = \mu_{Q_2}(v_{Q_1}^{I,B,\alpha}). \quad (54)$$

We can also explicitly introduce the strength of agreement into (46) and analogously define the degree of strong agreement of individuals k_1 and k_2 as to their preferences between alternatives s_i and s_j , e.g., as

$$v_{ij}^s(k_1, k_2) = s(|r_{ij}^{k_1} - r_{ij}^{k_2}|), \quad (55)$$

where $s : [0, 1] \rightarrow [0, 1]$ is some function representing the degree of strong agreement as, e.g.,

$$s(x) = \begin{cases} 1 & \text{for } x \leq 0.05 \\ -10x + 1.5 & \text{for } 0.05 < x < 0.15 \\ 0 & \text{for } x \geq 0.15 \end{cases} \quad (56)$$

such that $x' < x' \implies s(x') \geq s(x'')$, for each $x', x'' \in [0, 1]$. And there is such an $x \in [0, 1]$ that $s(x) = 1$. And then, following the steps (46)–(52), we obtain finally the degree of agreement of Q_2 pairs of important individuals as to their preferences between Q_1 relevant pairs of alternatives, called a *degree of s /Q1/Q2/I/B-consensus*, as

$$con^s(Q_1, Q_2, I, B) = \mu_{Q_2}(v_{Q_1}^{I,B,s}). \quad (57)$$

Example 7. Suppose that $n = m = 3$, $Q_1 = Q_2 = \text{'most'}$ are given by (3), $\alpha = 0.9$, $s(x)$ is defined by (56), and the individual preference relations are

$$R_1 = [r_{ij}^1] = \begin{array}{c|ccc} & j = 1 & 2 & 3 \\ \hline i = 1 & 0 & 0.1 & 0.6 \\ 2 & 0.9 & 0 & 0.7 \\ 3 & 0.4 & 0.3 & 0 \end{array} \quad R_2 = [r_{ij}^2] = \begin{array}{c|ccc} & j = 1 & 2 & 3 \\ \hline i = 1 & 0 & 0.1 & 0.7 \\ 2 & 0.9 & 0 & 0.7 \\ 3 & 0.3 & 0.3 & 0 \end{array}$$

$$R_3 = [r_{ij}^3] = \begin{array}{c|ccc} & j = 1 & 2 & 3 \\ \hline i = 1 & 0 & 0.2 & 0.6 \\ 2 & 0.8 & 0 & 0.7 \\ 3 & 0.4 & 0.3 & 0 \end{array}.$$

If we assume the relevance of the alternatives to be $B = \{b_i^B / s_i\} = 1/s_1 + 0.6/s_2 + 0.2/s_3$, the importance of the individuals to be $I = \{b_k^I / k\} = 0.8/1 + 1/2 + 0.4/3$, then we obtain the following degrees of consensus:

$$\begin{aligned} con(\text{'most'}, \text{'most'}, I, B) &\cong 0.35 \\ con^{0.9}(\text{'most'}, \text{'most'}, I, B) &\cong 0.06 \\ con^s(\text{'most'}, \text{'most'}, I, B) &\cong 0.06. \end{aligned}$$

And, similarly as for the group decision making solutions shown in Section 43.3, the aggregation via Zadeh's [17] calculus of linguistically quantified propositions employed above may be replaced by the OWA based aggregation given by (10) and (11). The procedure is analogous as that presented in Section 43.3 and will not be repeated here.

For more information on these degrees of consensus, see, e.g., works by Kacprzyk, Fedrizzi, Nurmi, and Zadrożny [28, 29, 37, 41–47], etc.

Degree of consensus of a group of individuals under fuzzy preference relations and a fuzzy majority is an important concept that can be used to monitor how far the group is from consensus, and then may help bring it close enough so that the solution concepts can give meaningful results. Clearly, all former remarks on issues related to granulation are valid in that the use of fuzzy preference relations and a fuzzy majority can provide a good compromise between the expressive power and ease of derivation.

43.7 Remarks on Some Voting Paradoxes and Their Alleviation

Voting paradoxes are an interesting and very relevant topic that has a considerable theoretical and practical relevance. We will give here just some simple examples of a couple of better known voting paradoxes and indicate some possibilities of how to alleviate them by using mainly some elements of fuzzy preferences. The Section is based on the works by Nurmi [48, 49], and Nurmi and Kacprzyk [50].

Therefore, our analysis can be viewed as a justification that a natural and human-consistent granulation of preferences via fuzzy preference relations can give qualitatively new results and help alleviate some known serious problems which occur when traditional approaches are used.

Table 43.1 presents an instance of Condorcet's paradox where there are three voter groups of equal size having preferences over alternatives A , B , and C as indicated by the rank order shown below each group. In fact, the groups need not be of equal size. What is essential for the paradox is that any two of them constitute a majority. Clearly, a social (collective) preference relation formed on the basis of pairwise comparisons and using majority rule, results in a cycle: A is preferred to B , B is preferred to C and C is preferred to A .

Table 43.1 Condorcet's paradox

<i>Group I</i>	<i>Group II</i>	<i>Group III</i>
<i>A</i>	<i>B</i>	<i>C</i>
<i>B</i>	<i>C</i>	<i>A</i>
<i>C</i>	<i>A</i>	<i>B</i>

An instance of Borda's paradox, in turn, is given in Table 43.2, where alternative *A* would win by a plurality of votes and, yet, both *B* and *C* would beat *A* in pairwise comparisons.

A common feature in these classic paradoxes is an incompatibility of several intuitively plausible requirements regarding social choices. In the case of Condorcet's paradox the result obtained by using the majority rule on a set of complete and transitive preferences is intransitive. In the case of Borda's paradox, the winner in the plurality sense is different from the winner in another sense, i.e., in the sense that requires the winner to beat all the other alternatives in pairwise comparisons.

Let us try to solve the above paradoxes using some fuzzy tools. The solutions presented are very much in the spirit of Sen's idea of broadening the amount of information about individuals. In particular, we shall take our point of departure in the notion of fuzzy individual preference relation. We consider the set *E* of individuals and the set *S* of decision alternatives. Each individual $i \in E$ is assumed to provide a fuzzy preference relation $R_i(x, y)$ over *S*. For each $x, y \in S$ the value $R_i(x, y)$ indicates the degree in which x is preferred to y by i with 1 indicating the strongest preference of x to y , 0.5 being the indifference between the two, and 0 standing for the strongest preference of y to x . Obviously, the assumption that the voters be endowed with fuzzy preference relations is precisely the kind of broadening of the information about individuals that Sen discusses.

Some properties of fuzzy preference relations are defined below (cf. [51, 52]):

Connectedness. A fuzzy preference relation *R* is connected if an only if $R(x, y) + R(y, x) \geq 1$, for each $x, y \in S$.

Reflexivity. A fuzzy preference relation *R* is reflexive if an only if $R(x, x) = 1$, for each $x \in S$.

Max-min transitivity. A fuzzy connected and reflexive relation *R* is max-min transitive if and only if $R(x, z) \geq \min[R(x, y), R(y, z)]$, for each $x, y, z \in S$.

For the case of Condorcet's paradox, given the broadening of information concerning voter preferences represented by fuzzy preference relations, we can solve it very much in the spirit of its 'father,' Marquis de Condorcet (cf. Nurmi [49]). A way out of cyclical collective preferences is to look at the sizes of majorities supporting various collective preferences. For example, if the number of voters preferring *a* to *b* is 5 out of 9, while that of voters preferring *b* to *c* is 7 out of 9, then, according to Condorcet, the latter preference is stronger than the former. By cutting the cycle of collective majority preferences at its weakest link, one ends up with a complete and transitive relation. Clearly, with nonfuzzy preference relation this method works only in cases where not all of the majorities supporting various links in the cycle are of same size. With fuzzy preferences one can form a social (collective) preference between any x and $y \in S$ using a variation of the average rule (cf. Intrilligator [53]), i.e.

$$R(x, y) = \frac{\sum_i R_i(x, y)}{m}, \quad (58)$$

Table 43.2 Borda's paradox

<i>Voters 1–4</i>	<i>Voters 5–7</i>	<i>Voters 8, 9</i>
<i>A</i>	<i>B</i>	<i>C</i>
<i>B</i>	<i>C</i>	<i>B</i>
<i>C</i>	<i>A</i>	<i>A</i>

where $R(x, y)$ is the degree of a social (collective) fuzzy preference of x over y .

Now, supposing that a preference cycle is formed on the basis of collective fuzzy preferences, one could simply ignore the link with the weakest degree of preference and thus possibly end up with a ranking. In general one can proceed by eliminating weakest links in collective preference cycles until there is a ranking.

The above method of successive elimination of the weakest links in preference cycles works with fuzzy and non-fuzzy preferences. When individual preferences are fuzzy, each voter is assumed to report his or her preferences so that the following matrix can be formed:

$$R_i = \begin{pmatrix} - & r_{12} & \dots & r_{1n} \\ r_{21} & - & \dots & r_{2n} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & - \end{pmatrix}, \quad (59)$$

where r_{ij} indicates the degree to which the individual prefers the i th alternative to the j th one. By averaging over the voters we obtain:

$$\bar{R} = \begin{pmatrix} - & \bar{r}_{12} & \dots & \bar{r}_{1n} \\ \bar{r}_{21} & - & \dots & \bar{r}_{2n} \\ \dots & \dots & \dots & \dots \\ \bar{r}_{n1} & \bar{r}_{n2} & \dots & - \end{pmatrix}. \quad (60)$$

Apart from the successive elimination method one can use another straightforward method to resolve Condorcet's paradox, once the \bar{R} -matrix is given. One can proceed as follows. One first computes the row sums of the matrix:

$$\bar{r}_i = \sum_j \bar{r}_{ij}. \quad (61)$$

These represent the total fuzzy preference weight assigned to the i th alternative in all pairwise preference comparisons, when the weight in each comparison is the average fuzzy preference value. Let now

$$p_i = \frac{\bar{r}_i}{\sum_i \bar{r}_i}. \quad (62)$$

Clearly $p_i \geq 0$ and $\sum_i p_i = 1$. Thus, p_i has the natural interpretation of a choice probability. An obvious way to utilize this is to form the social (collective) preference ordering on the basis of these choice probabilities. The result is necessarily a complete and transitive relation. Hence we can use the information broadening provided by fuzzy preferences to solve Condorcet's paradox.

For illustration, consider the example of Table 43.1 again and assume that each group consists of just one voter. Assume, furthermore, that the fuzzy preferences underlying the preference rankings are as in Table 43.3.

Table 43.3 Fuzzy Condorcet's paradox

Voter 1			Voter 2			Voter 3					
A	B	C	A	B	C	A	B	C			
A	—	.6	.8	A	—	.9	.3	A	—	.6	.3
B	.4	—	.6	B	.1	—	.7	B	.4	—	.1
C	.2	.4	—	C	.7	.3	—	C	.7	.9	—

Table 43.4 A fuzzy Borda's paradox

	4 Voters			3 Voters			2 Voters		
	A	B	C	A	B	C	A	B	C
A	—	.6	.8	A	—	.9	.3	A	—
B	.4	—	.6	B	.1	—	.7	B	.8
C	.2	.4	—	C	.7	.3	—	C	.9

The \bar{R} matrix is now

$$\bar{R} = \begin{pmatrix} - & .7 & .5 \\ .3 & - & .5 \\ .5 & .5 & - \end{pmatrix}.$$

Now, $P_A = 0.4$, $P_B = 0.3$, $P_C = 0.3$.

Obviously, the solution is based on somewhat different fuzzy preference relations over the three alternatives. Should the preference relations be identical, we would necessarily end up with the identical choice probabilities.

With fuzzy individual preference relations we can also resolve Borda's paradox. To do that, we simply apply the same procedure as in the resolution of Condorcet's paradox.

Let us take a look at a fuzzy Borda's paradox for illustration. Assume that the fuzzy preferences underlying Table 43.2 are those indicated in Table 43.4.

The matrix of average preference degrees is then the following:

$$\bar{R} = \begin{pmatrix} - & .6 & .5 \\ .4 & - & .6 \\ .5 & .4 & - \end{pmatrix}$$

The choice probabilities of A , B , and C are, therefore, 0.37, 0.33, and 0.30. We see that the choice probability of A is the largest. In a sense, then, the method does not solve Borda's paradox in the same way as the Borda count does since the plurality method ends up also with A being chosen instead of the Condorcet winner B . Note, however, that fuzzy preference relations give a richer picture of voter preferences than the ordinary preference rankings. In particular, A is strongly preferred to B and C by both the 4 and 3 voter groups. Hence, it is to be expected that its choice probability is the largest.

For additional information on voting paradoxes and some ways to solve them using fuzzy logic, we refer the reader to Nurmi and Kacprzyk [50].

Notice in our analysis of how to alleviate some more popular voting paradoxes using the granulation of preferences via fuzzy preference relations we have indicated many times merits of this simple solution. Once again, providing an appropriate expressive power, it is very constructive in the sense that it helps alleviate a serious difficulty, i.e., voting paradoxes.

43.8 Concluding Remarks

In this chapter we have briefly presented the use of fuzzy preference relations and fuzzy majorities in the derivation of group decision making, and voting solution concepts and degrees of consensus. We also briefly show how fuzzy preference relations and a fuzzy majority can help alleviate difficulties related to negative results in group decision making and voting paradoxes.

Our analysis was performed from the point of view of what impact a proper granulation of preferences and majority can have. More specifically, we have showed that by using traditional fuzzy preference relations and a fuzzy majority given as a fuzzy linguistic quantifier we have been able to attain, on the one hand, a proper expressive power (an adequate and human consistent representation of preferences)

and, on the other hand, to make available tools that can lead to many powerful results (solution concepts and their properties) and help solve some inherent difficulties exemplified by voting paradoxes.

References

- [1] J.B. Kim, Fuzzy rational choice functions. *Fuzzy Sets Syst.*, 10 (1983) 37–43.
- [2] M. Salles, Fuzzy utility. In: S. Barberá, P.J. Hammond, and C. Seidl (eds.): *Handbook of Utility Theory*, Kluwer, Boston, 1996.
- [3] K.J. Arrow. *Social Choice and Individual Values*, 2nd ed. Wiley, New York, 1963.
- [4] J.S. Kelly. *Arrow Impossibility Theorems*. Academic Press, New York, 1978.
- [5] H. Nurmi. *Comparing Voting Systems*, Reidel, Dordrecht, 1987.
- [6] J. Fodor and M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer, Dordrecht, 1994.
- [7] B. De Baets and J. Fodor. Twenty years of fuzzy preference structures (1978–1997). *Belg. J. Oper. Res. Statist. Comput. Sci.* 37 (1997) 61–82.
- [8] B. De Baets, E.E. Kerre and B. Van De Walle. Fuzzy preference structures and their characterization. *J. Fuzzy Maths* 3 (1995) 373–381.
- [9] B. De Baets, B. Van De Walle, and E.E. Kerre. Fuzzy preference structures without incomparability. *Fuzzy Sets Syst.* 76 (1995) 333–348.
- [10] F. Herrera, and E. Herrera-Viedma. Choice functions and mechanisms for linguistic preference relations. *Eur. J. Oper. Res.* 120 (2000) 144–161.
- [11] F. Herrera, E. Herrera-Viedma and J.L. Verdegay. A model of consensus in group decision making under linguistic assessments. *Fuzzy Sets Syst.* 78 (1996) 73–88.
- [12] F. Herrera, E. Herrera-Viedma, and J.L. Verdegay. Choice processes for non-homogeneous group decision making in linguistic setting. *Fuzzy Sets Syst.* 94 (1998) 297–308.
- [13] F. Herrera, E. Herrera-Viedma and J.L. Verdegay. Linguistic measures based on fuzzy coincidence for reaching consensus in group decision making. *Int. J. Approx. Reason.* 16 (1997) 309–334.
- [14] F. Herrera, E. Herrera-Viedma and J.L. Verdegay. A rational consensus model in group decision making using linguistic assessments. *Fuzzy Sets Syst.* 88 (1997a) 31–49.
- [15] F. Herrera and J.L. Verdegay. On group decision making under linguistic preferences and fuzzy linguistic quantifiers. In: B. Bouchon-Meunier, R.R. Yager and L.A. Zadeh (eds.), *Fuzzy Logic and Soft Computing*, World Scientific, Singapore, 1995, pp. 173–180.
- [16] B. Loewer and R. Laddaga. Destroying the consensus, In: Loewer B., (Guest ed.), Special issue on consensus. *Synthese* 62(1), 1985, 79–96.
- [17] L.A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers and Maths. with Appl.*, 9, 1983, 149–184.
- [18] R.R. Yager. On ordered weighted averaging operators in multicriteria decision making. *IEEE Trans. Syst. Man Cybern.*, SMC-18 (1988), 183–190.
- [19] R.R. Yager, and J. Kacprzyk. (eds.) *The Ordered Weighted Averaging Operators: Theory and Applications*. Kluwer, Boston, 1997.
- [20] M. Delgado, J.L. Verdegay, and M.A. Vila. On aggregation operations of linguistic labels. *Int. J. of Intelligent Systems* 8 (1993) 351–370.
- [21] J. Kacprzyk, and M. Roubens. (eds.) *Non-Conventional Preference Relations in Decision Making*, Springer-Verlag, Heidelberg, 1988.
- [22] J. Kacprzyk, H. Nurmi, and M. Fedrizzi (eds.) *Consensus under Fuzziness*. Kluwer, Boston, 1996.
- [23] J.L. García-Lapresta and B. Llamazares. Aggregation of fuzzy preferences: Some rules of the mean. *Social Choice and Welfare* 17 (2000) 673–690.
- [24] J. Kacprzyk. Collective decision making with a fuzzy majority rule. In: *Proceedings of WOGSC Congress* AFCET, Paris, 1984, pp. 153–159.
- [25] J. Kacprzyk. Zadeh's commonsense knowledge and its use in multicriteria, multistage and multiperson decision making. In: M.M. Gupta *et al.* (eds.), *Approximate Reasoning in Expert Systems*, North-Holland, Amsterdam, 1985, pp. 105–121.
- [26] J. Kacprzyk. Group decision-making with a fuzzy majority via linguistic quantifiers. Part I: A consensory-like pooling; Part II: A competitive-like pooling. *Cybern Syst: an Int. J.* 16 (1985) 119–129 (Part I), 131–144 (Part II).
- [27] J. Kacprzyk. Group decision making with a fuzzy linguistic majority. *Fuzzy Sets Syst* 18 (1986) 105–118.

- [28] J. Kacprzyk. On some fuzzy cores and ‘soft’ consensus measures in group decision making. In: J.C. Bezdek (ed.). *The Analysis of Fuzzy Information*, Vol. 2. CRC Press, Boca Raton, 1987, pp. 119–130.
- [29] J. Kacprzyk. Towards ‘human consistent’ decision support systems through commonsense-knowledge-based decision making and control models: A fuzzy logic approach. *Comput Artif Intel* 6 (1987) 97–122.
- [30] H. Nurmi. Approaches to collective decision making with fuzzy preference relations. *Fuzzy Sets Syst.* 6 (1981) 249–259.
- [31] H. Nurmi. Imprecise notions in individual and group decision theory: Resolution of Allais paradox and related problems. *Stochastica*, VI (1982) 283–303.
- [32] H. Nurmi. Voting procedures: A summary analysis, *Br. J. Pol. Sci.*, 13 (1983) 181–208.
- [33] H. Nurmi. Probabilistic voting. *Political Methodology*, 10 (1984) 81–95.
- [34] H. Nurmi. and J. Kacprzyk, On fuzzy tournaments and their solution concepts in group decision making. *Europ. J. of Operational Research*, 51 (1991) 223–232.
- [35] J.C. Bezdek, B. Spillman, and R. Spillman. A fuzzy relation space for group decision theory. *Fuzzy Sets Syst.* 1 (1978) 255–268.
- [36] J.C. Bezdek, B. Spillman, and R. Spillman. Fuzzy relation space for group decision theory: An application. *Fuzzy Sets Syst.* 2 (1979) 5–14.
- [37] M. Fedrizzi, J. Kacprzyk, and H. Nurmi. Consensus degrees under fuzzy majorities and fuzzy preferences using OWA (ordered weighted average) operators. *Control Cybern* 22 (1993) 71–80.
- [38] J. Kacprzyk, M. Fedrizzi, and H. Nurmi. OWA operators in group decision making and consensus reaching under fuzzy preferences and fuzzy majority. In: R.R. Yager and J. Kacprzyk (eds.). *The Ordered Weighted Averaging Operators: Theory and Applications*, Kluwer, Boston, 1997, pp. 193–206.
- [39] J. Kacprzyk and S. Zadrożny. Collective choice rules in group decision making under fuzzy preferences and fuzzy majority: A unified OWA operator based approach. *Control Cybern*, 31 (2002) 937–948.
- [40] J. Kacprzyk, and Zadrożny. An Internet-based group decision support system. *Management*, VII (28) (2003), 4–10.
- [41] J. Kacprzyk and M. Fedrizzi. ‘Soft’ consensus measures for monitoring real consensus reaching processes under fuzzy preferences. *Control. Cybern.* 15 (1986) 309–323.
- [42] J. Kacprzyk and M. Fedrizzi. A ‘soft’ measure of consensus in the setting of partial (fuzzy) preferences. *Eur. J. Oper. Res.* 34 (1988) 315–325.
- [43] J. Kacprzyk, M. Fedrizzi, and H. Nurmi. Group decision making and consensus under fuzzy preferences and fuzzy majority. *Fuzzy Sets Syst.* 49 (1992) 21–31.
- [44] J. Kacprzyk, H. Nurmi and M. Fedrizzi. Group decision making and a measure of consensus under fuzzy preferences and a fuzzy linguistic majority, In: L.A. Zadeh and J. Kacprzyk (eds.). *Computing with Words in Information/Intelligent Systems. Part 2. Foundations*, Physica-Verlag/Springer-Verlag, Heidelberg New York, (1999) pp. 233–243.
- [45] Zadrożny, S. An approach to the consensus reaching support in fuzzy environment. In: J. Kacprzyk, H. Nurmi and M. Fedrizzi (eds.). *Consensus under Fuzziness*. Kluwer, Boston, (1997). pp. 83–109.
- [46] J. Kacprzyk and M. Fedrizzi. A ‘human-consistent’ degree of consensus based on fuzzy logic with linguistic quantifiers. *Math. Soc. Sci.* 18 (1989) 275–290.
- [47] J. Kacprzyk and M. Fedrizzi (eds). *Multiperson Decision Making Models Using Fuzzy Sets and Possibility Theory*, Kluwer. Dordrecht, 1990.
- [48] H. Nurmi. Voting paradoxes and referenda, *Social Choice and Welfare*, 15 (1998) 333–350.
- [49] H. Nurmi. *Voting Paradoxes and How to Deal with Them*. Springer-Verlag, Berlin-Heidelberg/New York, 1999.
- [50] H. Nurmi. and Kacprzyk, J. Social choice under fuzziness: A perspective. In: J. Fodor, B. De Baets and P. Perny (eds.): *Preferences and Decisions under Incomplete Knowledge*. Physica Verlag (Springer Verlag), Heidelberg and New York, 2000, pp. 107–130.
- [51] M. Dasgupta and R. Deb. Transitivity and fuzzy preferences. *Soc. Choice Welfare* 13 (1996) 305–318.
- [52] K. Sengupta, Choice rules with fuzzy preferences: Some characterizations. *Social Choice and Welfare*, 16 (1999) 259–272.
- [53] M.D. Intrilligator. A probabilistic model of social choice. *Revi. Econ. Stud.* 40 (1973) 553–560.

44

FuzzJADE: A Framework for Agent-Based FLCs

Vincenzo Loia and Mario Veniero

44.1 Introduction

Due to performance requirements or constraints (whether software or hardware), most real fuzzy control applications require a limited number of inputs to the rule base in order to make the control surface suitable to define, test, and implement simple fuzzy control rules for simple environments. Nevertheless, in the more complex surroundings defined by real control processes or large-scale systems the number of inputs would not be so limited. Indeed, these surroundings are characterized by a simultaneous presence of many state variables and model fuzziness. A typical scenario could be provided by autonomous systems that typically, need a great number of sensors of different types, thus producing a huge input space.

Starting from our experiences in the design and implementation of hybrid complex systems faithful to the agent paradigm, in this chapter we propose a new integrated framework useful for building distributed fuzzy control systems by means of autonomous soft computing agents: FuzzJADE. The presented framework enables the conceptualization of fuzzy logic controllers as ontology-driven/supported interoperating JavaTM agents, allowing a fast, reusable, and scalable implementation of distributed fuzzy logic controllers (FLCs) enhanced by all features characterizing multiagent systems.

FuzzJADE can be successfully applied to contexts requiring complex interactions among distributed inference engines (IE). Particularly, it aims to answer some important issues about fuzzy control systems' scaling up capabilities. Above all FuzzJADE allows programmers to define reactive fuzzy systems able to respond to a complex environment complying with performance constraints without the application of non-fuzzy means such as throwing out useful data or decreasing the input space.

At the very basic, FuzzJADE can be viewed as an efficient framework to build autonomous fuzzy control agents (FCA) interacting through ontology-based FIPA-¹ compliant interaction protocols (IP) and executing fuzzy control activities at any level of complexity being hosted into a FIPA-compliant agents platform (AP). This approach enables to build very complex and scalable fuzzy control agents networks (FCANs) where each FCA can be fed by one or more other FCAs; here the former acts as FCAN sink, while the latter acts as FCAN source. Inner FCAN nodes work as local IE. These get their inputs from source FCAs and feed inputs of (locally) sink FCAN's nodes with its produced assignments of rules. In this

¹ Foundation for Intelligent, Physical Agents. <http://www.fipa.org>.

way, when the rule set associated to the inner FCA (a local control node) fires its output assignments can be added into the centroid calculation performed by the sinks. These are, in their turn, enabled to behave as fuzzy multiplexers able to make smooth transitions between multiple recommendations according to qualitative rules. The described decomposition, also known as fuzzy preprocessing, allows the mapping needed at each stage to be kept much simpler and performative than the usual approach while keeping unchanged features and granularity of the input space. Moreover, the model allows to distribute fuzzy control nodes onto autonomous, mobile, and intelligent agents enabling improved distributed inference handling strategies.

From another prospect, FuzzJADE can be viewed as a fuzzy-based framework supporting the fuzzy model information granulation process by integrating fuzzy models area with agent technology area. From this point of view, it supports several out of the essential factors driving all pursuits of information granulation [1, 2]. Namely, FuzzJADE enables the factorization of complex fuzzy models into simpler, interoperating submodels. Indeed, it proposes a hierarchical distributed fuzzy control model implemented by means of software agents, each of them providing fuzzy concept sensor, actuator, or inference engine services. In this way, each FCAN node represents a way to reduce the conceptual complexity of a fuzzy model. Each FCA works around information granules. It encodes (constructs) them both fuzzifying pure numerical values (the physical sensor encoding level) and inferring them by means of qualitative rules (the inference engine encoding level). Then, other agents decode obtained information granules translating them again to the numerical level defuzzifying the resulting granules through suitable operators. FuzzJADE supports and handles all interactions needed by the granulation process hiding its complexity to the FCAN designer. Furthermore, it enables communication among granular worlds by means of both a shared distributed knowledge management system and the sharing-based pattern the FCA is faithful to. Indeed, FuzzJADE distributed knowledge management system is queryable by any FCA embedded into one out of several FCAN hosted on (possibly different) FIPA-compliant APs. Finally, it enables the definition of interconnected fuzzy knowledge domains and subdomains, browsable by any FCA.

FuzzJADE has been implemented in the form of a JADE [3] add-on. This choice has been driven principally both by JADE's optimal support to network communication activities and its reduced messages round-trip time [4]. Moreover, JADE is a fully FIPA-compliant platform enabling inter/intraplatform agents mobility [5]. Finally, JADE is an LGPL (Lesser General Public Licence Version 2)-released open-source platform easily extensible through the add-ons mechanism and is gaining an increasing industry interest.

FuzzJADE design and development phases have been driven by the following guidelines:

- Deeply analyze the technological context of the framework in order to identify key elements to be dealt with.
- Define the distributed, agent-based, fuzzy control model to be supported, adopting a suitable agent-oriented software engineering (AOSE) technique.
- Formalize the semantic dimension of communications among model-defined elements identifying their ontological domain.
- Define suitable cooperation models and IPs to enable sociality-based control activities.
- Fully develop the framework in the form of additional FIPA-compliant AP services.
- Define and implement a test bed selecting a real control process. The test-bed has to emphasize the values coming from the adoption of the framework and, at the same time, produce results comparable to those obtainable by classical approaches.

The chapter is organized as follows. Section 44.2 presents FuzzJADE distributed organization and model describing the main framework's components, such as fuzzy control sensor, inference engine and consumer FCA special roles. The section tries to sketch the modeling process that has been followed by authors according to the GAIA [6–8] AOSE modeling technique. Section 44.3 describes the FuzzJADE knowledge management (KM) system component, showing its main services and, at the same time, detailing some relevant ones. Here are also explained the basics of some notable IP involving FCAs and the KM system. Section 44.4 deals with the FuzzJADE communication level and its ontological foundations. Finally, Section 44.5 closes the chapter presenting conclusions and future works.

44.2 FuzzJADE Agent-Based Distributed FLC Model

Figure 44.1 allows to deal with FuzzJADE as a framework aimed at modeling FLCs in the form of a collection of virtual organizations, grouping agents working together to achieve a common goal or set of goals. All these virtual organizations are supervised by the more general FCAN organization offering tools and services enabling complex control/management procedures, workflows, and interactions among all the FLCs. The provided services are principally centered on distributed control and knowledge management. Figure 44.1 describes structural entities and relationships in the FuzzJADE environment by means of an organization diagram. The FCAN organization owns a ‘KM system,’ offering fuzzy concepts yellow pages service and in charge of handling knowledge base and catalogs of the fuzzy control network.

Each DFLC running under a FuzzJADE-enabled platform is seen as a suborganization having its own teams and eventually sharing resources (tailorers, sensors, and actuators) with other already-running DFLCs. The administrative team of each DFLC is responsible, when needed, to tailor (customize) fuzzy knowledge to be handled or used by sensing and actuating teams. FuzzJADE offers two ways to achieve this task. The first one, performed by a ‘concepts tailor’ roled agent, consists in requesting to the yellow pages agent to update its fuzzy concept knowledge base. This, in its turn, performs the update just after having concorded the requested modification with all the tailorers registered onto the KM system as involved into the tailoring process of such concept. As soon as the update has been performed, the yellow pager coordinates the updating of local believes of each registered fuzzy concept user (a ‘concept user’ roled agents), depending on the aforementioned concept. The second way to adjust distributed fuzzy knowledge is through the tailoring of fuzzy rule bases distributed on the network’s control nodes. This task is achieved by a ‘rule base tailor’ roled agent, which is able to impose the manipulation of the believed rules on a ‘fuzzy inference engine.’

The sensing and actuating teams, respectively, are in charge of sensing fuzzy input values, eventually fuzzifying crisp values coming from physical sensors and performing fuzzy control activities, e.g., controlling actuators with crisp output obtained properly defuzzifying inferred or received fuzzy control

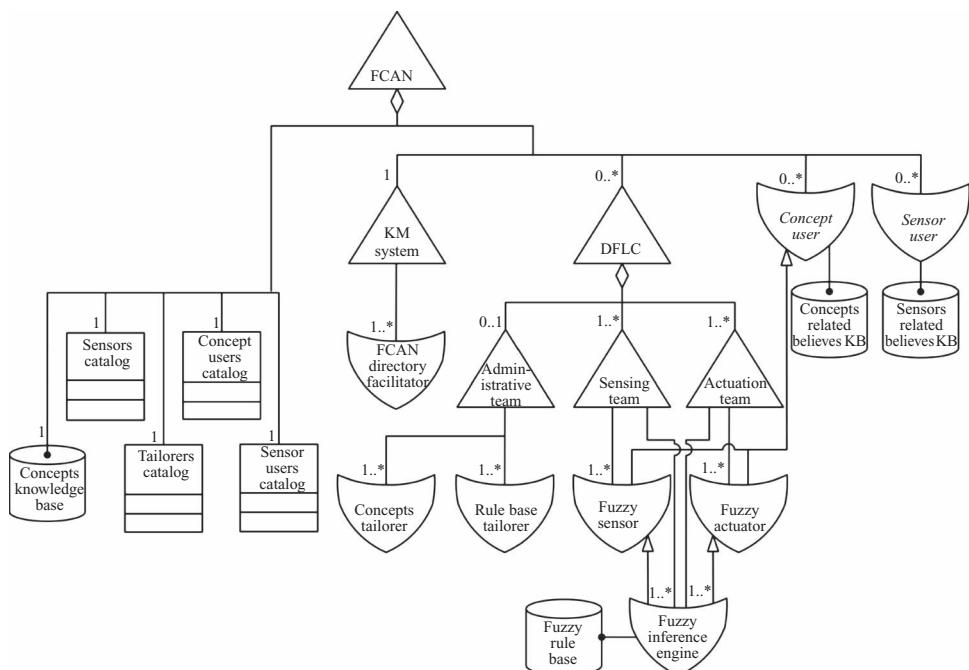


Figure 44.1 FuzzJADE organization diagram

values. Members of these teams are ‘fuzzy sensor’ and ‘fuzzy actuator’ roled agents. Each actuator agent needing the measurement of a given concept subscribes to the corresponding sensor in order to be notified whenever the measured value changes. The last role, ‘fuzzy inference engine,’ belongs to both sensing and actuating teams. Indeed it concurrently performs both the characterizing tasks. It processes fuzzy values coming from ‘fuzzy sensor’ roled agents (in this way working as an actuating team member) and, at the same time, produces an inferred output control value eventually feeding it as input to one or more ‘fuzzy actuator’ roled agents (thus working as a sensing team member). In real applications, all FCAN inner nodes should be ‘fuzzy inference engine’ roled agents.

The given description lists both the main entities belonging to FuzzJADE-enabled AP and the FCAN organization main activities. To go deeper in the FuzzJADE framework intended goals, have a look at the goal/task implication and delegation diagram, depicted in Figure 44.2. This diagram shows that the main goal of a FuzzJADE-enabled AP (‘DFLCs assisted’) is satisfied when all fuzzy control activities have been performed by any hosted DFLCs and the global fuzzy knowledge has been managed. It is worth noticing here that, being these two activities typically not terminating ones, the aforementioned goals can be viewed as logical services of a fuzzy-enabled AP. Both the ‘fuzzy control achieved’ and ‘fuzzy knowledge managed’ are, in turn, split into a set of subgoals. For instance, ‘fuzzy control achieved’ is satisfied when fuzzy inputs have been sensed, fuzzy inference performed, and fuzzy control action (FLC outputs) processed. Moreover, the diagram depicts two further subgoals or-ed with the previous ones: ‘fuzzy concepts tailored’ and ‘fuzzy rules tailored.’ These subgoals deal with FLCs where some kind of learning and adjustment of concepts or rules is performed or by means of automatic deductions or through human supervision. With respect to ‘fuzzy knowledge managed,’ the diagram shows how it is satisfied when one out of the knowledge management subgoals is satisfied. Each of these, in their turn, refers

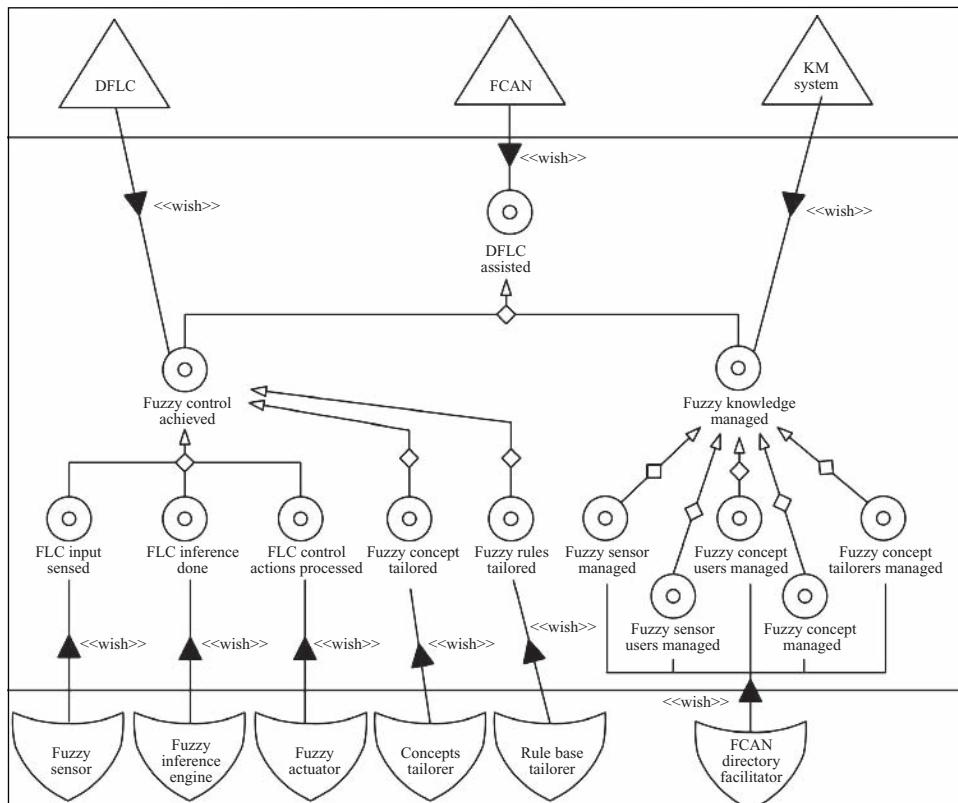


Figure 44.2 FuzzJADE goal/task implication and delegation diagram

to a particular fuzzy knowledge management aspect. ‘fuzzy concepts managed’ is a goal related to the storage and distribution of fuzzy concepts definitions in order to make them available to any kind of fuzzy concept user (FCAN node) needing it to fulfill its own activities or tasks. Remaining subgoals are relative to the handling of special catalogs in charge of tracking agents involved with or interested in any kind of available information, e.g., which FCA senses a given concept, which FCA needs to be notified when a given concept sensor appears, which FCA is responsible for the tailoring of a given concept, and so on.

At the bottom of Figure 44.2 are depicted the assignments of subgoals to the roles defined through the FCAN organization, thus showing the subgoals delegation structure of this latter.

44.3 FCAN Knowledge Management Service

In this section will be presented the KM service provided by the KM system in order to emphasize the benefits deriving from the usage of FuzzJADE framework. For this component we will summarize all the relevant features, enabling distributed fuzzy control agent-based activities trying to illustrate the properties of the underlying model.

The KM system fulfills the ‘fuzzy knowledge managed’ goal and its subgoals by provisioning a set of services to the fuzzy-enabled AP. This set of services is shown in the task/goal implication structure diagram depicted in Figure 44.3.

Each service requires the execution of a well-defined workflow (or a set of them), during which each role executes one or more tasks following a fixed IP.

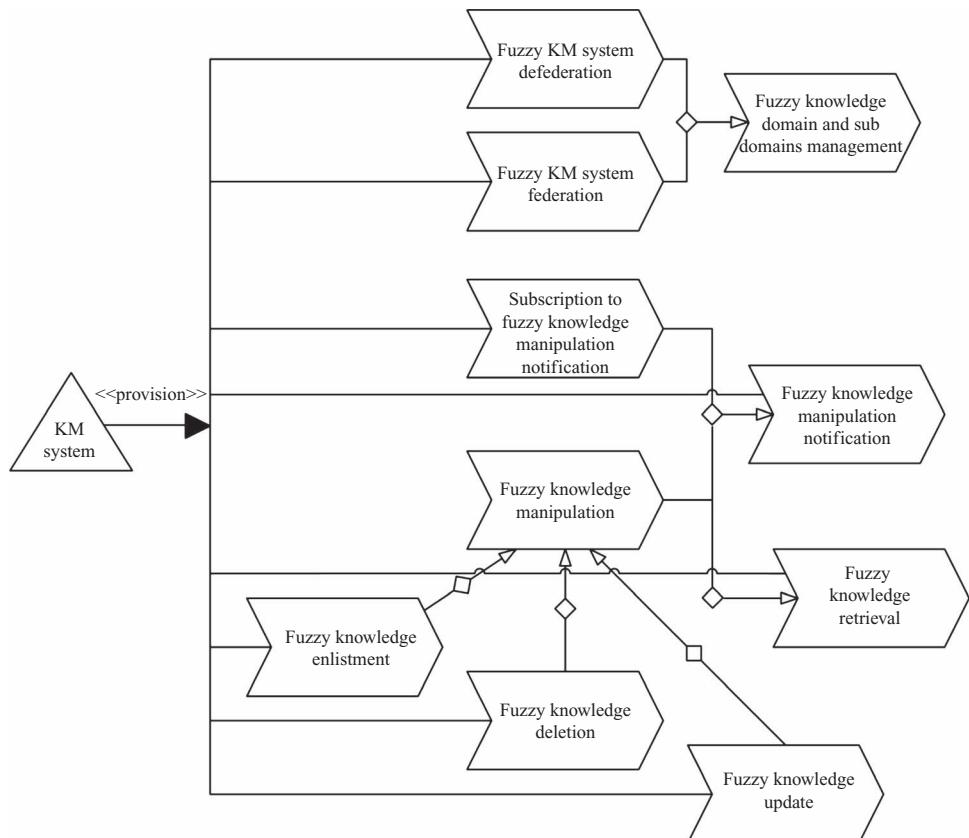


Figure 44.3 FCAN DF implication structure diagram

It is worth noticing that KM system's provided services and their behavioral properties are directly inherited from what specified in [9] with regard to the directory facilitator (DF) service. According to the multiagent paradigm, the KM system has been implemented in the form of a JADE platform service provided by a FCAN DF roled agent. Therefore, the FCAN DF roled agent is the trusted, benign custodian of the FCAN distributed knowledge. It is trusted in that it tries to maintain an accurate, complete, and timely knowledge base. It is benign in the sense that it provides the actual information in its catalogs on a nondiscriminatory basis to all authorized FCAs. We will call fuzzy-enabled AP one having a FCAN DF roled agent hosted on it.

Each fuzzy-enabled AP has a default FCAN DF agent, whose local name is the reserved value 'fcandf' and whose services are available to all DFLCs, hosted onto the same AP or onto an external FIPA-compliant AP. However, other FCAN DF agents can be activated with a different agent identifier, and several of these (including the default one) can be federated in order to provide a single distributed yellow pages cataloge. This federation implements a network of fuzzy control knowledge domains and sub-domains, adopting the same model of a JADE DF federation. Table 44.1 details the FCAN DF role schema.

At the very basic the FCAN DF allows other FCAs to publish information and provides fuzzy-control-related services so that other FCAs can find and, successively, exploit them as shown in Figure 44.4.

The default FCAN DF on a fuzzy-enabled AP has a reserved AID of

```
(agent-identifier
  :name fcandf@hap_name
  :addresses (sequence hap_transport_address))
```

The interaction with the FCAN DF, conforming to the multiagent paradigm, is possible through asynchronous message passing. An interaction can be performed exchanging FIPA ACL messages, where each message represents a linguistic act in the context of a fixed IP.

Table 44.1 FCAN directory facilitator role schema

Role schema: FCAN directory facilitator

Description:

Fulfils 'fuzzy knowledge managed' goal by provisioning 'yellow pages' services to a fuzzy-enabled AP

It also acquaints specific subscribed fuzzy control network agents and federates with other FCAN DFs on fuzzy-enabled APs

Protocols and activities:

FCAN DF Query IP, FCAN DF Request IP,
FCAN DF Subscribe IP, FCAN DF Federation IP,
FCAN Tailoring Agreement IP

Permissions:

reads/writes	supplied <i>FuzzyConceptDefinition</i>	
	supplied <i>consumerAID</i>	// acquaintance subscriber
	supplied <i>tailoreAID</i>	
	supplied <i>federatedAID</i>	// federation subscriber

Responsibilities:

Liveness:

$$\text{MANAGEFUZZYKNOWLEDGE}^2 = \underline{\text{RegisterDF}} \text{ (SubscribeStakeholder}^\omega \| \\ \text{FulfillShSearch}^\omega \| \text{FulfillShRequest}^\omega \| \text{FulfillFederationRequest}^\omega)$$

Safety: A successful connection to the knowledge base storage will be held during the whole role life

²MANAGEFUZZYKNOWLEDGE liveness property decomposition is omitted. For further details refer to Sections 44.3.1, 44.3.2, 44.3.3, and 44.3.4.

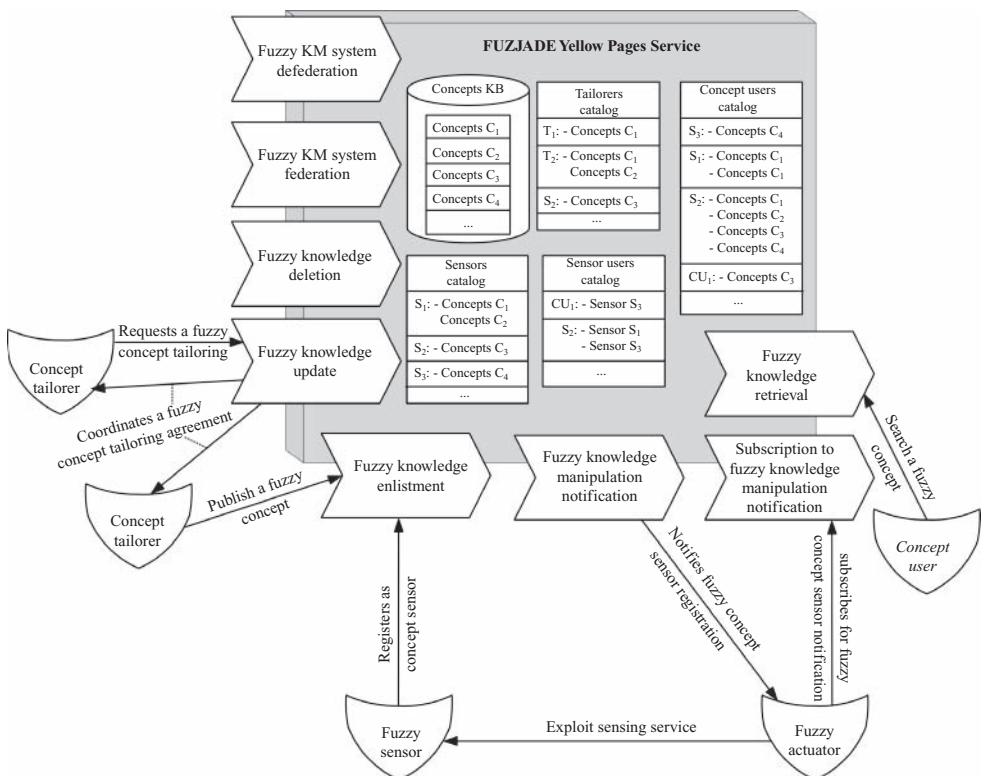


Figure 44.4 FuzzJADE yellow pages service

The interaction with the FCAN DF is done using the FIPA semantic language content language [10,11] as content language and the FCAN management ontology (an ontology specifically defined to deal with FCA network as defined by FuzzJADE framework).

In order to simplify these interactions, FuzzJADE provides a class utility (specifically the class FCANDFService) by means of which it is possible to manipulate, search, subscribe, and tailor fuzzy knowledge through static methods calls.

44.3.1 Querying FCAN DF Knowledge Base

With regard to fuzzy knowledge retrieval the FCAN DF allows to perform complex and extended search using the “FCAN DF Query” IP. Here, each initiator is able to use identifying referential expressions (IRE) to specify the needed information. For details about the expressive power of usable IREs, refer to the IP grammar specification listed below in this section.

The FCAN DF Query IP allows a fuzzy agent to request information about fuzzy concepts to the directory facilitator. This IP is based on the FIPA Query IP and is identified by the token fipa-query as the value of the protocol parameter of the ACL message.

The representation of this IP is given in Figure 44.5, which is based on extensions to UML1.x [12].

44.3.1.1 Explanation of the Protocol Flow

The FCA, acting as IP initiator, requests the participant directory facilitator to perform an inform action using one out of of query-if or query-ref [13] communicative acts (CA). A query-if communication is

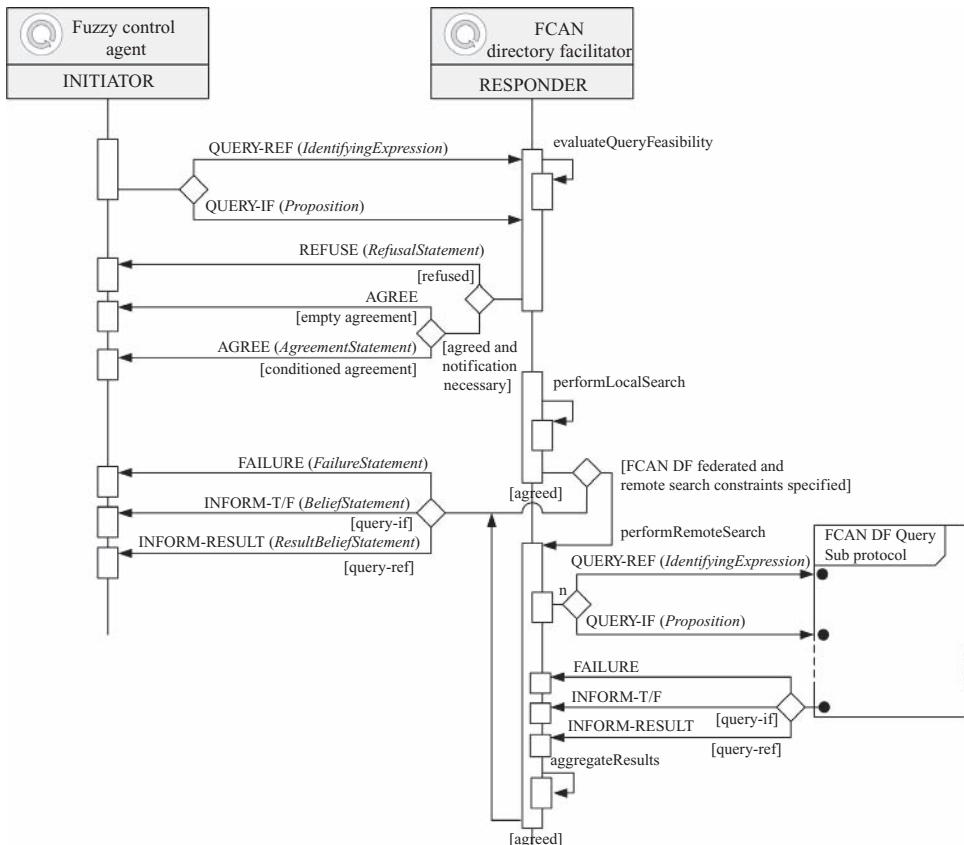


Figure 44.5 FCAN DF query interaction protocol

used when the initiator wants the DF to argue about the truth of a particular proposition involving fuzzy concepts, while the query-ref communication is used when the initiator wants to identify fuzzy concepts by means of propositional assertions. The DF evaluates the received CA and makes a decision whether to accept or refuse it. In the latter case [refused] becomes true and the participant communicates a refuse with its associated refusal statement. The refusal statement is in the form of a pair <intended action requested, refusal motivation>. Otherwise, [agreed] becomes true. In this case, if conditions indicate that an explicit agreement is required (i.e. [notification necessary] is true due to eventually time-consuming evaluations) then the participant communicates the (optional) agreement, possibly along with its feasibility conditions.

The FCAN DF encompasses a searching mechanism that searches first locally and then extends the search to other FCAN DFs if allowed and required. Once agreed the participant engages a local search or a remote search if it is federated with other FCAN DFs and the specified search constraints enable it. In this latter case, the FCAN DF initiates a new instance of this IP, having as participants all the directly federated DFs aggregating received results. The chosen search mechanism is a concurrent breadth-first search across FCAN DFs. This way the whole ontological domain and subdomains are searched against the initial query. Once the search has been completed, if the participant fails, then it communicates a failure to the initiator specifying the CA it intended to perform and the failure reason. On the contrary, in a successful response the participant replies with one out of the following versions of inform:

- an inform-t/f communication (responding to a query-if) where the message content states the belief of the participant about the truth or falsehood of the proposition, or,
- an inform-result communication (responding to a query-ref) with a message containing the participant belief about the objects satisfying the referring expression for which the query was specified.

At any point in the IP, the receiver of a communication can inform the sender that it did not understand what was communicated by returning a not-understood message having as content a pair *<received message, not-understood field>*. As such, Figure 44.5 does not depict a not-understood communication as it can occur at any point in the IP. The communication of a not-understood within an IP terminates the entire IP.

Any other low-level feature of this IP matches the corresponding from the FIPA-query [14].

44.3.1.2 IP Messages Details

To encode the content of both initiator and participant messages a proper subset of the one defined by the FIPA SL content language specification has been used. It has been customized in order to fit the need of the IP (when applied to the query-if or query-ref CAs), taking into account the ontological domain it applies to. Any message content outside the language generated by the following grammar results in a not-understood response act. In what follows we present only the reduced grammar productions.

The grammar used to encode the initiator query-if message content is defined as follow:

```

Proposition      = Wff
Wff             = AtomicFormula
                  | "(" UnaryLogicalOp Wff ")"
                  | "(" BinaryLogicalOp Wff ")"
AtomicFormula   = "(" PredicateSymbol TermOrIE+ ")"
PredicateSymbol = "has-local-name"
                  | "has-unit"
                  | "belongs-to-namespace"
                  | "is-sensed-by"
                  | "is-tailored-by"
TermOrIE        = Term
Term            = FunctionalTerm
                  | Constant
                  | Sequence
                  | Set
FunctionalTerm  = "(" FunctionalSymbol Parameter* ")"
FunctionalSymbol = "fuzzy-concept-identification"
                  | "agent-identifier"

```

where the allowed parameters corresponding to the specified functional symbols are the ones defined, respectively, in the fuzzy concept (specifically defined to fulfill the applicative domain) and the FIPA agents management ontology [9] (with regard to the homonymous frames). In the same way, the semantic of predicate symbols is driven by the homonymous frames in the fuzzy concept management ontology.

The grammar used to encode the initiator query-ref message content is the same as the one just defined for the query-if CA, with the addition of the IdentifyingExpression production and the extension of the Term definition.

```

IdentifyingExpression = "(" ReferentialOperator Variable Wff ")"
                  | "(" ReferentialOperator "(" "sequence" Variable+ ")" Wff ")"
                  | "(" ReferentialOperator "(" "set" Variable+ ")" Wff ")"
Term = Variable
      | FunctionalTerm
      | Constant
      | Sequence
      | Set

```

Here the intended semantic of referential operators is defined in [10].

Finally, some notes on the content of response messages replied by the participant. The adopted grammar is the full FIPA SL content language grammar. As stated by the protocol, the possible CAs are inform, inform-ref, agree, not-understood, refuse, and failure.

For the last three CAs, the message content is in the form of a tuple $\langle \text{ActionExpression}, \text{Proposition} \rangle$. Here *Proposition* specifies the c.a. explanatory reason and *ActionExpression* is specialized as follows: in the case of a not-understood message it specifies the received CA while, in the remaining cases, it describes the requested action to be carried out by the participant, i.e., in this IP, an inform-ref CA.

The agreement message content, where necessary, may be both empty or contain a feasibility statement, eventually stating conditions on which the execution of requested action is depending on. Typically this will be the successful completion of a knowledge base search action.

An inform CA content will contain a belief predicate stating the DF believing about the truth of the queried proposition. Finally, an inform-ref CA will contain a predicate asserting the DF substitutions performed into the ReferentialExpression. For the underlying semantic models refer to [13].

44.3.2 Manipulating FCAN DF Knowledge Base

In order to manipulate the FCAN knowledge base, the FCAN DF is able to perform the following functions defined onto the ontological domain of objects of type fuzzy-concept-identification, fuzzy-concept, fcan-sensor-agent-description, fcan-tailorer-agent-description, and fcan-agent-description (see Section 44.4 for further details).

- *add* allows an initiator FCA to request the FCAN DF to add the given predicate to its knowledge base.
- *remove* allows an initiator FCA to request the FCAN DF to remove the given predicate from its knowledge base.
- *modify* allows an initiator FCA to request the FCAN DF to modify its belief about a given predicate replacing it with a new one.

The ‘FCAN DF Request’ IP must be used by FCAs wishing to request a FCAN DF to perform one of these actions. This IP allows a fuzzy agent to request DF to perform an action related to fuzzy knowledge manipulation. ‘FCAN DF Request’ IP is based on the FIPA Request IP [15] and is identified by the token *fipa-request* as the value of the protocol parameter of the ACL message. The representation of this IP is given in Figure 44.6, which is based on extensions to UML1.x [12].

44.3.2.1 Explanation of the Protocol Flow

The FCA, acting as IP initiator, requests the participant FCAN DF to perform an inform action using request (see [13]) CAs. The DF evaluates the feasibility of the requested action and decides whether to accept or refuse it. In the latter case [refused] becomes true and the participant communicates a refuse with its associated refusal statement. Otherwise, [agreed] becomes true. In this case, if conditions indicate that an explicit agreement is required (i.e. [notification necessary] is true due to possibly time-consuming action involved, such as the concept-tailoring process), then the participant communicates to the initiator the (optional) agreement, eventually along with its feasibility conditions.

Once the request has been agreed upon, the participant engages in the action perform. This process may involve complex subinteractions among the FCAN DF and other parties or may resolve into a simple knowledge base update. The first case refers to fuzzy concept tailoring and requires that all the involved concept tailors agree with the desired modification. The agreement is performed by means of the ‘FCAN Tailoring Agreement’ IP that the FCAN DF initiates with all the known concept tailors registered with the FCAN DF or FCAN DF Federation (in the case FCAN DF belongs to such a federation). After the global acceptance the protocol can proceed. In the case at least a tailorer rejects the proposal the participant issue a failure stating the occurred rejection.

As result of the performed action the participant issues either

- an inform-done, if it successfully executed the request, or
- a failure, if it failed in its attempt to fulfill the request.

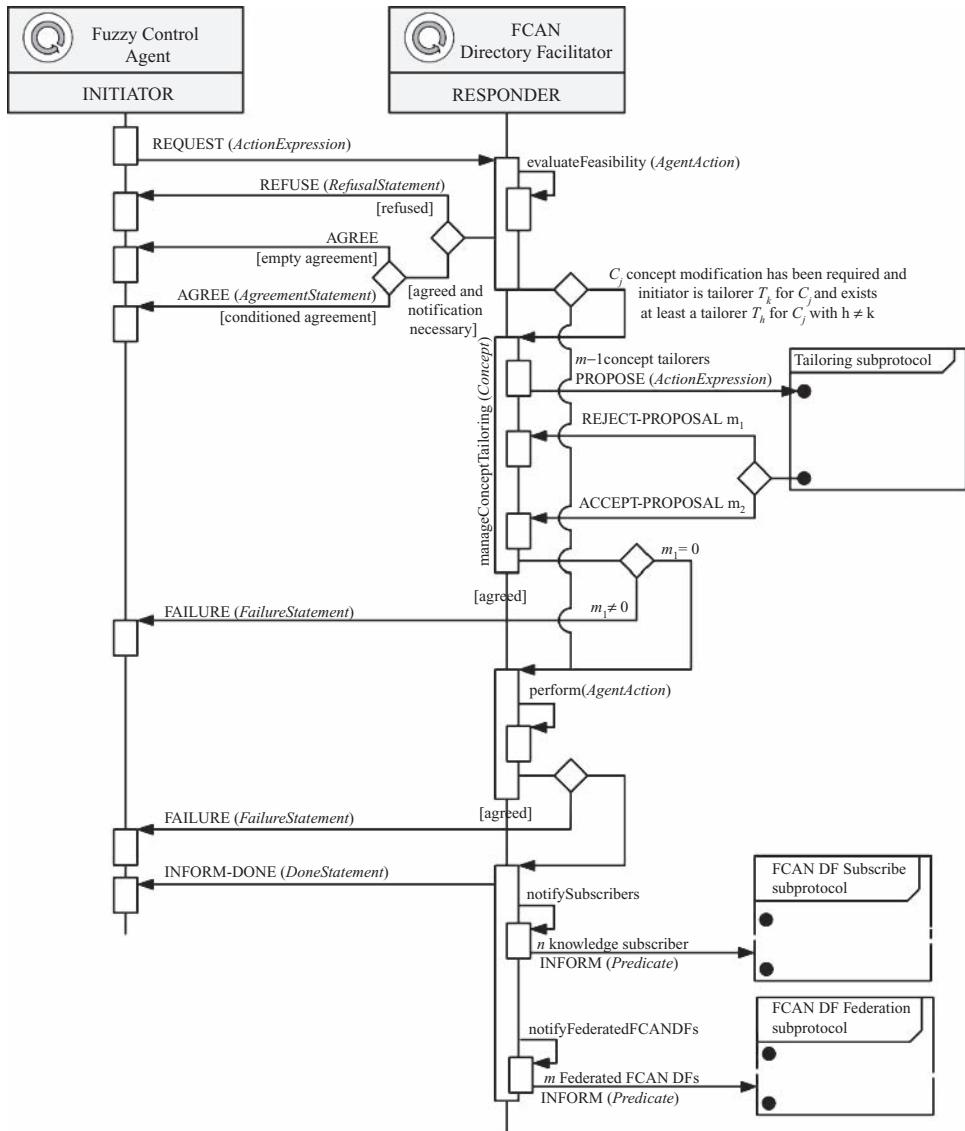


Figure 44.6 FCAN DF request interaction protocol

Again, at any point in the IP, the receiver can inform the initiator that it did not understand what was communicated. This is achieved by returning a not-understood message having the content in the form of a pair \langle received message, ununderstood field \rangle . The communication of a not-understood within the IP terminates the entire IP. Any other low-level feature of this IP matches the corresponding one in the FIPA-Request IP [15].

44.3.3 (Un)subscribing to FCAN DF Knowledge Base

The FCAN DF supports a subscription-based extended knowledge base querying mechanism in order to allow FuzzJADE FCAs to subscribe in order to be notified about insertions, deletions, and modifications

of certain knowledge base information. In order to subscribe to the FCAN DF, an initiator must use the ‘FCAN DF Subscribe’ IP. This protocol is a direct extension of the FIPA Subscribe [16] IP and is identified by the *fipa-subscribe* token as the value of the protocol parameter of the ACL message.

Being the subscription act a persistent version of the query-ref the message content respects the same grammar specified for a query-ref ACL message and specifies an allowed identifying expressions in the ontological domain. The protocol can be terminated at any type through a cancel act.

44.3.4 (*Un*)federating FCAN DF

The FCAN DF supports a federation mechanism in order to distribute FCAN fuzzy-related knowledge across ontological domains and subdomains. The federation process is a two-step mechanism. First a given FCAN DF is requested to federate with one or more other FCAN DFs by means of the ‘FCAN DF Request’ IP. Here the request CA specifies a *federateWith* action expression. Then, if agreed, the FCAN DF engages in ‘FCAN DF Federation’ IP with the specified set of FCAN DFs. This subprotocol is again based on FIPA Subscribe IP, even though both Initiator and Responder are constrained to be FCAN DFs.

Figure 44.7 depicts the general interactions in the described process and, at the same time, the knowledge distribution model. Each FCAN DF holds a fuzzy ontological subdomain. The federation has the effect to merge FCAN DF held ontological subdomain into a superdomain allowing each FCA to query about the merged domain and to be notified about changing references.

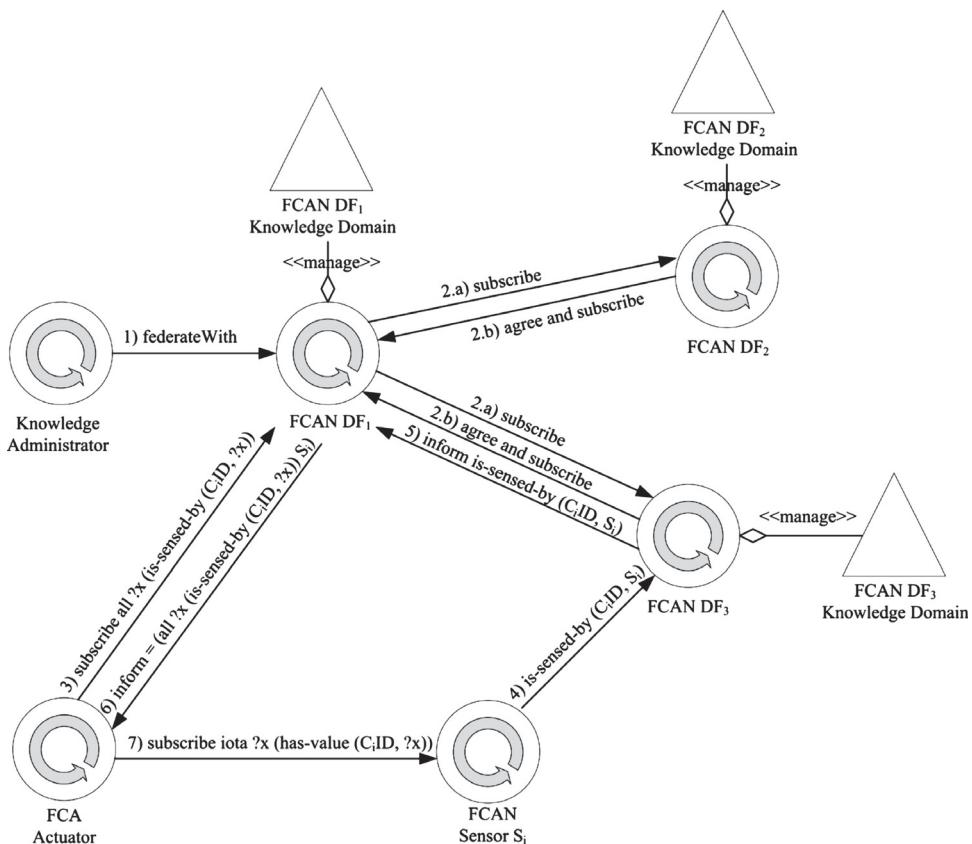


Figure 44.7 FCAN DF federation processes

44.4 Semantic Dimension of FCAs Communications

44.4.1 Modal Logic-Based Interaction Model

As shown in Section 44.3 (specifically with regard to the interactions with the DF) each interaction among actors from the FCAN conforms to a well-defined IP, generally obtained extending one out of the FIPA standardized set of IPs. All IP messages are encoded using the FIPA Semantic Language content language [10]. This is the modal logic language that sustains the theory of agency and defines the FIPA-ACL semantics [17].

The FuzzJADE support to the Agent Platform aims at taking better benefits from the semantic dimension of the FIPA-ACL language, moving the fuzzy logic control modeling in the direction of the semantic modeling and, at the same time, letting control actions to be viewed as agents mental attitudes where actions are performed on a reflexive base depending on the evaluation of the agents believes. This is particularly true when dealing with both interactions with the KM system that are oriented to the manipulation of the distributed knowledge base) and fuzzy inference engines rule bases tailoring process. Indeed, FuzzJADE implements the whole set of IPs needed to support FCAN nodes interactions embedding them into the definition of specific roles. These are, in their turn, assignable to agents through core class utilities. FuzzJADE implements the underlying operational model hiding its inherent formal complexity to the developer. This allows the production of agents whose interactions soundly conform to the FIPA-ACL semantics.

44.4.2 Fuzzy Knowledge Base Ontological Model

The semantic modeling support provided by FuzzJADE extends to an attempt to model the fuzzy control domain in order to make it effectively usable for interagent communications. The fuzzy-enabling framework comes with a set of ontologies (Figure 44.8) allowing to model fuzzy sets, fuzzy concepts, fuzzy rule bases, and many other features, allowing them to be easily used as ACL message content elements. All these ontologies are defined through extension mechanisms based on inclusion that enhance, once more, the scalability of FuzzJADE with regard to its modeling capabilities. For each ontology has been provided an OWL definition, thus revealing our further interest into exploring the semantic dimension of the provided model.

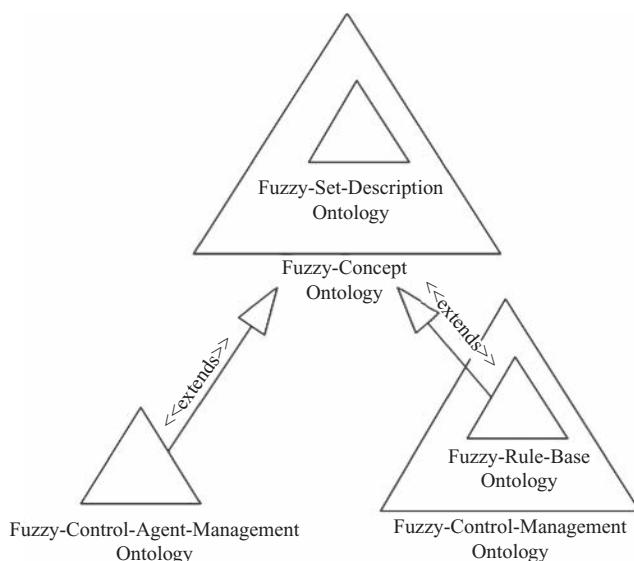


Figure 44.8 FuzzJADE main ontologies

The diagram depicted in Figure 44.8 shows the structural dependencies among the ontologies supporting FCAs communications. Here, the ‘Fuzzy-Set-Description’ ontology defines concepts allowing to describe the fuzzy sets shapes of terms defining needed fuzzy concepts. These are, in their turn, described by the ‘Fuzzy-Concept’ ontology. Both ontologies are used as building blocks for the description of fuzzy rule bases and constitutes FuzzJADE *Domain Ontologies*. Domain ontologies are finally extended to define ontologies supporting interagent communications, such as ‘Fuzzy-Control-Agent-Management’ and ‘Fuzzy-Control-Management’ ontologies. Both define concepts related to FCAN nodes and knowledge handling. In particular, the latter has been extended from the former in order to express FCA’s rule bases tailoring related concepts.

In what follows, we present the basic ontological models of fuzzy sets and concepts, deferring to a next publication other details.

44.4.2.1 Basic Ontologies

The ontological model used to define fuzzy set shapes is based on three main elements: Fuzzy-Set-Description, Fuzzy-Set-Point, and Fuzzy-Set-Function. Figure 44.9 shows a formal graph representation focusing mainly on their taxonomy.

The *Fuzzy-Set-Description* class stands for any parameterizable fuzzy set shape. Indeed, this class is defined as a base abstract class. Many subclasses have been defined as part of the application domain modeling process, each representing an a priori well-known fuzzy set shape. Each subclass has well-defined properties (data-type or object properties) needed to fully define the corresponding shape.

The *Fuzzy-Set-Point* class provides a means to represent a point by using a coordinate pair. Here the ordinate value represents the membership value of the shape at a particular abscissa. Namely, this class has been used to model shapes obtained by means of the interpolation of a set of points or to represent shapes inferred by FCAN IE agents.

Finally, the *Fuzzy-Set-Function* class represents functions encoded through an XML-based mathematical language (such as MathML or OpenMath). Instances of this class are used when defining shapes by means of functions generating a shape edge. Namely, this is the case of all the classes in the diagram whose name has been prefixed with the term *General*. The ontological model for fuzzy concepts has been obtained by extending the aforementioned one. Basing on six more main elements, its formal graph representation is depicted in Figure 44.10.

Here, the formal graph represents the whole ‘Fuzzy-Concept’ ontology and is concerned with central domain concepts that are typical subject of an interaction. Main elements from this ontology are Fuzzy-Set, Fuzzy-Term, Fuzzy-Concept, and Fuzzy-Value.

A fuzzy set is a mapping of a set of real numbers onto a membership value in the range [0, 1]. In this ontology, the *Fuzzy-Set* class is a means to express complex mappings defining basic shapes descriptions (specialized from a *Fuzzy-Set-Description*) or by using fuzzy logic operators such as complement, sum, union, and intersection.

The *Fuzzy-Term* class denotes components used to describe *Fuzzy-Concepts*. Described using a *term-name*, along with its *defining-set* or *linguistic-expression*, *Fuzzy-Term* provides the basis for a grammar allowing to describe *Fuzzy-Concepts* in a human-like manner. The *Fuzzy-Concept* class allows to describe concepts from the domain of FCAN activities. It is defined in terms of a unique uniform resource identifier (URI) (*Fuzzy-Concept-Identification*, a placeholder for a URI), a measurement *unit* (when required), upper and lower bounds of the universe of discourse, and, finally, a set of *concept-terms* that will be used to describe the fuzzy concept. Each *Fuzzy-Concept* comes with the recommended defuzzification method (*defuzzify-by* property).

Fuzzy-Concepts published and administered by Concepts Tailorer (see Section 44.2) can be required by any FCA node which, in its turn, can search against the FCAN-DF to retrieve concepts definition prior to start their own control activities. The last main concept is the the *Fuzzy-Value* class. This class stands for all measured values for a fuzzy concept. Similar to the definition of a *Fuzzy-Term*, it is typically produced by a sensor-roled FCA through a fuzzification process, then used as rule engine input by inference engine-roled agents, and, finally, used (along with its *referred-concept*) by FCAN sink nodes that through a defuzzification operation (possibly according to the proposed defuzzification method) produce crisp control actions.

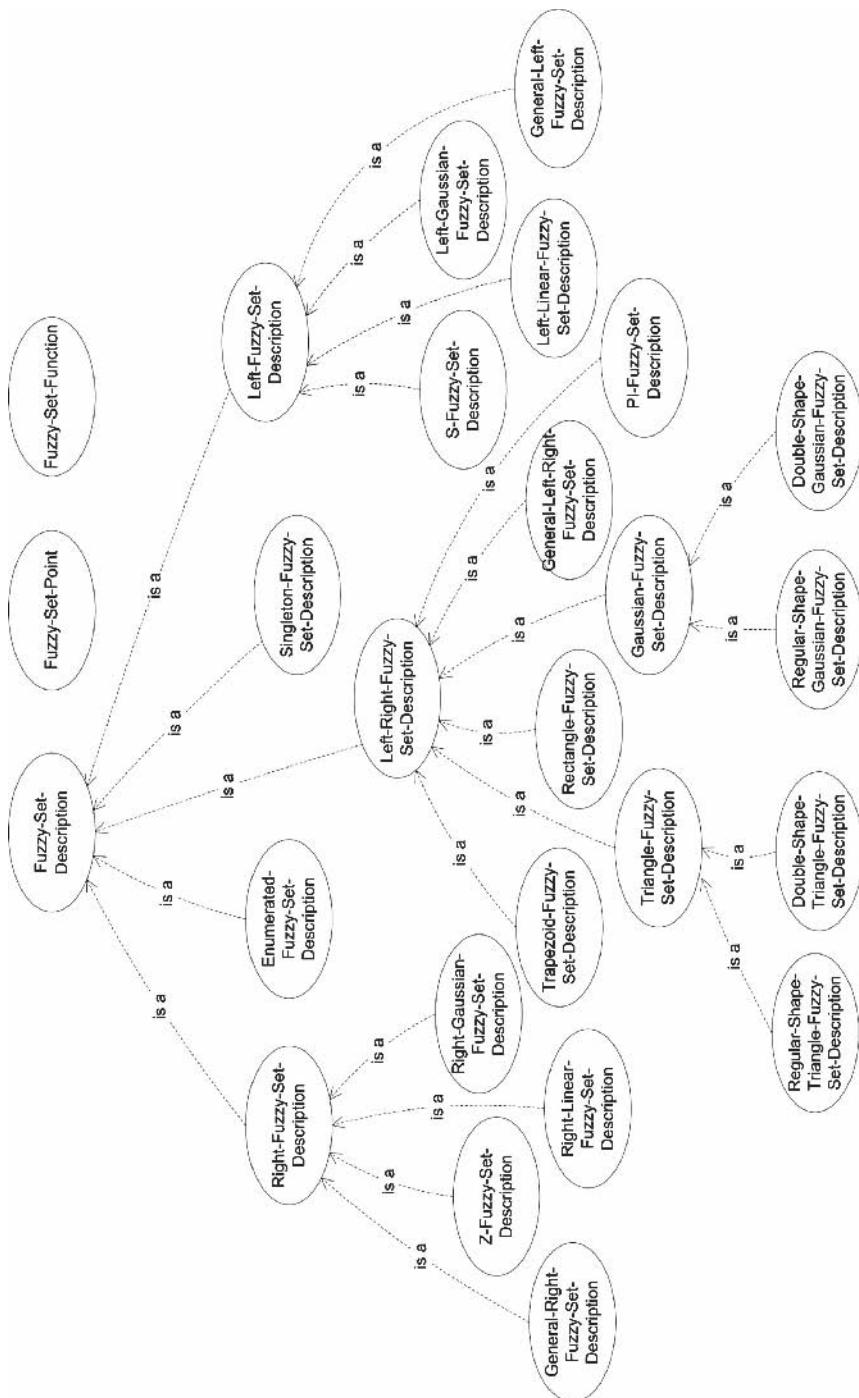


Figure 44.9 Fuzzy-set-description taxonomy

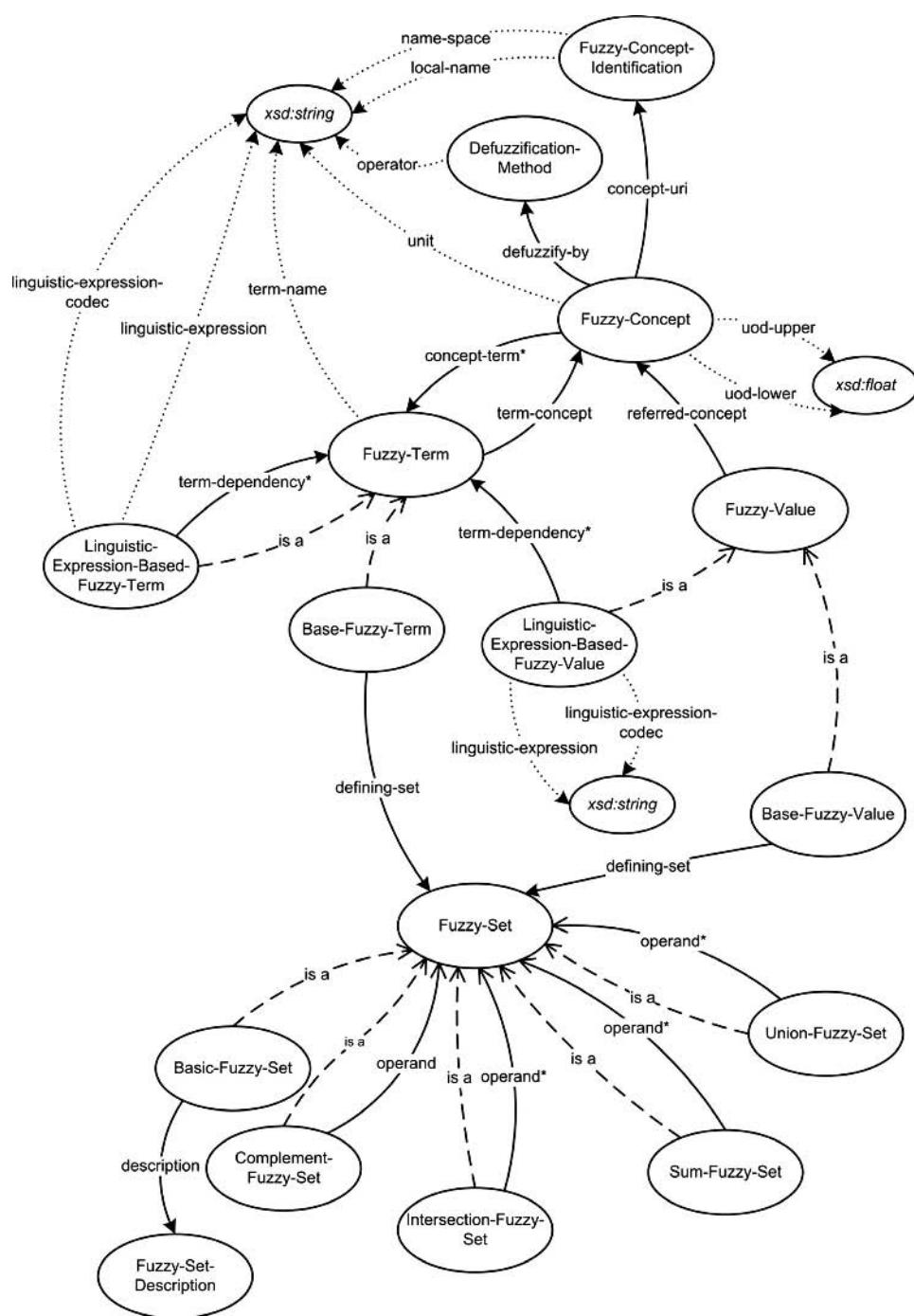


Figure 44.10 Fuzzy-concept ontology

For each class, FuzzJADE defines a FIPA SL representation and a suitable JavaBean-based class to support message content encoding. This simplifies communications among agents and, at the same time, hides its inherent management complexity to the developer.

44.5 Conclusions and Future Works

This chapter presented an agent-based model and an enabling framework, FuzzJADE, allowing the development of distributed FLCs by means of soft computing agents according to a hierarchical model. Each interaction among agents is allowed through IPs compliant with the FIPA ACL semantic. FuzzJADE proposed model can be successfully applied to control contexts requiring complex interactions among distributed IE, supporting the information granulation enabled by fuzzy modeling. FuzzJADE tries to spread this support along the whole information processing pyramid offering facilities applicable to any of the processing pyramid layer. In particular, at the lowest level, being concerned with numeric processing, FuzzJADE provides FCAs specialized to act as fuzzy value sensors able to fuzzify crisp inputs sensed by embedded physical sensors. Indeed, these agents are specialized to support the encoding phase of information granulation. At intermediate level, FuzzJADE enables inference engines in charge of producing larger inferred information granules. Finally, at the highest level we find FCAs acting as consumers and specializable to be devoted to symbol-based processing. FuzzJADE enables many other interesting features related to granular computing characterization (e.g., communications among granular worlds) allowing the sharing of both resources (namely FCAs) and knowledge (concepts and rules).

The proposed model reaches high levels of scalability due to the underlying adopted agents technology. Indeed, the model allows the runtime construction, maintenance, and tailoring of hierarchical fuzzy control networks. These, in their turn, under agent paradigm main characteristics enables the development of any desired schema, allowing to take into account simple serialized schemas as far as hierarchical prioritized ones.

One out of the several applications built or in progress adopting FuzzJADE is the SITI [18,19] Project (Safety In Tunnel Intelligent) conducted by TRAIN consortium (Consorzio per la Ricerca e lo Sviluppo di Tecnologie per il TRASporto INnovativo), with the University of Salerno among its associated members. SITI is an innovative system aiming at monitoring and steering vehicular flows nearby tunnels with high accidents risk rate. Its mission is to reduce the risk of accidents which occur inside tunnel through the adaptive generation of speed limits and information to the drivers approximating tunnels. The core of the system is a parallelized HFC implemented by means of a several interconnected FCAN (some of which adopting, in their turn, a hierarchical prioritized structure). Each FCAN has been conceived separately and designed in order to share fuzzy resources with the others.

FuzzJADE framework is released by LASA (LAboratorio Sistemi ad Agenti, Department of Mathematics and Informatics of the University of Salerno) in open-source software under the terms of the LGPL. FuzzJADE is a continuously evolving framework and this chapter belongs to a set of publications devoted to its evolution. At present, the distribution implements fully the described model. The version 2.0 of the distribution will be publicly available for download at <http://www.lasa.dmi.unisa.it/> on the first quarter of 2007. As future works, out of conducting activities devoted to make the framework more developer friendly, we plan to improve its underlying model by both exploring deeply its semantic dimension and developing tools allowing more complex and automated ontology-driven processes to distribute intelligence in a multiagent-based system.

References

- [1] W. Pedrycz. Granular computing: An introduction. In: *Proceedings of the 5th Joint Conference on Information Sciences*, Atlantic City, NJ, Vol. I, 2001, pp. 1349–1354.
- [2] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer Academic Publishers, Boston/Dordrecht/London, 2003.
- [3] JADE (Java Agent DEvelopment Framework) (TILAB, formerly CSELT). <http://jade.cselt.it/>, accessed November 2006.

- [4] F. Bellifemine, G. Caire, and T. Trucco (TILAB, formerly CSELT), G. Rimassa (University of Parma), JADE Programmer's GUIDE.
- [5] F. Bellifemine, G. Caire, and T. Trucco (TILAB S.p.A., formerly CSELT), G. Rimassa (FRAMeTech s.r.l.), R. Mungenast (PROFACTOR GmbH), JADE Administrator's GUIDE.
- [6] M. Wooldridge, N.R. Jennings, and D. Kinny. The Gaia methodology fro agent-oriented analysis and design. *JAAMAS* 3(3) (2000) 285–312.
- [7] P. Moraitsis, E. Petraki, and N. Spanoudakis. Engineering JADE agents with the Gaia methodology. In: R. Kowalszyk *et al.* (eds), *Agents Technologies, Infrastructures, Tools, and Applications for e-Services*, LNAI 2592. Springer-Verlag, Berlin, 2003, pp. 77–91.
- [8] P. Moraitsis and N. Spanoudakis. Combining Gaia and JADE for multi-agent systems design. 4th int. symp. from agent theory to agent implementation (AT2AI4). In: R. Trapp (ed.), *Proceedings of the 17th European Meeting on Cybernetics and Systems Research (EMCSR 2004)*, Australian Society for Cybernetic Studies. Vienna, Austria, April 13–16, 2004.
- [9] Foundation for Intelligent Physical Agents. *FIPA Agent Management Specification*, 2002. <http://www.fipa.org/specs/fipa00023/2004-03-18>, accessed November 2006.
- [10] Foundation for Intelligent Physical Agents. *FIPA SL Content Language Specification*, 2002. <http://www.fipa.org/specs/fipa00008/2002-12-06>, accessed November 2006.
- [11] G. Caire (TILAB, formerly CSELT), D. Cabanillas (Technical University of Catalonia - UPC), JADE Tutorial. Application-defined content languages and ontologies.
- [12] J.J. Odell, H. Van Dyke Parunak, and B. Bauer. Representing agent interaction protocols in UML. In: P. Ciancarini and M. Wooldridge (eds), *Agent-Oriented Software Engineering*, Springer, Berlin, 2001, pp. 121–140.
- [13] Foundation for Intelligent Physical Agents. *FIPA Communicative Act Library Specification*, 2000. <http://www.fipa.org/specs/fipa00037/2002-12-06>, accessed November 2006.
- [14] Foundation for Intelligent Physical Agents. *FIPA Query Interaction Protocol Specification*, 2002. <http://www.fipa.org/specs/fipa00027/2002-12-06>, accessed November 2006.
- [15] Foundation for Intelligent Physical Agents. *FIPA Request Interaction Protocol Specification*, 2002. <http://www.fipa.org/specs/fipa00026/2002-12-06>, accessed November 2006.
- [16] Foundation for Intelligent Physical Agents. *FIPA Subscribe Interaction Protocol Specification*, 2002. <http://www.fipa.org/specs/fipa00035/2002-12-06>, accessed November 2006.
- [17] V. Louis and T. Martinez. An operational model for the FIPA-ACL semantics. In: *Agent Communication Workshop*, AAMAS 2005, Utrecht University, Utrecht, The Netherlands, July 25–29, 2005.
- [18] V. Galdi, V. Loia, A. Piccolo, and M. Veniero. Fuzzy pro-active agents as key issue to increase traffic safety for next generation tunnels. In: *2007 IEEE International Conference on Fuzzy Systems (FuzzIEEE2007)*, Imperial College, London, UK, July 2007, Session: Fuzzy-based Agents for Ambient Intelligence Environments.
- [19] A. Sacripanti. SITI (Safety in Tunnel Intelligence): An Italian global project. In: *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, 2004, October 3–6, 2004, pp. 521–526.

45

Granular Models for Time-Series Forecasting

Marina Hirota Magalhães, Rosangela Ballini, and Fernando Antonio Campos Gomide

45.1 Introduction

Currently, there is a vast literature describing modeling approaches in the field of time-series forecasting for a variety of areas, such as medicine, finance, hydrology, and meteorology.

In practice, most techniques used to model time series assume linear relationships among the variables. Often, these techniques are based on the classical Box and Jenkins methodology [1], the autoregressive moving average (ARMA) model being among the ones widely adopted. Recently, neural-network-based models have emerged as attractive non-linear alternatives [2] because they are effective in approximating non-linear input–output relationships [3]. Hybrid models that combine neural networks and fuzzy systems have been suggested and shown to provide better results than classic regression models and neural networks [4–7]. Data clustering is a recent approach that has been proposed as a methodology to construct forecasting models [8–10]. Clustering-based forecasting models are potentially powerful because time series often show granular input–output relationships and clustering data is a form to construct granular relationships. Granular modeling is also useful to develop local non-linear functional relationships to produce global models combining local granular functional models [11].

Clustering-based forecasting models are developed in two phases. The first phase uses a clustering algorithm to obtain a set of data clusters. The goal here is to group time-series data showing similar behavior through an appropriate group structure found by the clustering algorithm. Clustering algorithms such as change-point detection methods using backpropagation neural networks [12] or variations of fuzzy c -means (FCM) [13] are often the choice. The second phase develops a prediction model for each cluster using either linear (typically ARMA) or non-linear (neural networks, fuzzy predictors) regression.

More generally, granular forecasting addresses approaches to form associations between information granules and generate results in the same granular instead of pure numeric format. In this sense, information granules, viewed as linguistic labels, induce models at higher abstraction levels for rapid prototyping [14]. Granular clustering [15] is an innovative idea introduced recently. Granulation of temporal data to develop global view on time series is critical to capture the essential features of time series [16]. Recursive information granulation involving temporal and spatial granulation is an essential step to process data streams [17]. In this vein, analysis of numeric data to analyze time-series data and extract time correlations among multiple time-series data streams at different granularities is a key issue [18]. Granular modeling through regression analysis developed within the framework of fuzzy relational equations and

possibility and necessity measures [19] is an approach for granular forecasting that seems particularly suitable to address generality and computational efficiency. A comprehensive introduction to granular computing and applications in data processing is found in [20].

In this chapter we first suggest a granular functional modeling approach using FCM and local, linear regression models. Next, we introduce a granular relational forecasting model based on fuzzy clustering. The granular forecasting model is constructed in two phases. The first phase uses FCM clustering and the second employs a classification scheme to predict time-series values anchored in similarities between historical and prediction data behavior. This approach differs from the ones suggested in [12] and [21] and from the granular functional modeling approach proposed here because predictions do not use local prediction models, but pattern recognition and classification procedures instead. In granular forecasting modeling, cluster structures are driven by mean absolute prediction error performance measure instead of combination of criteria such as the one suggested in [22]. This idea produces better cluster structures because it uses forecasting performance to group data, as opposed to the structure of data induced by the clustering algorithm itself.

After this introduction, the chapter proceeds as follows. The next section details the granular forecasting models for time series addressed in this chapter. Section 45.3 compares the granular functional forecasting model approach with the fuzzy prediction model developed in [13], a hybrid model based on state recognition and time-series prediction (HATSP) conceptually similar to the one introduced in this paper. Section 45.4 addresses the average monthly streamflow forecasting problem using data of a major hydroelectric power plant situated at the northeast of Brazil. Performance of the granular forecasting model is compared with the periodic autoregressive moving average (PARMA), the model currently adopted by the power industry. Comparisons with the granular functional forecasting, multilayer feedforward (MLP), and fuzzy neural network (FNN) models developed in [23] and [24] are also included. The FNN is among the most effective models for streamflow prediction. Section 45.5 concludes the chapter and summarizes issues for future developments.

45.2 Granular Forecasting Models

This section addresses the time-series prediction problem and details the granular forecasting models.

Given samples of a time series, $v_{t-1} \in \mathbb{R}$, $t = 1, \dots$, the aim is to estimate the value of v_t , using the information from a set of past values of v_t . \mathbb{R} is the set of real numbers. Thus, we deal with one-step-ahead prediction. Consider a set of N , $(l+1)$ -dimensional pattern vectors denoted hereafter as data patterns \mathbf{p}_j , for short, $j = 1, \dots, N$. Each \mathbf{p}_j is constructed using $(l+1)$ past values of the time-series samples as follows:

$$\mathbf{p}_j = \begin{bmatrix} v_{t-l}^j & v_{t-l+1}^j \cdots v_{t-1}^j & v_t^j \end{bmatrix} \quad j = 1, 2, \dots, N. \quad (1)$$

Alternatively, we may construct data pattern \mathbf{p}_j using, in addition to the l previous values v_{t-k} , $k = 1, \dots, l$, the $(l-1)$ corresponding slopes, that is, the first differences $(v_{t-k+1} - v_{t-k})$, $k = 2, \dots, l$. In this case, the data pattern \mathbf{p}_j become:

$$\mathbf{p}_j = \left[\left(v_{t-l+1}^j - v_{t-l}^j \right) \cdots \left(v_{t-1}^j - v_{t-2}^j \right) \quad v_{t-l}^j \cdots v_{t-1}^j \quad v_t^j \right] \quad j = 1, 2, \dots, N. \quad (2)$$

Higher order differences can also be adopted to construct data patterns, but here we emphasize the first differences case only.

Granular forecasting models are built in two phases. The first phase uses FCM to cluster data patterns \mathbf{p}_j , $j = 1, \dots, N$. We recall that the FCM is essentially an iterative optimization technique to find cluster centers expected to characterize the relevant classes in a finite set of data. The classes form a fuzzy partition of the data set. The FCM uses a performance index identified in terms of cluster centers. The performance index measures the weighted sum of the distances between cluster centers and data in the corresponding fuzzy clusters. The algorithm assumes that the desired number of clusters M is given and starts selecting an initial fuzzy partition expressed in the form of a membership matrix $U = [\mu_{ij}]$. Entries μ_{ij} denote the membership degrees of data patterns \mathbf{p}_j in the i th cluster and are such that $\sum_{i=1}^M \mu_{ij} = 1$.

Next, cluster centers $c_i \in \Re^{l+1}$ ($c_i \in \Re^{2l}$ if we adopt data patterns as in (2)) are computed using

$$c_i = \frac{\sum_{j=1}^N (\mu_{ij})^m \mathbf{p}_j}{\sum_{j=1}^N (\mu_{ij})^m}, \quad i = 1, 2, \dots, M, \quad (3)$$

and the fuzzy partition updated as follows:

$$\mu_{ij} = \left[\sum_{k=1}^M \left(\frac{\|\mathbf{p}_j - c_i\|^2}{\|\mathbf{p}_j - c_k\|^2} \right)^{\frac{2}{m-1}} \right]^{-1}, \quad (4)$$

where $\|\cdot\|$ denotes Euclidean norm.

These steps continue until there is no significant change between the current and the previous fuzzy partitions. The parameter $m > 1$ defines the fuzziness of the partition (see [13,25] for more details about FCM).

This chapter concerns time-series prediction and, in this context, uses a cluster validity measure that differs from the usual combination of criteria as, e.g., in [22]. More specifically, here cluster validity is evaluated using the mean absolute prediction error. As it will be explained in the next section, the lower the mean absolute prediction error, the better the cluster structure.

The second phase of granular forecasting modeling classifies prediction data patterns $\mathbf{p}_q, q = 1, \dots, P$, called hereafter prediction patterns for short, according to the fuzzy clusters found in the first phase. The two approaches introduced in this chapter differ in the way the second phase is done. The first approach develops a functional forecasting model, while the second develops a granular forecasting model.

The general scheme to build granular forecasting models is shown in Figure 45.1.

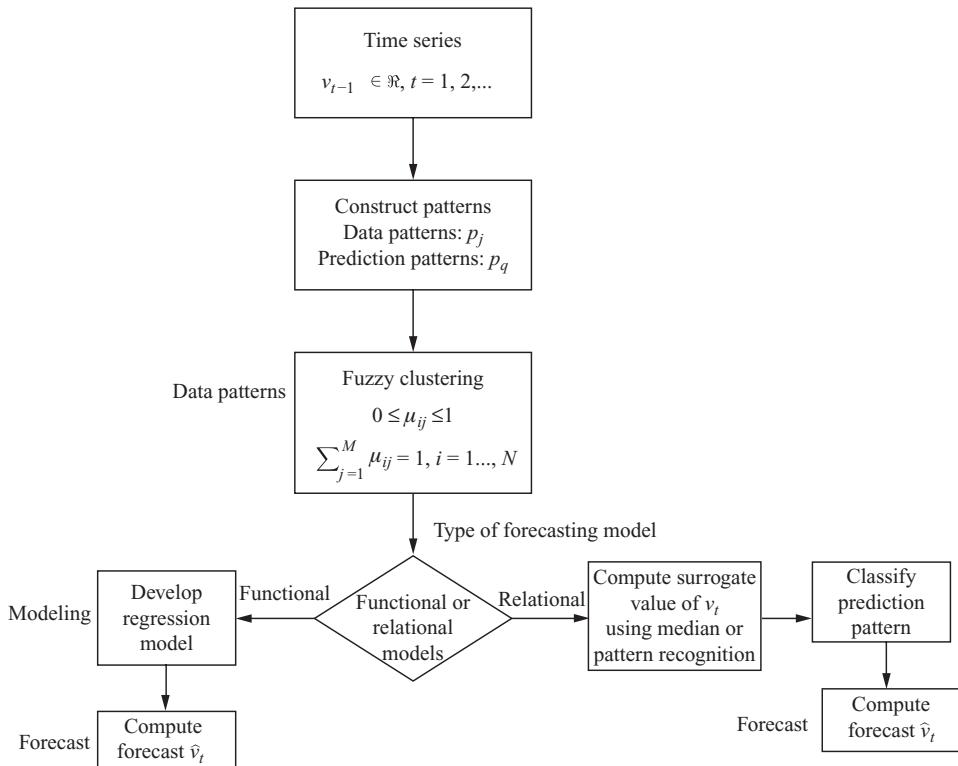


Figure 45.1 Granular forecasting modeling

45.2.1 Granular Functional Forecasting

Granular functional modeling for time-series forecasting uses FCM to group data patterns and develops a local model for each group. The local model can take any appropriate form such as linear functions, polynomials, or neural networks. Here we emphasize, without loss of generality, the simplest linear autoregressive case. The method includes the following steps.

First, a regression model $f_i(\mathbf{p}, \mathbf{a})$ is fitted to each cluster using data patterns \mathbf{p}_j as a training set. Function $f_i(\mathbf{p}, \mathbf{a})$ has \mathbf{a} as a vector of parameters, namely, the coefficients of the linear model. Next, (l) -dimensional prediction patterns \mathbf{p}_q are built using known values of the variable at $(t - k)$, $k = 1, \dots, l$. More precisely, prediction patterns \mathbf{p}_q are assembled as follows:

$$\mathbf{p}_q = [v_{t-l}^q \ v_{t-l+1}^q \ \dots \ v_{t-1}^q] \quad q = 1, 2, \dots, P. \quad (5)$$

The membership degree u_{iq} of prediction patterns \mathbf{p}_q in the i th cluster, $i = 1, \dots, M$, is found and the forecast value \hat{v}_t^q computed as follows:

$$\hat{v}_t^q = \sum_{i=1}^M u_{iq} f_i(\mathbf{p}_q, \mathbf{a}). \quad (6)$$

Algorithm 1 summarizes the steps to develop granular functional forecasting models.

Algorithm 1. Granular Functional Forecasting (GFM).

Input samples of a time series v_1, v_2, \dots , and choose the value l .

1. Construct data patterns, \mathbf{p}_j , $j = 1, \dots, N$.
2. Choose the number of clusters M , $1 < M < N$; run the FCM algorithm to cluster the \mathbf{p}_j data patterns. Save clusters centers $C = [c_1 \ \dots \ c_i \ \dots \ c_M]^T$.
3. Construct prediction patterns \mathbf{p}_q , $q = 1, \dots, P$.
4. Compute the membership degree u_{iq} of each prediction pattern \mathbf{p}_q using the expression:

$$u_{iq} = \left[\sum_{k=1}^M \left(\frac{\|\mathbf{p}_q - c_i\|^2}{\|\mathbf{p}_q - c_k\|^2} \right)^{\frac{2}{m-1}} \right]^{-1}.$$

5. Compute forecasted value \hat{v}_t^q using

$$\hat{v}_t^q = \sum_{i=1}^M u_{iq} f_i(\mathbf{p}_q, \mathbf{a}).$$

45.2.2 Granular Relational Forecasting

Similarly as GFM, a prediction step produces a forecast for each prediction pattern \mathbf{p}_q , $q = 1, \dots, P$. Here it is important to emphasize that prediction pattern \mathbf{p}_q is constructed equally as \mathbf{p}_j except that \mathbf{p}_q should have, at its last component, the value v_t to be forecasted. Prediction patterns \mathbf{p}_q are $(l + 1)$ -dimensional vectors built either using known values of the variable at $(t - k)$, $k = 1, \dots, l$,

$$\mathbf{p}_q = [v_{t-l}^q \ v_{t-l+1}^q \ \dots \ v_{t-1}^q \ v_t] \quad q = 1, 2, \dots, P \quad (7)$$

or, alternatively, using known $(t - k)$ values plus the $(l - 1)$ corresponding slopes, respectively,

$$\mathbf{p}_q = [(v_{t-l+1}^q - v_{t-l}^q) \ \dots \ (v_{t-1}^q - v_{t-2}^q) \ v_{t-l}^q \ \dots \ v_{t-1}^q \ v_t] \quad q = 1, 2, \dots, P. \quad (8)$$

Note that in v_t we omit the upper index q to differentiate v_t from known past values v_{t-k}^q , $k = 1, \dots, l$.

The prediction step uses (4) to compute u_{iq} , the membership degree of the prediction pattern \mathbf{p}_q in the i th cluster. The forecasted value of v_t , \hat{v}_t , is found as a weighted combination of the cluster centers as follows:

$$\begin{aligned}\hat{v}_t &= u_{1q}c_{1h} + \cdots + u_{Mq}c_{Mh} \\ &= \sum_{i=1}^M u_{iq}c_{ih},\end{aligned}\tag{9}$$

where $h = (l + 1)$ if data patterns are constructed as in (1) or $h = 2l$ if data patterns as in (2) are adopted.

A key point to notice here is that, since the last component of the prediction pattern \mathbf{p}_q in (7) is the value to be predicted, a mechanism must be found to replace v_t during the classification phase. In other words, during classification the value of v_t in \mathbf{p}_q must be replaced by an appropriate surrogate. In this chapter we introduce two mechanisms to obtain surrogates. The first uses a straightforward procedure: the surrogate is taken as the median of the known v_t^j , $j = 1, 2, \dots, N$ values, namely, the median recognition procedure (MRP). The second mechanism takes into account the first l components of the data and prediction patterns to compose, respectively, l -data (\mathbf{s}_j) and l -prediction (\mathbf{s}_q) patterns. Pattern \mathbf{s}_q is matched against the N patterns \mathbf{s}_j to find \mathbf{s}_j^* , the one that is closest to \mathbf{s}_q . The surrogate value of v_t is taken as the value of v_t^j of the data pattern \mathbf{p}_j corresponding to \mathbf{s}_j^* . This mechanism is a form of pattern recognition procedure (PRP). The median (GRM-MRP) and pattern recognition (GRM-PRP) procedures are detailed in the next section.

Median Recognition Procedure (GRM-MRP)

Statistical measures such as mean, mode, and median are frequently adopted to summarize data. The median is particularly useful because it is the middle value in an ordered sequence of data. If there are no ties, half of the observations will be smaller and half will be larger. The median is unaffected by any extreme observations in a set of data. Thus, whenever significant variation is present, it is appropriate to use the median rather than the mean to describe a set of data. The MRP uses the median of the historical values of v_t as its surrogate in the prediction pattern \mathbf{p}_q . Algorithm 2 details the GRM-MRP modeling steps.

The first step of Algorithm 2 constructs data and prediction patterns \mathbf{p}_j and \mathbf{p}_q . Therefore we must decide if the data patterns should incorporate first differences or time-series samples only. This means to select h and choose the value of l of the data patterns.

The second step needs the value of M , the number of clusters, to run the FCM algorithm. It is well known that the task of finding the optimal value of M is currently an open problem in the literature, despite significant efforts made in this direction [25]. Since forecasts provided by Algorithm 2 depend on the cluster structure, the choice of M is vital.

Algorithm 2. Granular Relational Forecasting (GRM-MRP).

Input samples of a time series v_1, v_2, \dots , and choose the value h ,

1. *Construct data and prediction patterns, \mathbf{p}_j and \mathbf{p}_q .*

1.1 *Compute the median of v_t^j , $j = 1, \dots, N$.*

1.2 *Use the median as a surrogate value of v_t in prediction pattern \mathbf{p}_q .*

2. *Choose the number of clusters M , $1 < M < N$; run the FCM algorithm to cluster \mathbf{p}_j , $j = 1, \dots, N$, data patterns.*

Save clusters centers $C = [c_1 \ \dots \ c_i \ \dots \ c_M]^T$.

3. *Compute the membership degree u_{iq} of each prediction pattern \mathbf{p}_q using the expression:*

$$u_{iq} = \left[\sum_{k=1}^M \left(\frac{\|\mathbf{p}_q - c_k\|^2}{\|\mathbf{p}_q - c_i\|^2} \right)^{\frac{2}{m-1}} \right]^{-1}.$$

4. Compute forecasted value \hat{v}_t using

$$\hat{v}_t = \sum_{i=1}^M u_{iq} c_{ih}.$$

In this chapter we suggest the use of the mean absolute prediction error (*MAPE*) to experimentally choose l , h , and M . The *MAPE* is defined as follows

$$MAPE = \frac{1}{P} \sum_{k=1}^P |v_k - \hat{v}_k|, \quad (10)$$

where P is the number of predictions, \hat{v}_k is the forecast, and v_k is the actual observed value.

In general, the nature of the problem hints a value for h . For instance, in streamflow forecast, a problem we address in a later section, slopes are important to capture the transition between dry and wet periods. Therefore, in this case data patterns should adopt $h = 2l$. The value of l can be estimated from correlation analysis or *MAPE*. Once the h and l are chosen, an appropriate value of M can be found, computing *MAPE* for different values of M . We adopt the value of M for which *MAPE* is the lowest. After h , l , and M are chosen, the FCM algorithm is run and the clusters centers stored. The underlying clustering algorithm may utilize an augmented version of the FCM technique, the context-based clustering [25], to assign cluster centers in data space regions in which prediction error are unacceptable. The use of *MAPE*, therefore, translates into a context-based FCM (CFCM) mechanism.

The last step of the algorithm finds the membership degrees of the prediction pattern \mathbf{p}_q and computes the forecasted value using (9), which is a weighted combination of the cluster centers.

Pattern Recognition Procedure (GRM-PRP)

The main idea of the PRP procedure is to use information about the trend of the prediction pattern \mathbf{p}_q to select a value in the historical database that is as close as possible to the actual value of v_t . Trend information is used with the help of h -dimensional r -data and r -prediction patterns, \mathbf{s}_j and \mathbf{s}_q , respectively. The r -data pattern can be constructed using two alternative ways. The first considers only r samples of past values

$$\mathbf{s}_j = [v_{t-r}^j \ v_{t-r+1}^j \ \cdots \ v_{t-1}^j], \quad j = 1, \dots, N, \quad (11)$$

and the second consider r past values and $(r-1)$ corresponding slopes assembled in a $(2r-1)$ dimensional vectors

$$\mathbf{s}_j = [(v_{t-r+1}^j - v_{t-r}^j) \cdots (v_{t-1}^j - v_{t-2}^j) \ v_{t-r}^j \cdots v_{t-1}^j], \quad j = 1, \dots, N. \quad (12)$$

Similarly, the r -prediction patterns can be constructed as either

$$\mathbf{s}_q = [v_{t-r}^q \ v_{t-r+1}^q \ \cdots \ v_{t-1}^q], \quad q = 1, \dots, P, \quad (13)$$

or

$$\mathbf{s}_q = [(v_{t-r+1}^q - v_{t-r}^q) \cdots (v_{t-1}^q - v_{t-2}^q) \ v_{t-r}^q \cdots v_{t-1}^q], \quad q = 1, \dots, P. \quad (14)$$

After constructing the r -patterns, the next step is to match the r -prediction pattern \mathbf{s}_q against the N data patterns \mathbf{s}_j . Here we choose the Euclidean norm as a matching measure. The best match is the \mathbf{s}_j closest to \mathbf{s}_q ; that is,

$$\|\mathbf{s}_q - \mathbf{s}_j^*\| = \min_{1 \leq j \leq N} \|\mathbf{s}_q - \mathbf{s}_j\|. \quad (15)$$

Thus, \mathbf{s}_j^* is the r -data pattern closest to \mathbf{s}_q . Once \mathbf{s}_j^* is found, we look for the corresponding data pattern \mathbf{p}_j^* and take the value of v_i^j at the last component of vector \mathbf{p}_j^* as the surrogate of v_t in the prediction

pattern \mathbf{p}_q . The choice of M in step 2 follows the same procedure as in GRM-MRP. Algorithm 3 details the GRM-PRP steps.

In Algorithm 3, h is either $h = r$ or $h = 2r - 1$, depending on if we construct patterns using (11), (12), (13), or (14). Selection of h and r proceeds similarly as in GRM-MRP.

Next section compares the granular functional forecasting model (GFM) with state recognition and time-series prediction (HATSP) model addressed in [22].

Algorithm 3. Granular Relational Forecasting (GRM-PRP).

Input samples of a time series v_1, v_2, \dots , and choose the value of h .

1. Construct data and prediction patterns, \mathbf{p}_j and \mathbf{p}_q .
2. Choose the number of clusters M , $1 < M < N$; run the FCM algorithm.
Save clusters centers $C = [c_1 \cdots c_i \cdots c_M]^T$.
3. Construct r -data and prediction patterns.
4. Given a r -prediction pattern \mathbf{s}_q , compute the r -data pattern \mathbf{s}_j^* , such that

$$\|\mathbf{s}_q - \mathbf{s}_j^*\| = \min_{1 \leq j \leq N} \|\mathbf{s}_q - \mathbf{s}_j\|.$$

5. Use the value at the last component of \mathbf{p}_j^* as the surrogate value for v_t in \mathbf{p}_q .

6. Compute the membership degrees of the prediction pattern \mathbf{p}_q using

$$u_{iq} = \left[\sum_{k=1}^M \left(\frac{\|\mathbf{p}_q - c_i\|^2}{\|\mathbf{p}_q - c_k\|^2} \right)^{\frac{2}{m-1}} \right]^{-1}.$$

7. Compute forecasted values \hat{v}_t using

$$\hat{v}_t = \sum_{i=1}^M u_{iq} c_{ih}.$$

45.3 Comparison between GFM and HATSP Model

In this section we consider the fuzzy prediction model HATSP developed in [22] to evaluate the performance of GFM. HATSP uses a similar idea as the one adopted in this chapter. Briefly, the algorithm in [22] is a two-phase hybrid prediction algorithm for state recognition and time-series prediction, HATSP.

The first phase of HATSP clusters data patterns similarly as in GFM, but uses a hierarchical unsupervised fuzzy clustering (HUFC) algorithm [26] instead of FCM. The second phase develops a local prediction model (autoregressive model or neural network) for each cluster. Predictions are the forecasted values derived from the local models weighted by membership degrees (see [21] for further details).

Following [22], we employ the chaotic time series of one dimension generated by the logistic mapping

$$v_t = 4v_{t-1}(1 - v_{t-1}), \quad (16)$$

where $t = 1, \dots$. The experiment presented in [21] uses the first 900 points as training data and the next 200 data points as the test set. We adopt the same procedure to perform the experiments reported next.

Figure 45.2 illustrates the use of the GFM with the standard FCM (Figure 45.2a) and its context-based counterpart (Figure 45.2b), namely, CFCM [25]. Both use 25 clusters, but we notice that cluster centers are placed differently in the data space. Using 25 clusters, the GFM with FCM achieves $NMSE = 3.9909 \times 10^{-4}$, while with CFCM it reaches $NMSE = 2.9112 \times 10^{-5}$.

Similarly as in HATSP, we adopt the normalized mean square error ($NMSE$) to evaluate predictions and observe how prediction errors influence the number of clusters. To make comparison meaningful, in this section we use $NMSE$ instead of $MAPE$ to find the number of clusters. The $NMSE$ is defined as follows:

$$NMSE = \frac{\sum_{k=1}^P (v_k - \hat{v}_k)^2}{\sum_{k=1}^P (v_k - \bar{v})^2}, \quad (17)$$

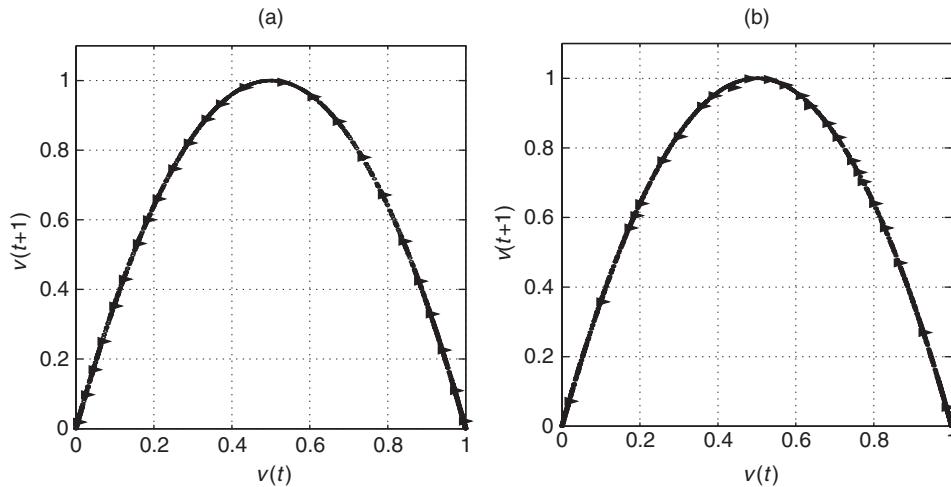


Figure 45.2 Chaotic time-series data set and cluster centers, using (a) FCM and (b) CFCM clustering

where P is the number of predictions, \hat{v}_k is the forecast, v_k is the desired output and \bar{v} the respective mean value. Note that $NMSE = 1$ means to predict the average.

The number of clusters vary from $c = 2$ to $c = 101$. Figure 45.3 shows that, from the $NMSE$ criterion point of view, prediction error of GFM decreases as the number of clusters increases.

HATSP uses 101 clusters [21] to reach $NMSE = 6.11 \times 10^{-7}$, while with 101 clusters GFM with FCM achieves $NMSE = 2.11 \times 10^{-6}$. With CFCM, GFM reaches $NMSE = 8.01 \times 10^{-7}$ with 70 clusters only, a significant improvement.

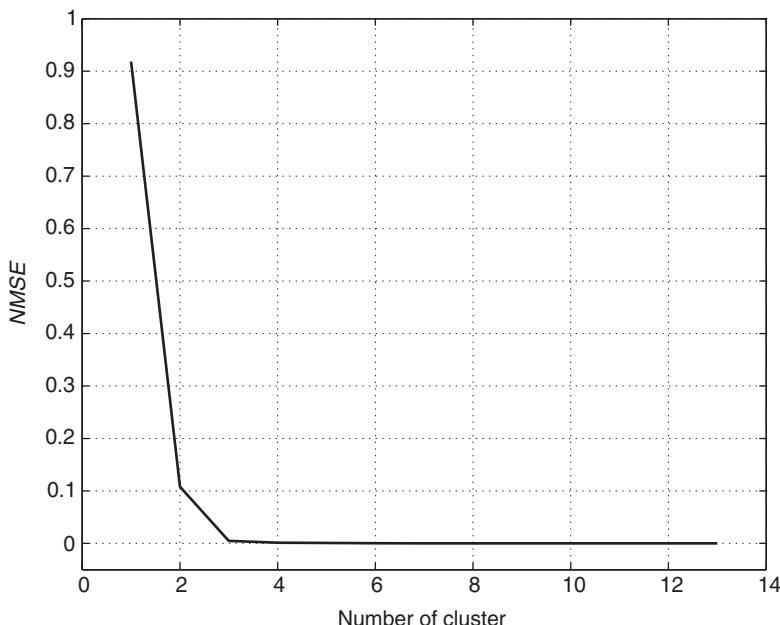


Figure 45.3 Normalized mean square error of GFM for the logistic map

Similarly as HATSP, GFM with both FCM and CFCM do require development of local models for each cluster. In the case of linear autoregressive models HATSP, GFM with FCM and CFCM need to compute M pseudoinverse matrices to find models coefficients before a forecast value is produced.

Next section addresses the use of granular forecasting GFM, granular relational GRM with both, pattern recognition procedure GRM-PRP, and median recognition procedure GRM-MRP models to forecast monthly averages of natural streamflow time series. In addition to GFM, GRM-PRP and GRM-MRP are compared with three alternative models: the classic PARMA [27], MLP, and FNN [23].

45.4 Seasonal Streamflow Forecasting

45.4.1 The Problem

Planning and operation of water resources and energy systems is a complex and difficult task once it involves non-linear production characteristics and depends on numerous variables. A key variable is the natural streamflow. Streamflow values covering the entire planning period must be accurately forecasted because they strongly influence production planning. In energy systems planning, short- and long-term forecasts of streamflow are mandatory for simulation, optimization, and decision making.

Most hydroelectric systems involve geographically distinct regions and have hydrometric data collected through a very sparse and different data acquisition networks. This results in uncertainties in the hydrological information available. Furthermore, the inherently non-linear relationship between input and output flow complicates streamflow forecast considerably. Another difficulty in streamflow forecasting concerns its non-stationary nature due to wet and dry periods over the year [3]. Generally, wet periods (from October to March in Brazil) present higher streamflow variability, a challenge for streamflow time-series modeling and prediction.

45.4.2 Application in Streamflow Forecasting

In this section we use the fuzzy prediction model to forecast average monthly inflows for a large hydroelectric plant, namely, Sobradinho, situated at northeast of Brazil. Hydrologic data covering the period from 1931 to 1990 are used to construct patterns and clusters, and data from 1991 to 1998 used to test and compare the performance of different algorithms.

The inflows oscillate between minimum and maximum values following the seasonal variation during the 12-month period. The seasonality of the flows justifies the use of 12 different models, one for each month of the year, as currently adopted by many hydrological systems worldwide. Therefore, 12 fuzzy clustering prediction models were developed to forecast monthly inflow averages.

Let us assume, as an example, that our aim is to forecast the streamflow values for $t = 9$ (September). Consider a 68 (1931–1998) historical streamflow data set normalized within $[-1, 1]$ as follows:

$$v^j = 2 \times \left(\frac{v^j - \min}{\max - \min} \right) - 1, \quad (18)$$

where \min and \max represent, respectively, the minimum and maximum of the streamflow values. The first 60 (1931–1990) data are used for modeling and the remaining 8 (1991–1998) for testing.

For GFM and GRM-MRP, data patterns (with $l = 1$, chosen using $MAPE$) of the form

$$\mathbf{p}_j = \begin{bmatrix} v_8^j & v_9^j \end{bmatrix} \quad (19)$$

are constructed and clustered. Therefore, in this example data patterns are two-dimensional vectors whose components are samples of August and September streamflows averages v_8^j and v_9^j . For the given data set, the appropriate number of clusters was chosen, varying $M = 1, \dots, 10$. $M = 4$ is the value for which $MAPE$ (10) is the lowest. The result is shown in Figure 45.4.

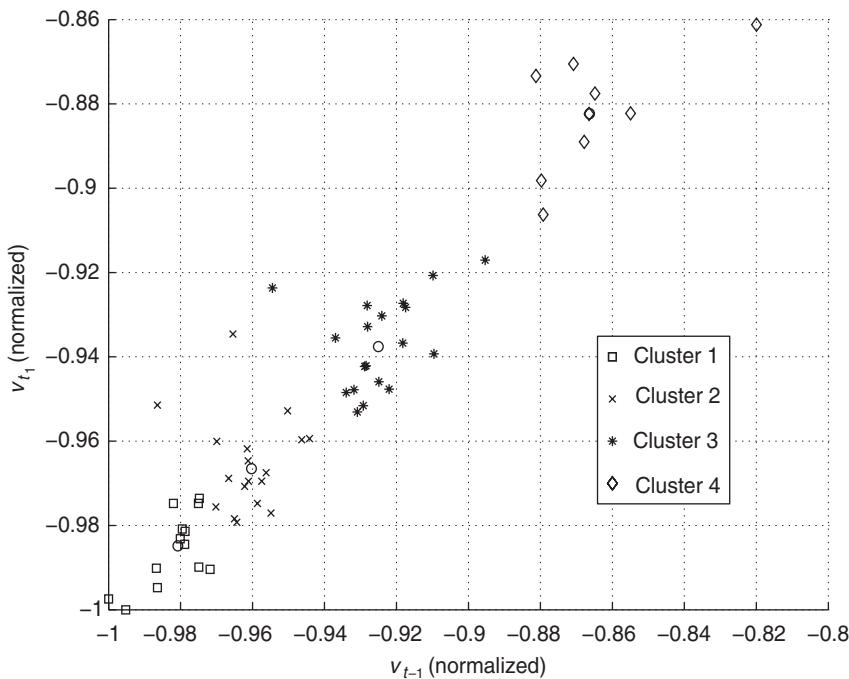


Figure 45.4 Clusters found using FCM for September: Circles denote cluster centers

During the classification phase, prediction patterns are constructed using (7); that is,

$$\mathbf{p}_q = [v_8^q \quad v_9]. \quad (20)$$

In GRM-MRP, the median of the v_9^j values, $j = 1, \dots, 60$ (1931–1990), is chosen as the surrogate value for v_9 of the prediction pattern for September 1991 ($q = 1$). After classification, using the fuzzy clusters found in the first phase, we compute the forecasted value for v_9 using (9):

$$\begin{aligned} \hat{v}_9 &= u_{1,61}c_{1,2} + \dots + u_{4,61}c_{4,2} \\ &= \sum_{i=1}^4 u_{i,61}c_{i,2}. \end{aligned} \quad (21)$$

Here, $M = 4$, $q = 61$, and $l = 1$. This scheme is repeated for each September within the period 1992–1998 ($q = 2, \dots, 8$). The result is depicted in Figure 45.5.

GRM-PRP uses data patterns \mathbf{p}_j given in (19). They are clustered similarly as in MRP. In this case, the number of clusters suggested using MAPE is $M = 8$. The result is shown in Figure 45.6.

During classification, MAPE indicates $r = 2$ to construct r -data and r -prediction patterns. The patterns are as follows (see (12) and (14)):

$$\mathbf{s}_j = \left[\begin{pmatrix} v_8^j & v_7^j \end{pmatrix} \quad v_7^j \quad v_8^j \right], \quad (22)$$

$$\mathbf{s}_q = \left[\begin{pmatrix} v_8^q & v_7^q \end{pmatrix} \quad v_7^q \quad v_8^q \right]. \quad (23)$$

Slopes are considered to forecast September inflows because, according to MAPE, first-order differences improve prediction. Interestingly, usually September is a transition period between dry and wet seasons. To forecast September 1991 ($q = 1$), we use August and July of 1991 inflow averages plus

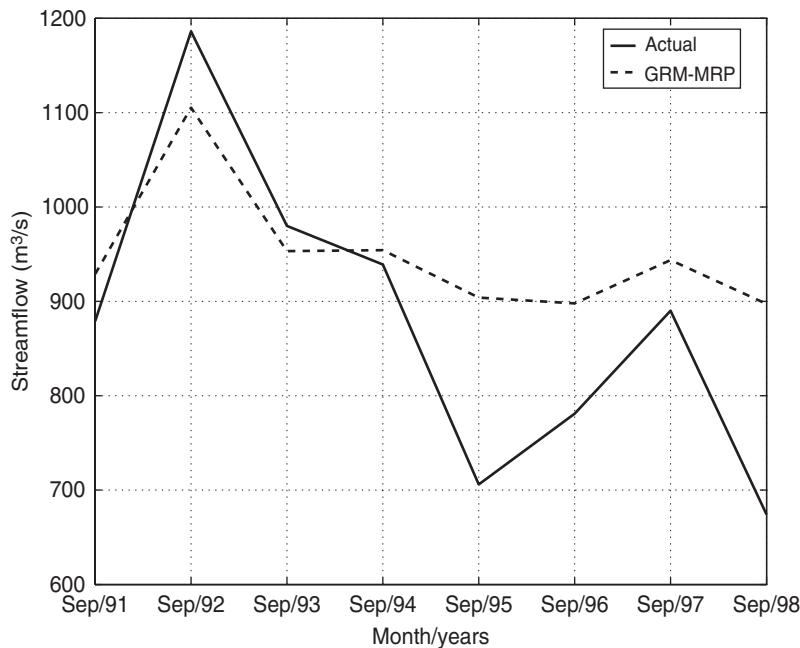


Figure 45.5 MRP streamflow prediction for September: Solid and dashed lines are actual and predicted values

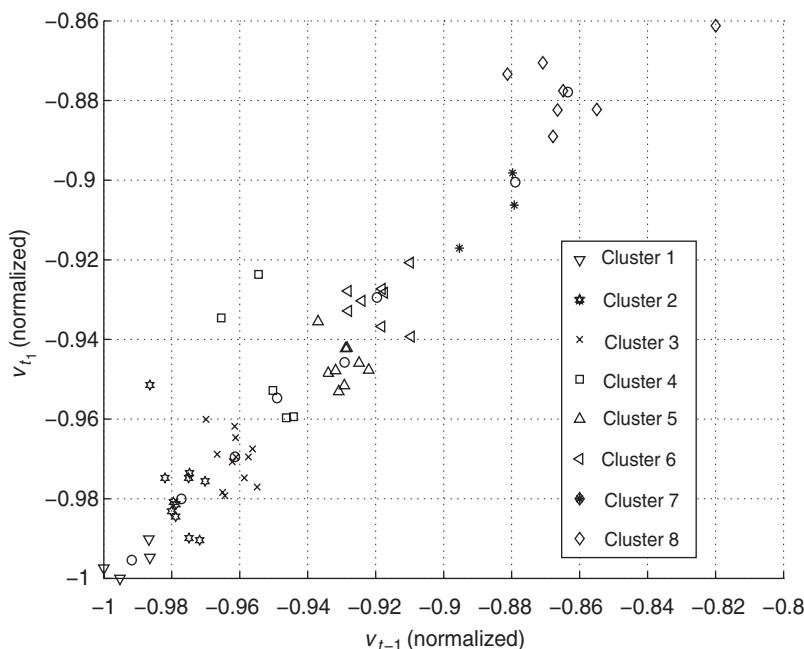


Figure 45.6 Clusters found using FCM for September: Circles denote cluster centers

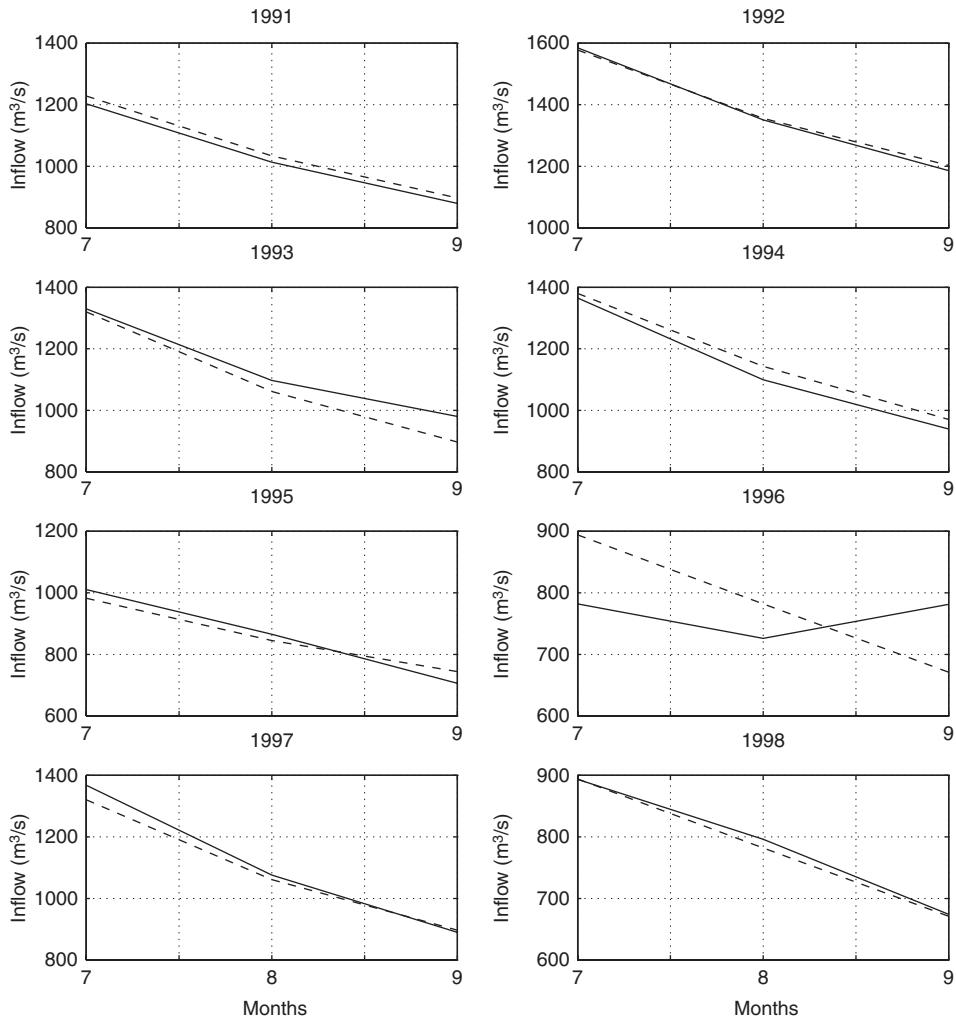


Figure 45.7 Streamflow trends: Solid and dashed lines represent values of patterns \mathbf{p}_q and \mathbf{p}_j , respectively

the respective slopes to construct the corresponding r -data and r -prediction patterns and employ (15) to find the closest \mathbf{s}_j among $N = 60$ r -data patterns. The result is depicted in Figure 45.7, where we note the actual data contained in prediction pattern and the closest data pattern \mathbf{s}_j^* found. In Figure 45.7, September streamflow values (v_9^j) are added to verify if the surrogate is a reasonable approximation of v_9^j . Clearly, the surrogate values found are, except for 1996, very close to the actual ones. Recall that the surrogate value of v_9 in the prediction pattern is the one that follows v_8^* in \mathbf{p}_j^* .

After a surrogate for v_9 is found, the prediction pattern is classified according to the fuzzy clusters. Using (9), the predicted value for September of 1991 ($q = 61$) is

$$\begin{aligned}\widehat{v}_9 &= u_{1,61}c_{1,2} + \cdots + u_{8,61}c_{8,2} \\ &= \sum_{i=1}^8 u_{i,61}c_{i,2}.\end{aligned}\tag{24}$$

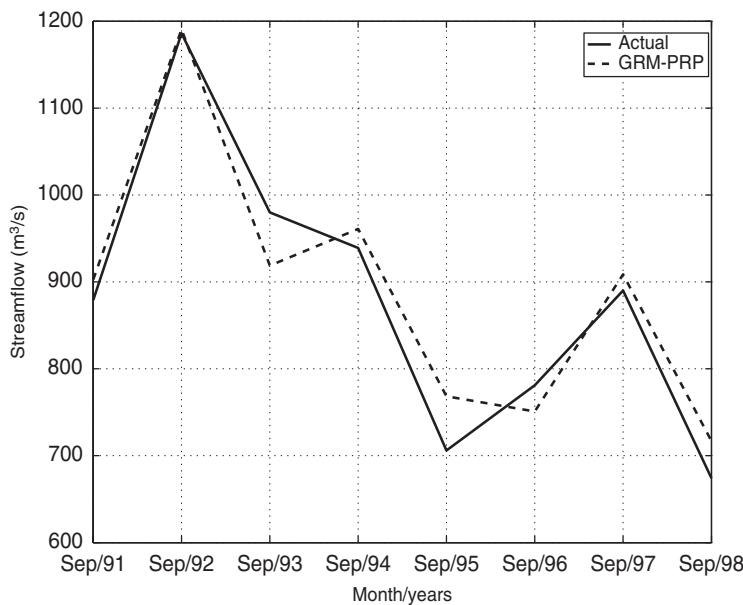


Figure 45.8 PRP streamflow prediction for September: Solid and dashed curves represent actual and predicted values, respectively

These steps are repeated to forecast the remaining Septembers ($q = 62, \dots, 68$, corresponding to 1992–1998 period). The result is shown in Figure 45.8.

The procedure just detailed is repeated for each month, from January to December. The number of clusters (M) and the values of r required by GRM-PRP, GRM-MRP, and GFM are summarized in Table 45.1.

It is interesting to note that, for Sobradinho reservoir, March and September are transitions between wet and dry periods. March values are preceded by high streamflow values during December, January, and February, followed by decreasing values during April, May, and June. Thus, the respective streamflow variations (slopes) are positive before March and negative after March. Hence, surrogate values for v_t

Table 45.1 Monthly characteristics of GRM-MPR, GRM-PRP, and GFM

Months	GRM-PRP			GRM-MRP	GFM
	M	r	Slope inclusion	M	M
Jan	2	3	No	2	2
Feb	3	3	No	6	2
Mar	4	3	Yes	3	2
Apr	7	3	No	5	2
May	8	3	No	8	2
Jun	10	2	No	9	5
Jul	8	1	No	5	2
Aug	8	1	No	4	6
Sep	8	2	Yes	4	6
Oct	5	2	No	2	2
Nov	4	2	No	2	2
Dec	3	2	No	2	6

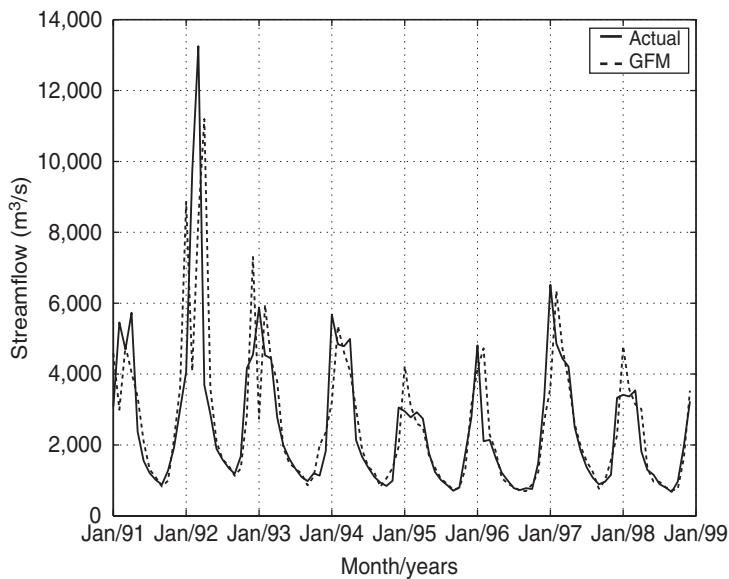


Figure 45.9 GFM predictions for 1991–1998: Continuous and dashed lines are actual and forecasted values

are influenced by slopes. September has opposing behavior: streamflow variations decrease during July and August and increase during October and November.

Forecasts produced by GFM, PARMA, Feed forward MLP, FNN, GRM-MRP, and GRM-PRP models are shown in Figures 45.9, 45.10, 45.11, 45.12, 45.13, and 45.14, respectively. Similarly, GRM, GFM, PARMA, MLP, and FNN models were developed for each month. Characteristics and parameters of the algorithms and global prediction error values are given in Tables 45.2 and 45.3, respectively.

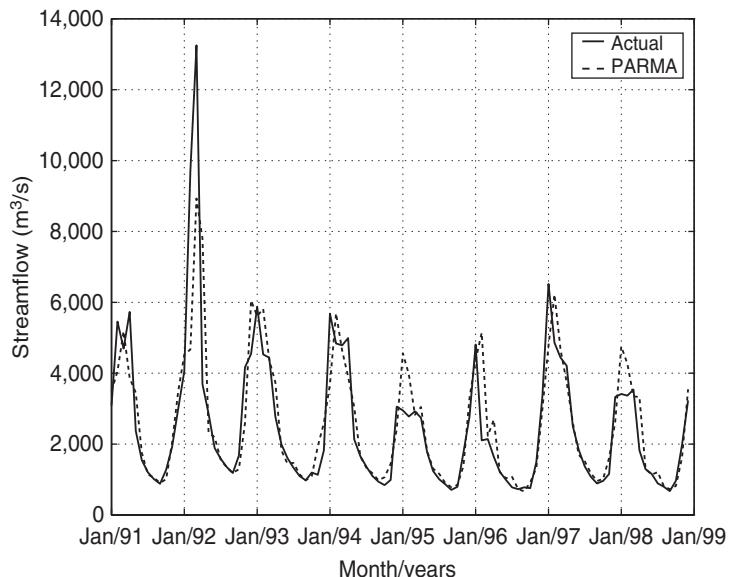


Figure 45.10 PARMA predictions for 1991–1998. Continuous and forecasted values

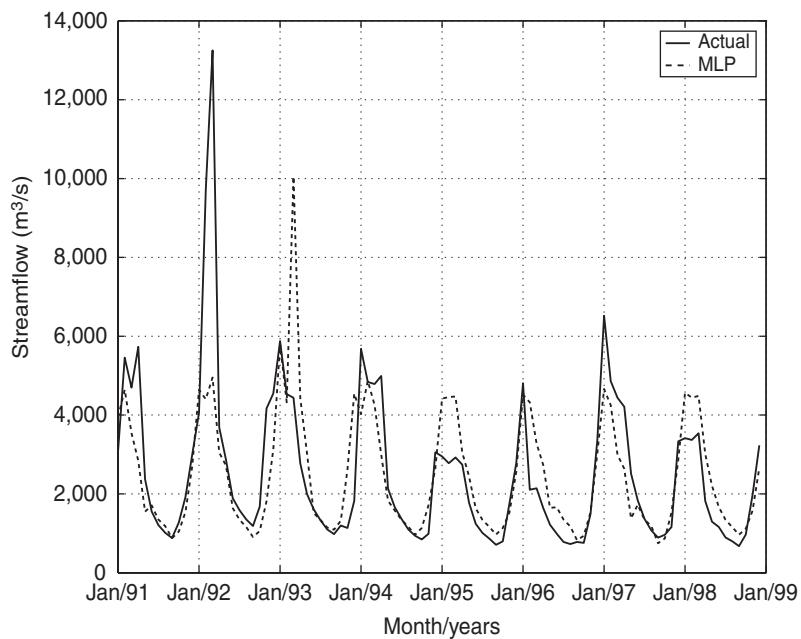


Figure 45.11 MLP predictions for 1991–1998: Continuous and dashed lines are actual and forecasted values

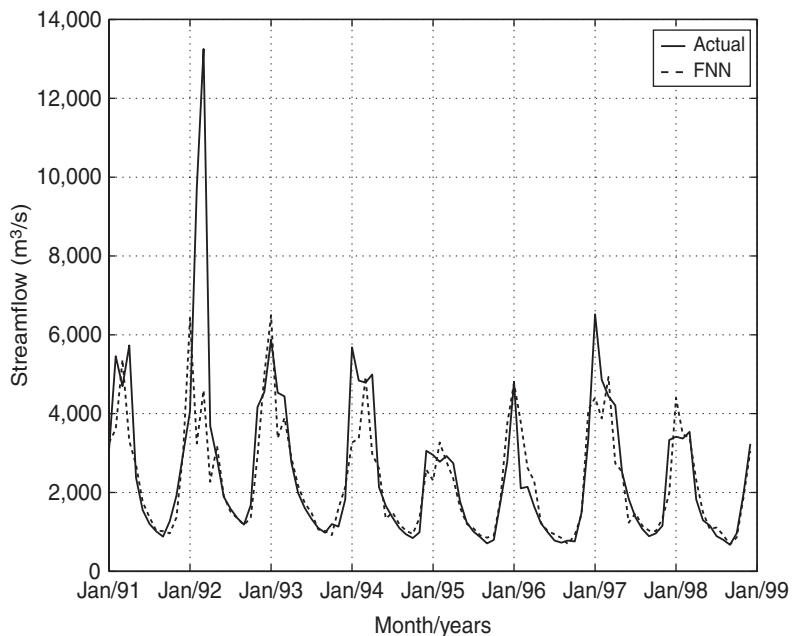


Figure 45.12 FNN predictions for 1991–1998: Continuous and dashed lines are actual and forecasted values

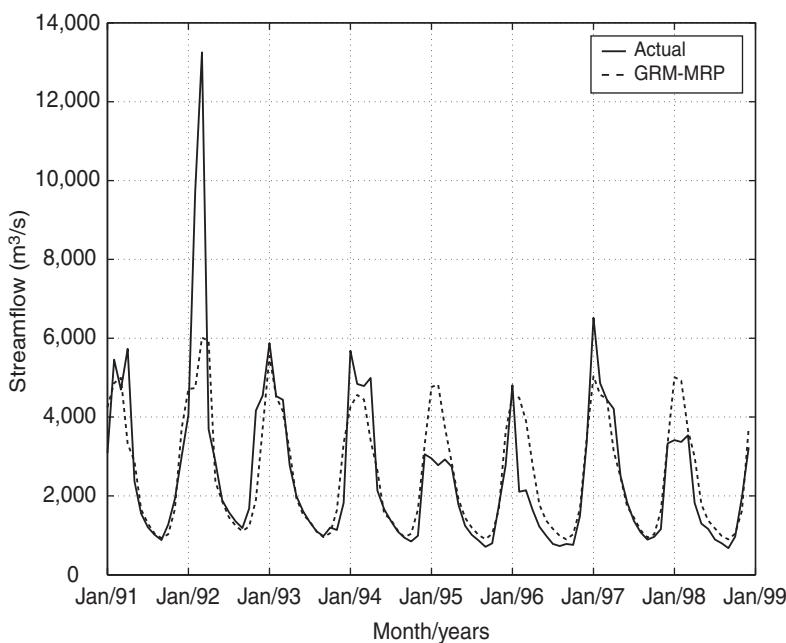


Figure 45.13 GRM–MRP predictions for 1991–1998: Continuous and dashed lines are actual and forecasted values

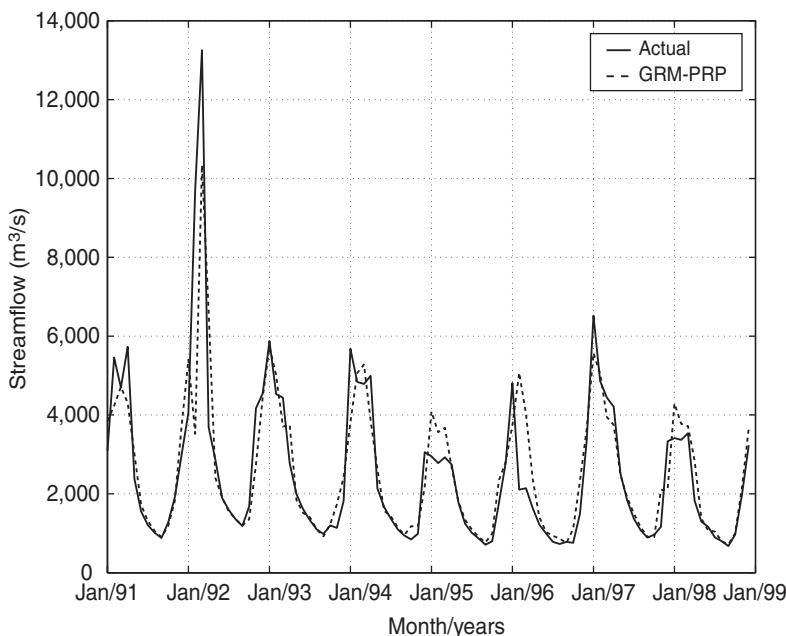


Figure 45.14 GRM-PRP predictions for 1991–1998: Continuous and forecasted values

Table 45.2 Monthly characteristics of PARMA, MLP, and FNN models

Months	PARMA	MLP		FNN	
	Order (p_m, q_m)	Number of inputs	Number of neurons	Number of inputs	Rules
Jan	(1, 0)	2	7	2	49
Feb	(1, 0)	6	31	6	48
Mar	(1, 0)	1	14	1	47
Apr	(1, 1)	3	28	3	50
May	(1, 0)	3	6	3	41
Jun	(3, 0)	6	9	6	46
Jul	(2, 1)	2	18	2	44
Aug	(1, 0)	2	23	2	47
Sep	(1, 0)	2	30	2	44
Oct	(1, 0)	3	33	3	48
Nov	(1, 1)	3	30	3	47
Dec	(1, 0)	1	32	1	46

Since there is no consensus on a universal error performance criterion, Table 45.3 includes different error measures: root mean square error ($RMSE$), mean absolute error (MAE), mean relative error ($MRE(\%)$), maximum relative error ($RE_{max}(\%)$), correlation coefficient (ρ), and variance (σ^2) criteria, respectively. They are defined as follows:

$$RMSE = \left(\frac{1}{P} \sum_{k=1}^P (v_k - \hat{v}_k)^2 \right)^{\frac{1}{2}}, \quad (25)$$

$$MAE = \frac{1}{P} \sum_{k=1}^P |v_k - \hat{v}_k|, \quad (26)$$

$$MRE = \frac{100}{P} \sum_{k=1}^P \frac{|v_k - \hat{v}_k|}{v_k}, \quad (27)$$

$$RE_{max} = \max \left(100 \left| \frac{v_k - \hat{v}_k}{v_k} \right| \right), \quad (28)$$

$$\rho = \frac{\sum_{k=1}^P (v_k - \bar{d})(\hat{v}_k - \bar{y})}{\left(\sum_{k=1}^P (v_k - \bar{d})^2 (\hat{v}_k - \bar{y})^2 \right)^{\frac{1}{2}}}, \quad (29)$$

$$\sigma^2 = \frac{1}{P-1} \sum_{k=1}^P (\hat{v}_k - \bar{v})^2, \quad (30)$$

Table 45.3 Global prediction errors

Methods	RMSE (m^3/s)	MAE (m^3/s)	MRE (%)	RE_{max} (%)	ρ	σ^2 ($\times 10^6$)
GFM	1471.20	745.52	22.79	203.54	0.72	2.18
PARMA	1079.30	593.83	20.09	144.27	0.84	1.17
MLP	1462.80	820.24	31.31	149.70	0.69	2.16
FNN	1330.40	606.37	17.80	79.31	0.76	1.72
GRM-MRP	1191.60	622.60	22.81	113.21	0.80	1.44
GRM-PRP	1005.00	537.10	18.93	140.75	0.86	1.02

where P means the number of predictions, \hat{v}_k is the forecast, v_k are the actual values, and \bar{v} is the respective mean. Correlation coefficient measures how well the predicted streamflow correlate with the observed ones. Correlation coefficient values closer to unity means better forecasting.

Global prediction errors shown in Table 45.3 suggests that GRM-MRP performance is comparable, but slightly lower than PARMA and RNN models from the point of view of MAE , MRE , and RE_{max} .

GRM-PRP model performs globally better than GFM, PARMA, MLP, and GRM-MRP. GRM-PRP gives lower global errors ($RMSE$ and MAE) and considerably higher correlation coefficient than FNN does. MRE , however, indicates that FNN performs slightly better than GRM-PRP.

Overall, we conclude that GRM-PRP and GRM-MRP predictive fuzzy clustering models are very effective because they provide forecasts comparable to the ones obtained by MLP and FNN models. However, the predictive clustering model is much simpler than MLP and FNN. In particular, since there is an isomorphism between FNN and fuzzy rule-based models, FNN complexity grows exponentially as the granulation of the input space increases. This is not the case with GRM-PRP because its complexity is $O(r)$.

45.5 Conclusion

In this chapter, we have introduced granular models for time-series forecasting. The models use FCM clustering techniques to group historical data patterns. Prediction patterns are classified using fuzzy clustering and forecasts are computed as a combination of the cluster centers weighted by the membership degrees of prediction patterns in each cluster. Three granular forecasting procedures have been introduced. The first procedure is based on weighted combination of local models, one for each cluster. The remaining two procedures use data median and similarity between time-series patterns.

To evaluate the performance of the models, we first addressed a chaotic time series of one dimension generated by a logistic mapping and compared the results of GFM against HATSP, a similar predictive fuzzy clustering-based model proposed in literature. The $NMSE$ criterion shows that GFM outperforms HATSP.

The granular forecasting models were also used to forecast average natural streamflows and the results compared with PARMA, MLP, and FNN. Overall, the GRM-MRP and GRM-PRP methods perform globally better than GFM, PARMA, MLP, and FNN as well. The granular forecasting model is computationally faster and simpler to use than its counterparts.

Acknowledgment

The authors thank the Research Foundation of the State of São Paulo (FAPESP) for grant 03/10019–9, and the Brazilian National Research Council (CNPq) for grants 133038/2003–3 and 304299/2003–0.

References

- [1] G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. *Time Series Analysis, Forecasting and Control*, 3rd ed. Holden Day, Oakland, CA, 1994.
- [2] A.S. Weigend and N.A. Gershenfeld. *Time Series Prediction: Forecasting de Future and Understanding the Past*. Addison-Wesley, Santa FE, NM, 1994.
- [3] H.R. Maier and G.C. Dandy. Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications. *Environ. Modelling Softw.* 15 (2000) 101–124.
- [4] R. Ballini, M. Figueiredo, S. Soares, M. Andrade, and F. Gomide. A seasonal streamflow forecasting model using neurofuzzy network. In: B. Bouchon-Meunier, R.R. Yager and L. Zadeh (eds), *Information, Uncertainty and Fusion*, 1st ed. Kluwer Academic Publishers, Norwell, MA, 2000, pp. 257–276.
- [5] L. See and S. Openshaw. A hybrid multi-model approach to river level forecasting. *Hydrol. Sci. J.* 45 (2000) 523–536.
- [6] F.J. Chang and Y.C. Chen. A counterpropagation fuzzy neural network modeling approach to real time streamflow prediction. *J. Hydrol.* 245 (2001) 153–164.

- [7] M. Figueiredo, R. Ballini, S. Soares, M. Andrade, and F. Gomide. Learning algorithms for a class of neurofuzzy network and application. *IEEE Trans. Syst. Man Cybern.* 34(3) (2004) 293–301.
- [8] M.H. Magalhães, R. Ballini, R. Gonçalves, and F. Gomide. Predictive fuzzy clustering model for natural stream-flow forecasting. In: *Proceedings of the IEEE International Conference on Fuzzy Systems*, Budapest, Hungary, July 2004, pp. 390–394.
- [9] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* 15 (1985) 116–132.
- [10] M. Setnes, R. Babuska, and H. Verbruggen. Rule-based modeling: precision and transparency. *IEEE Trans. Syst. Man Cybern.* 28(1) (1998) 165–169.
- [11] J. Yen, L. Wang, and C. Gillespie. Improving the interpretability of tsk fuzzy models by combining global learning and local learning. *IEEE Trans. Fuzzy Syst.* 6(4) (1998) 530–537.
- [12] K.J. Oh and I. Han. An intelligent clustering forecasting system based on change-point detection and artificial neural networks: Application to financial economics. In: *Proceedings of the 34th Hawaii International Conference on System Science*, Maui, Hawaii, January 2001.
- [13] J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, 1981.
- [14] W. Pedrycz and A.V. Vasilakos. Linguistic models and linguistic modeling. *IEEE Trans. Syst. Man Cybern.* 29(6) (1999) 745–757.
- [15] W. Pedrycz and A. Bargiela. Granular clustering: A granular signature of data. *IEEE Trans. Syst. Man Cybern.* 32(2) (2002) 212–224.
- [16] A. Bargiela and W. Pedrycz. Granulation of temporal data: A global view on time series. In: *Fuzzy Information Processing Society, 22nd International Conference of the North American – NAFIPS*, Chicago, IL, July 2003, pp. 191–196.
- [17] A. Bargiela and W. Pedrycz. Recursive information granulation: aggregation and interpretation issues. *IEEE Trans. Syst. Man Cybern.* 33(1) (2003) 96–112.
- [18] M. Sayal and M.-C. Shan. Analysis of numeric data streams at different granularities. *IEEE Int. Conf. Granular Comput.* 1 (2005) 237–242.
- [19] A. Bargiela. Granular modeling through regression analysis. In: *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems – IPMU*, Vol. 1, Paris, France, July 2006, pp. 1474–1480.
- [20] A. Bargiela and W. Pedrycz. *Granular Computing: An Introduction*. Kluwer Academic Publisher, Norwell, MA, 2003.
- [21] A.B. Geva. Non-stationary time series prediction using fuzzy clustering. In: *Proceedings of the 18th International Conference of the North American Fuzzy Information Processing Society*, New York, June 1999, pp. 413–417.
- [22] A.B. Geva. Hierarchical-fuzzy clustering of temporal-patterns and its applications for time-series prediction. *Pattern Recognit. Lett.* 20(14) (1999) 1599–1532.
- [23] M. Figueiredo and F. Gomide. Adaptive neuro fuzzy modelling. In: *Proceedings of FUZZ–IEEE*, Barcelona, Spain, July 1997, pp. 1567–1572.
- [24] M. Figueiredo and F. Gomide. Design of fuzzy systems using neurofuzzy networks. *IEEE Trans. Neural Netw.* 10(4) (1999) 815–827.
- [25] W. Pedrycz. *Knowledge-Based Clustering From Data to Information Granules*. John Wiley & Sons, Roboken, NJ, 2005.
- [26] A.B. Geva. Feature extraction and state recognition in biomedical signals with hierarchical unsupervised fuzzy clustering methods. *Med. Biol. Eng. Comput.* 36(5) (1998) 608–614.
- [27] A.V. Vecchia. Maximum likelihood estimation for periodic autoregressive moving average models. *Technometrics* 27(4) (1985) 375–384.

46

Rough Clustering

Pawan Lingras, S. Asharaf, and Cory Butz

46.1 Introduction

The conventional clustering techniques mandate that an object must belong to precisely one cluster. Such a requirement is found to be too restrictive in many data mining applications [1–3]. In practice, an object may display characteristics of different clusters. In such cases, an object should belong to more than one cluster, and as a result, cluster boundaries necessarily overlap. Fuzzy set representation of clusters, using algorithms such as fuzzy c -means, make it possible for an object to belong to multiple clusters with a degree of membership between 0 and 1 [4]. In some cases, the fuzzy degree of membership may be too descriptive for interpreting clustering results. Rough-set-based clustering provides a solution that is less restrictive than conventional clustering and less descriptive than fuzzy clustering.

Rough set theory has made substantial progress as a classification tool in data mining [5–11]. The basic concept of representing a set as lower and upper bounds can be used in a broader context such as clustering. Clustering in relation to rough set theory is attracting increasing interest among researchers [12–21]. Lingras [22, 23] described how a rough-set-theoretic classification scheme can be represented using a rough set genome. In subsequent publications [24–26], modifications of k -means and *Kohonen self-organizing maps* (SOMs) were proposed to create intervals of clusters based on rough set theory.

The rough clustering methods described above are based on Euclidean distances in the original input data space. *Support vector clustering* (SVC) [27] is a kernel-based clustering method that is capable of identifying clusters having arbitrary shapes. Here, the clustering problem is formulated as a *quadratic programming* (QP) [28] problem to learn a minimum radius sphere enclosing the image of the data set to be clustered in a high-dimensional feature space. In SVC, this problem is solved by employing a method, called kernel trick [29], that helps solve the QP problem without explicit mapping of data points from the input data space to the higher dimensional feature space. Once the QP problem is solved, SVC uses a graph-based cluster labeling method to identify the arbitrary-shaped clusters existing in the input data space. *Rough support vector clustering* (RSVC) [30] is a soft clustering method derived from the SVC paradigm. It achieves soft data clustering by a natural fusion of rough set theory and SVC. In RSVC, the QP problem involved in SVC is modified to impart a rough-set-theoretic flavor. The modified QP problem obtained for RSVC turns out to be the same as the one involved in SVC. Therefore, the existing solution strategies used for solving the SVC QP problem can be used for solving the RSVC QP problem as well. The cluster labeling method of RSVC is a modified version of the one used in SVC.

This chapter describes the evolutionary, neural, and statistical approaches for creating rough clusters based on the Euclidean distance measure. These approaches are compared with RSVC based on non-linear transformation of input vectors. The theoretical and axiomatic comparison is supplemented with

experiments on real-world and artificial data sets. A brief discussion on extensions and hybridization of rough k -means as well as other rough-set-based approaches is also provided. Finally, the feasibility of these approaches for large data sets is discussed.

The remainder of this chapter is organized as follows. Section 46.2 reviews the adaptation of rough sets for clustering. Evaluating rough set genomes are discussed in Section 46.3. Section 46.4 shows how k -means can be applied in rough sets. Modifications of the Kohonen algorithm are discussed in Section 46.5. Section 46.6 examines RSVC method and gives a comparison between rough k -means and RSVC methods. Section 46.7 describes other rough-set-based clustering approaches. Feasibility of the proposed approaches for large data sets is discussed in Section 46.8. Section 46.9 contains our conclusions.

46.2 Adaptation of Rough Set Theory for Clustering

Due to space limitations, some familiarity with rough set theory is assumed [10]. Rough sets were originally proposed using equivalence relations. However, it is possible to define a pair of upper and lower bounds ($\underline{A}(X)$, $\overline{A}(X)$) or a rough set for every set $X \subseteq U$ as long as the properties specified by Pawlak [8, 10] are satisfied. Yao and Lin [31, 32] described various generalizations of rough sets by relaxing the assumptions of an underlying equivalence relation. Such a trend toward generalization is also evident in rough mereology proposed by Polkowski and Skowron [33] and the use of information granules in a distributed environment by Skowron and Stepaniuk [34]. The present study uses such a generalized view of rough sets. If one adopts a more restrictive view of rough set theory, the rough sets developed in this chapter may have to be looked upon as interval sets.

Let us consider a hypothetical classification scheme

$$U/P = \{X_1, X_2, \dots, X_k\} \quad (1)$$

that partitions the set U based on an equivalence relation P . Let us assume due to insufficient knowledge that it is not possible to precisely describe the sets X_i , $1 \leq i \leq k$, in the partition. Based on the available information, however, it is possible to define each set $X_i \in U/P$ using its lower $\underline{A}(X_i)$ and upper $\overline{A}(X_i)$ bounds. We will use vector representations \mathbf{u} and \mathbf{v} for objects and \mathbf{x}_i for cluster X_i .

We are considering the upper and lower bounds of only a few subsets of U . Therefore, it is not possible to verify all the properties of the rough sets [8, 10]. However, the family of upper and lower bounds of $\mathbf{x}_i \in \mathbf{U}/\mathbf{P}$ are required to follow some of the basic rough set properties such as

- (C1) An object \mathbf{v} can be part of at most one lower bound.
- (C2) $\mathbf{v} \in \underline{A}(\mathbf{x}_i) \implies \mathbf{v} \in \overline{A}(\mathbf{x}_i)$.
- (C3) An object \mathbf{v} is not part of any lower bound $\iff \mathbf{v}$ belongs to two or more upper bounds.

Property (C1) emphasizes the fact that a lower bound is included in a set. If two sets are mutually exclusive, their lower bounds should not overlap. Property (C2) confirms the fact that the lower bound is contained in the upper bound. Property (C3) is applicable to the objects in the boundary regions, which are defined as the differences between upper and lower bounds. The exact membership of objects in the boundary region is ambiguous. Therefore, property (C3) states that an object cannot belong to only a single boundary region. Note that (C1)–(C3) are not necessarily independent or complete. However, enumerating them will be helpful later in understanding the rough set adaptation of evolutionary, neural, and statistical clustering methods.

46.3 Rough Set Genome and Its Evaluation

Some familiarity with genetic algorithms [35, 36] is assumed here. A rough set genome consists of n genes, one gene per object in U [22, 23]. A gene for an object is a string of bits that describes which lower and upper approximations the object belongs to. Properties (C1)–(C3) provide certain restrictions

on the memberships. An object $\mathbf{u} \in U$ can belong to the lower approximation of at most one class \mathbf{x}_i . If an object belongs to the lower approximation of \mathbf{x}_i , then it also belongs to the upper approximation of \mathbf{x}_i . If an object does not belong to the lower approximation of any \mathbf{x}_i , then it belongs to the upper approximation of at least two \mathbf{x}_i . Based on these observations, the string for a gene can be partitioned into two parts, *lower* and *upper*. Both the lower and upper parts of the string consist of k bits each. The i th bit in lower/upper string tells whether the object is in the lower/upper approximation of \mathbf{x}_i . If $\mathbf{u} \in \underline{A}(\mathbf{x}_i)$, then based on the property (C2), $\mathbf{u} \in \overline{A}(\mathbf{x}_i)$. Therefore, the i th bit in both the lower and upper strings will be turned on. Based on the property (C1), all the other bits must be turned off. If \mathbf{u} is not in any of the lower approximations, then according to property (C3), it must be in two or more upper approximations of \mathbf{x}_i , $1 \leq i \leq k$, and corresponding i th bits in the upper string will be turned on.

Figure 46.1 shows examples of all valid and some invalid genes for $k = 3$. Genes $gene_1$ to $gene_7$ are all the acceptable values of genes for $k = 3$. An object represented by $gene_1$ belongs to $\underline{A}(\mathbf{x}_1)$ and $\overline{A}(\mathbf{x}_2)$. An object represented by $gene_6$ belongs to $\underline{A}(\mathbf{x}_2)$, and by property (C2) to $\overline{A}(\mathbf{x}_2)$. Any other value not given by $gene_1$ to $gene_7$ is not valid. Figure 46.1 also shows four of the 57 invalid values. The $invalidGene_1$ is invalid because an object cannot be in $\underline{A}(\mathbf{x}_1)$ and not be in $\overline{A}(\mathbf{x}_1)$. The $invalidGene_2$ is invalid because an object cannot be in $\underline{A}(\mathbf{x}_2)$ and in $\overline{A}(\mathbf{x}_3)$ at the same time. The $invalidGene_3$ is invalid because an object cannot be in $\underline{A}(\mathbf{x}_1)$ and in $\underline{A}(\mathbf{x}_3)$ at the same time. Since the object represented by $invalidGene_4$ belongs only to $\overline{A}(\mathbf{x}_1)$, according to property (C3) it is invalid.

A genetic algorithm package such as the one used in the study [37] makes it possible to describe a set of valid gene values or alleles. All the standard genetic operations will then only create genomes that have these values. Therefore, the conventional genetic operations can be used with rough set genomes in such a package.

Examples of valid genes

	Lower			Upper		
	$\underline{A}(\mathbf{x}_3)$	$\underline{A}(\mathbf{x}_2)$	$\underline{A}(\mathbf{x}_1)$	$\overline{A}(\mathbf{x}_3)$	$\overline{A}(\mathbf{x}_2)$	$\overline{A}(\mathbf{x}_1)$
$gene_1$	0	0	0	0	1	1
$gene_2$	0	0	0	1	0	1
$gene_3$	0	0	0	1	1	0
$gene_4$	0	0	0	1	1	1
$gene_5$	0	0	1	0	0	1
$gene_6$	0	1	0	0	1	0
$gene_7$	1	0	0	1	0	0

Examples of invalid genes

	Lower			Upper		
	$\underline{A}(\mathbf{x}_3)$	$\underline{A}(\mathbf{x}_2)$	$\underline{A}(\mathbf{x}_1)$	$\overline{A}(\mathbf{x}_3)$	$\overline{A}(\mathbf{x}_2)$	$\overline{A}(\mathbf{x}_1)$
$invalidGene_1$	0	0	1	0	0	0
$invalidGene_2$	0	1	0	1	1	0
$invalidGene_3$	1	0	1	0	0	0
$invalidGene_4$	0	0	0	0	0	1

Figure 46.1 Example of valid (top) and invalid (bottom) genes in a rough set genome

The quality of a conventional classification scheme is determined by using the within group error [38], denoted by Δ , and given by

$$\Delta = \sum_{i=1}^k \sum_{\mathbf{u}, \mathbf{v} \in \mathbf{x}_i} d(\mathbf{u}, \mathbf{v}), \quad (2)$$

where \mathbf{u} and \mathbf{v} are objects from the same class \mathbf{x}_i .

The function d provides the distance between two objects. The distance $d(\mathbf{u}, \mathbf{v})$ is given by

$$d(\mathbf{u}, \mathbf{v}) = \sqrt{\frac{\sum_{j=1}^m (u_j - v_j)^2}{m}}. \quad (3)$$

For a rough set classification scheme, the exact values of classes $\mathbf{x}_i \in U/P$ are not known. Given two objects $\mathbf{u}, \mathbf{v} \in U$, we have three distinct possibilities:

1. Both \mathbf{u} and \mathbf{v} are in the same lower approximation $\underline{A}(\mathbf{x}_i)$.
2. Object \mathbf{u} is in a lower approximation $\underline{A}(\mathbf{x}_i)$ and \mathbf{v} is in the corresponding upper approximation $\overline{A}(\mathbf{x}_i)$, and case 1 is not applicable.
3. Both \mathbf{u} and \mathbf{v} are in the same upper approximation $\overline{A}(\mathbf{x}_i)$, and cases 1 and 2 are not applicable.

For these possibilities, one can define three corresponding types of within group errors, Δ_1 , Δ_2 , and Δ_3 , as

$$\begin{aligned} \Delta_1 &= \sum_{i=1}^k \sum_{\mathbf{u}, \mathbf{v} \in \underline{A}(\mathbf{x}_i)} d(\mathbf{u}, \mathbf{v}), \\ \Delta_2 &= \sum_{i=1}^k \sum_{\mathbf{u} \in \underline{A}(\mathbf{x}_i), \mathbf{v} \in \overline{A}(\mathbf{x}_i), \mathbf{v} \notin \underline{A}(\mathbf{x}_i)} d(\mathbf{u}, \mathbf{v}), \\ \Delta_3 &= \sum_{i=1}^k \sum_{\mathbf{u}, \mathbf{v} \in \overline{A}(\mathbf{x}_i), \mathbf{u}, \mathbf{v} \notin \underline{A}(\mathbf{x}_i)} d(\mathbf{u}, \mathbf{v}). \end{aligned}$$

The total error of rough set classification will then be a weighted sum of these errors:

$$\Delta_{total} = w_1 \times \Delta_1 + w_2 \times \Delta_2 + w_3 \times \Delta_3. \quad (4)$$

Since Δ_1 corresponds to situations where both objects definitely belong to the same class, the weight w_1 should have the highest value. On the other hand, Δ_3 corresponds to a situation where both objects may or may not belong to the same class. Hence, w_3 should have the lowest value. That is, $w_1 > w_2 > w_3$. There are many possible ways of developing an error measure for rough set classifications. The measure Δ_{total} is perhaps one of the simplest. More sophisticated alternatives may be used, depending on the application.

If we used genetic algorithms to minimize Δ_{total} , the genetic algorithms would try to classify all the objects in upper approximations by taking advantage of the fact that $w_3 < w_1$. This may not necessarily be the best classification scheme. We want the rough set classification to be as precise as possible. Therefore, a precision measure needs to be used in conjunction with Δ_{total} for evaluating the quality of a rough set genome. A possible precision measure can be defined [8] as

$$precision = \frac{\text{Number of objects classified in lower approximations}}{\text{Total number of objects}}. \quad (5)$$

The objective of the genetic algorithms will then be to maximize the quantity

$$objective = p \times precision + \frac{e}{\Delta_{total}}, \quad (6)$$

where p and e are additional parameters. The parameter p describes the importance of the precision measure in determining the quality of a rough set genome. Higher values of p will result in smaller boundary region. Similarly, e indicates the importance of within group errors relative to the size of the boundary region. It should perhaps be reiterated that the genetic operations implemented by the GALib [37] makes it possible to describe a set of valid gene values or alleles. All the standard genetic operations will then only create genomes that have these values. Therefore, there is no need to modify the conventional genetic operations nor provide penalty for invalid gene values in the objective function.

46.4 Adaptation of K-Means to Rough Set Theory

Here, we refer readers to [39, 40] for discussion on conventional k -means algorithm. Incorporating rough sets into k -means clustering requires the addition of the concept of lower and upper bounds. Calculation of the centroids of clusters from conventional k -means needs to be modified to include the effects of these bounds. The modified centroid calculations for rough sets are then given by

$$\begin{aligned}
 & \text{if } \underline{A}(\mathbf{x}) \neq \emptyset \text{ and } \overline{A}(\mathbf{x}) - \underline{A}(\mathbf{x}) = \emptyset \\
 & \quad x_j = \frac{\sum_{v \in \underline{A}(\mathbf{x})} v_j}{|\underline{A}(\mathbf{x})|} \\
 & \text{else if } \underline{A}(\mathbf{x}) = \emptyset \text{ and } \overline{A}(\mathbf{x}) - \underline{A}(\mathbf{x}) \neq \emptyset \\
 & \quad x_j = \frac{\sum_{v \in (\overline{A}(\mathbf{x}) - \underline{A}(\mathbf{x}))} v_j}{|\overline{A}(\mathbf{x}) - \underline{A}(\mathbf{x})|} \\
 & \text{else} \\
 & \quad x_j = w_{lower} \times \frac{\sum_{v \in \underline{A}(\mathbf{x})} v_j}{|\underline{A}(\mathbf{x})|} + w_{upper} \times \frac{\sum_{v \in (\overline{A}(\mathbf{x}) - \underline{A}(\mathbf{x}))} v_j}{|\overline{A}(\mathbf{x}) - \underline{A}(\mathbf{x})|}, \tag{7}
 \end{aligned}$$

where $1 \leq j \leq m$. The parameters w_{lower} and w_{upper} correspond to the relative importance of lower and upper bounds, and $w_{lower} + w_{upper} = 1$. If the upper bound of each cluster were equal to its lower bound, the clusters would be conventional clusters. Therefore, the boundary region $\overline{A}(\mathbf{x}) - \underline{A}(\mathbf{x})$ will be empty, and the second term in the equation will be ignored. Thus, equation (7) will reduce to conventional centroid calculations.

The next step in the modification of the k -means algorithms for rough sets is to design criteria to determine whether an object belongs to the upper or lower bound of a cluster given as follows. For each object vector \mathbf{v} , let $d(\mathbf{v}, \mathbf{x}_j)$ be the distance between itself and the centroid of cluster \mathbf{x}_j . Let $d(\mathbf{v}, \mathbf{x}_i) = \min_{1 \leq j \leq k} d(\mathbf{v}, \mathbf{x}_j)$. The ratio $d(\mathbf{v}, \mathbf{x}_i)/d(\mathbf{v}, \mathbf{x}_j)$, $1 \leq i, j \leq k$, are used to determine the membership of \mathbf{v} . Let $T = \{j : d(\mathbf{v}, \mathbf{x}_i)/d(\mathbf{v}, \mathbf{x}_j) \leq \text{threshold} \text{ and } i \neq j\}$.

1. If $T \neq \emptyset$, $\mathbf{v} \in \overline{A}(\mathbf{x}_i)$ and $\mathbf{v} \in \overline{A}(\mathbf{x}_j)$, $\forall j \in T$. Furthermore, \mathbf{v} is not part of any lower bound. The above criterion guarantees that property (C3) is satisfied.
2. Otherwise, if $T = \emptyset$, $\mathbf{v} \in \underline{A}(\mathbf{x}_i)$. In addition, by property (C2), $\mathbf{v} \in \overline{A}(\mathbf{x}_i)$.

It should be emphasized that the approximation space A is not defined based on any predefined relation on the set of objects. The upper and lower bounds are constructed based on the criteria described above.

46.5 Kohonen Algorithm Modifications

In this section, we describe a rough set extension of the well-known *Kohonen self-organizing map* [41]. The rough-set-based Kohonen algorithm uses the concept of lower and upper bounds in the equations for updating the weights of winners. The Kohonen rough set architecture is similar to the conventional

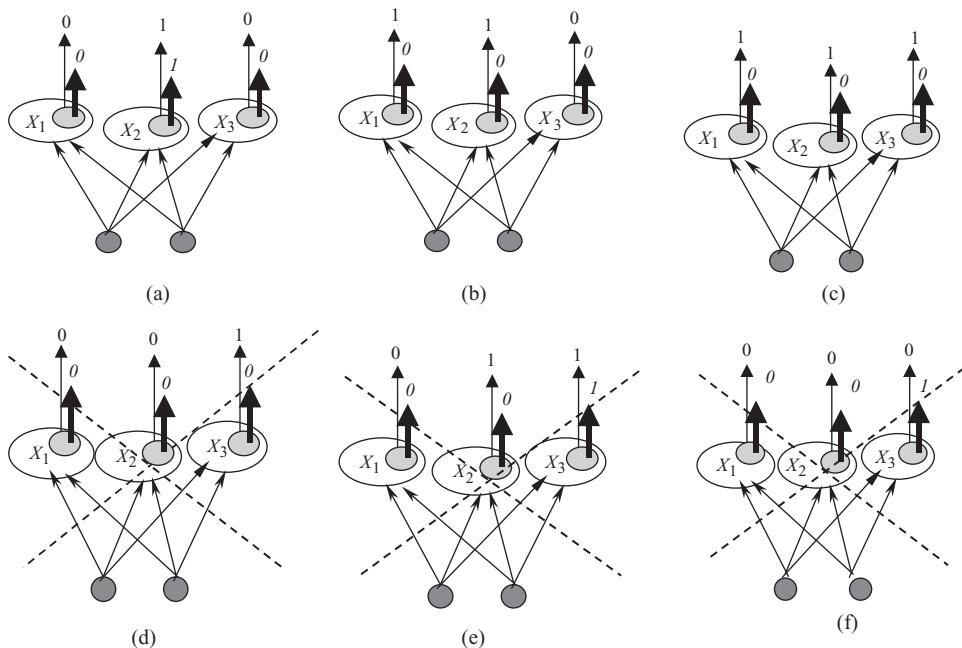


Figure 46.2 Modified Kohonen neural network

Kohonen architecture. It consists of two layers, an input layer and the Kohonen rough set layer (rough set output layer). These two layers are fully connected. Each input layer neuron has a feed-forward connection to each output layer neuron. Figure 46.2 illustrates the Kohonen rough set neural network architecture for a one-dimensional case. A neuron in the Kohonen layer consists of two parts, a lower neuron and an upper neuron. The lower neuron has an output of 1 if an object belongs to the lower bound of the cluster. Similarly, a membership in the upper bound of the cluster will result in an output of 1 from the upper neuron. Since an object belonging to the lower bound of a cluster also belongs to its upper bound, when the lower neuron has an output of 1, the upper neuron also has an output of 1. However, a membership in the upper bound of a cluster does not necessarily imply the membership in its lower bound. Therefore, the upper neuron contains the lower neuron. Figure 46.2 provides some cases to explain outputs from the Kohonen rough set neural networks based on properties (C1)–(C3). Figure 46.2a–c shows some of the possible outputs, while Figure 46.2d–f shows some of the invalid outputs from the network. Figure 46.2a shows a case where an object belongs to lower bound of cluster x_2 . Based on the property (C2), it also belongs to the upper bound of x_2 . Figure 46.2b shows a situation where an object belongs to the upper bounds of clusters x_1 and x_2 . The object in Figure 46.2c belongs to the upper bounds of clusters x_1 , x_2 , and x_3 . Figure 46.2d shows an invalid situation where an object belongs only to the upper bound of the cluster x_3 . This is a violation of property (C3). Figure 46.2e shows a violation of property (C1), where an object belongs to lower bound of x_3 as well as the upper bound of x_2 . Similarly, a violation of property (C2) can be seen in the invalid case of Figure 46.2f. Here, the object belongs only to the lower bound of cluster x_3 and not its upper bound. The modification of the Kohonen algorithm must ensure that the properties (C1)–(C3) are obeyed by avoiding cases such as the ones shown in Figure 46.2d–f.

The interval clustering provides good results if initial weights are obtained by running the conventional Kohonen learning. The next step in the modification of the Kohonen algorithm for obtaining rough sets is to design criteria to determine whether an object belongs to the upper or lower bounds of a cluster. The assignment criteria for the modified Kohonen algorithm is the same as the modified k -means algorithm discussed in the previous section. For each object vector v , let $d(v, x_j)$ be the distance between itself

and the weight vector \mathbf{x}_j of cluster X_j . Let $d(\mathbf{v}, \mathbf{x}_i) = \min_{1 \leq j \leq k} d(\mathbf{v}, \mathbf{x}_j)$. The ratios $\frac{d(\mathbf{v}, \mathbf{x}_i)}{d(\mathbf{v}, \mathbf{x}_j)}$ are used to determine the membership of \mathbf{v} as follows. Let $T = \{j : d(\mathbf{v}, \mathbf{x}_i)/d(\mathbf{v}, \mathbf{x}_j) \leq \text{threshold} \text{ and } i \neq j\}$.

1. If $T \neq \emptyset$, $\mathbf{v} \in \overline{A}(\mathbf{x}_i)$ and $\mathbf{v} \in \overline{A}(\mathbf{x}_j)$, $\forall j \in T$. Furthermore, \mathbf{v} is not part of any lower bound. The above criterion guarantees that property (C3) is satisfied. The weight vectors \mathbf{x}_i and \mathbf{x}_j are modified as

$$\begin{aligned}\mathbf{x}_i^{new} &= \mathbf{x}_i^{old} + \alpha_{upper}(t) \times (\mathbf{v} - \mathbf{x}_i^{old}), \text{ and} \\ \mathbf{x}_j^{new} &= \mathbf{x}_j^{old} + \alpha_{upper}(t) \times (\mathbf{v} - \mathbf{x}_j^{old}).\end{aligned}$$

2. Otherwise, if $T = \emptyset$, $\mathbf{v} \in \underline{A}(\mathbf{x}_i)$. In addition, by property (C2), $\mathbf{v} \in \overline{A}(\mathbf{x}_i)$. The weight vectors \mathbf{x}_i is modified as

$$\mathbf{x}_i^{new} = \mathbf{x}_i^{old} + \alpha_{lower}(t) \times (\mathbf{v} - \mathbf{x}_i^{old}).$$

Usually, $\alpha_{lower} > \alpha_{upper}$. It can easily be verified that the above algorithm preserves properties (C1)–(C3).

46.6 Rough Support Vector Clustering

For a proper understanding of RSVC, knowledge of SVC is essential. Since SVC is a relatively new technique, we give it a brief introduction and then discuss RSVC.

46.6.1 Support Vector Clustering

SVC is a clustering method that uses ‘kernel trick’ [29]. Here, the computation in a high-dimensional feature space is achieved using a kernel function, without explicitly mapping data points to the high-dimensional feature space. In SVC, we look for the smallest sphere in a feature space that encloses the image of the data such as shown in Figure 46.3. If this sphere is mapped back to data space, it forms a set of contours that enclose the data points, such as shown in Figure 46.4. These contours are interpreted as cluster boundaries. Points enclosed by each separate contour are associated with the same cluster. The kernel parameters can control the number of clusters. Here, the outliers are handled with the help of a soft margin formulation.

To define the formulation, let $\{\mathbf{u}_i\} \subseteq U$ be an m -dimensional data set having n points, with $\mathbf{u}_i \in R^m$ being the data space. Now using a non-linear transformation ϕ from U to some high-dimensional feature space, we look for the smallest sphere of radius R enclosing all the points in U . Now the primal problem can be stated as

$$\min_{R, \xi_i} R^2 + C \sum_{i=1}^n \xi_i$$

$$\text{s.t. } \|\phi(\mathbf{u}_i) - \mu\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0 \ \forall i. \quad (8)$$

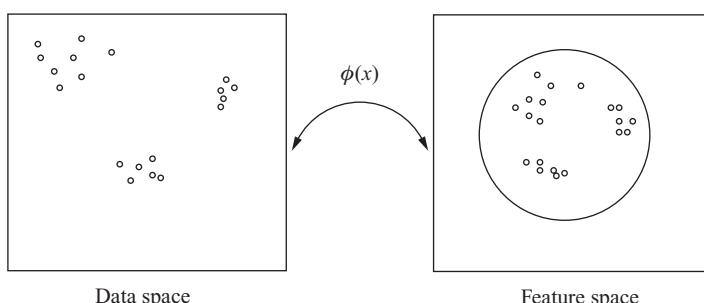


Figure 46.3 Support vector clustering: data space to feature space mapping. Here, ϕ is the implicit non-linear transformation achieved by the kernel function

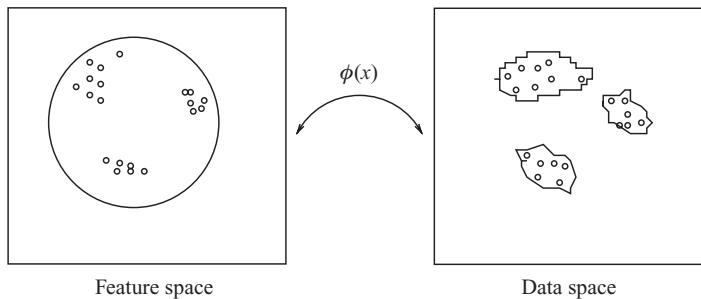


Figure 46.4 Support vector clustering: the data space contours (clusters) obtained by the reverse mapping of the feature space sphere. Here, ϕ is the implicit non-linear transformation achieved by the kernel function

Here, $C \sum_{i=1}^n \xi_i$ is the penalty term for the patterns with distance from the center of the sphere in feature space being greater than R (patterns that lie outside the feature space sphere), μ is the center of the sphere in the high-dimensional feature space, and $\|\cdot\|$ is the L_2 norm.

Since this is a convex quadratic programming problem, it is easy to solve its Wolfe dual [28] form. The dual formulation is

$$\begin{aligned} \min_{\alpha_i} \quad & \sum_{i,j=1}^n \alpha_i \alpha_j K(\mathbf{u}_i, \mathbf{u}_j) - \sum_{i=1}^n \alpha_i K(\mathbf{u}_i, \mathbf{u}_i) \\ \text{s.t. } & 0 \leq \alpha_i \leq C \quad \text{for } i = 1, \dots, n, \quad \sum_{i=1}^n \alpha_i = 1. \end{aligned} \quad (9)$$

Here, $K(\mathbf{u}_i, \mathbf{u}_j)$ represents the kernel function giving the dot product $\phi(\mathbf{u}_i) \cdot \phi(\mathbf{u}_j)$ in the high-dimensional feature space, and α_i s are the Lagrangian multipliers.

The value of α_i decides whether a point $\phi(\mathbf{u}_i)$ is inside, outside, or on the sphere. The points with $0 < \alpha_i < C$ form the support vectors. Hence the radius of the sphere enclosing the image of the data points is given by $R = G(\mathbf{u}_i)$, where $0 < \alpha_i < C$ and where

$$\begin{aligned} G^2(\mathbf{u}_i) &= \|\phi(\mathbf{u}_i) - \mu\|^2 \\ &= K(\mathbf{u}_i, \mathbf{u}_i) - 2 \sum_{j=1}^n \alpha_j K(\mathbf{u}_j, \mathbf{u}_i) + \sum_{j,k=1}^n \alpha_j \alpha_k K(\mathbf{u}_j, \mathbf{u}_k). \end{aligned} \quad (10)$$

Now the contours that enclose the points in data space are defined by $\{\mathbf{u} : G(\mathbf{u}) = R\}$. Thus, the computation in high-dimensional feature space and also reverse mapping to find the contours in data space are avoided with the help of Kernel function. Once these contours are found, the objects are assigned to different clusters. The cluster assignments employ a geometric method involving $G(\mathbf{u})$, based on the observation that given a pair of points that belong to different clusters, any path that connects them must exit from the sphere in the feature space. So we can define an adjacency matrix M by considering all pairs of points \mathbf{u}_i and \mathbf{u}_j whose images lie in or on the sphere in the feature space and then looking at the image of the path that connects them as

$$M[i, j] = \begin{cases} 1 & \text{if } G(y) \leq R \quad \forall y \in [\mathbf{u}_i, \mathbf{u}_j] \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Clusters are now defined as the connected components of the graph induced by M . The points that lie outside the sphere, known as bounded support vectors, can be assigned to the closest clusters.

46.6.2 Rough Support Vector Clustering

RSVC is an extension of the SVC paradigm that employs rough set theory to achieve soft clustering. To discuss the method formally, let us use the notion of rough sets to introduce a rough sphere. A *rough sphere* is defined as a sphere having an inner radius R defining its lower approximation and an outer radius $T > R$ defining its upper approximation. As in SVC, RSVC also uses a kernel function to achieve computation in a high-dimensional feature space. It tries to find the smallest rough sphere in the high-dimensional feature space enclosing the images of all the points in the data set. Now those points whose images lie within the lower approximation ($\underline{A}(\cdot)$) are points that definitely belong to exactly one cluster (called the hard core of a cluster) and those points whose images lie in the boundary region ($\overline{A}(\cdot) - \underline{A}(\cdot)$, i.e., in the upper approximation $\overline{A}(\cdot)$ but not in the lower approximation $\underline{A}(\cdot)$), may be shared by more than one cluster (called the soft core of the cluster). Some points are permitted to lie outside the sphere and are termed outliers. By using a non-linear transformation ϕ from data space to some high-dimensional feature space, we seek the smallest enclosing rough sphere of inner radius R and outer radius T . Now the primal problem can be stated formally as

$$\begin{aligned} \min_{R, T, \xi_i} \quad & R^2 + T^2 + \frac{1}{vn} \sum_{i=1}^n \xi_i + \frac{\delta}{vn} \sum_{i=1}^n \xi'_i \\ \text{s.t.} \quad & \|\phi(\mathbf{u}_i) - \mu\|^2 \leq R^2 + \xi_i + \xi'_i \quad 0 \leq \xi_i \leq T^2 - R^2 \quad \xi'_i \geq 0 \quad \forall i. \end{aligned} \quad (12)$$

Here, $\frac{1}{vn} \sum_{i=1}^n \xi_i$ is a penalty term for the patterns with distance from the center of the sphere in feature space being greater than R (patterns falling in the boundary region) and $\frac{\delta}{vn} \sum_{i=1}^n \xi'_i$ is a penalty term associated with the patterns whose distance from the center of the sphere in feature space is greater than T (patterns falling outside the rough sphere).

Since this is a convex quadratic programming problem, it is easy to write its Wolfe dual. The Lagrangian can be written as

$$\begin{aligned} L = R^2 + T^2 + \frac{1}{vn} \sum_{i=1}^n \xi_i + \frac{\delta}{vn} \sum_{i=1}^n \xi'_i + \sum_{i=1}^n \alpha_i (\|\phi(\mathbf{u}_i) - \mu\|^2 - R^2 - \xi_i - \xi'_i) \\ - \sum_{i=1}^n \beta_i \xi_i + \sum_{i=1}^n \lambda_i (\xi_i - T^2 + R^2) - \sum_{i=1}^n \eta_i \xi'_i, \end{aligned} \quad (13)$$

where the Lagrange multipliers α_i , β_i , λ_i , and η_i are non-negative $\forall i$. Using the Karush–Kuhn–Tucker (KKT) [28] conditions on equation (13), we obtain

$$\begin{aligned} \sum_{i=1}^n \alpha_i &= 2 & \mu &= \frac{1}{2} \sum_{i=1}^n \alpha_i \phi(\mathbf{u}_i) \\ \beta_i - \lambda_i &= \frac{1}{vn} - \alpha_i & \frac{\delta}{vn} - \alpha_i &= \eta_i \\ \alpha_i (\|\phi(\mathbf{u}_i) - \mu\|^2 - R^2 - \xi_i - \xi'_i) &= 0 \\ \lambda_i (\xi_i - T^2 + R^2) &= 0 \\ \beta_i \xi_i &= 0 & \eta_i \xi'_i &= 0. \end{aligned}$$

From the above equations the Wolfe dual form can be written as

$$\begin{aligned} \min_{\alpha_i} \quad & \sum_{i,j=1}^n \alpha_i \alpha_j K(\mathbf{u}_i, \mathbf{u}_j) - \sum_{i=1}^n \alpha_i K(\mathbf{u}_i, \mathbf{u}_i) \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq \frac{\delta}{vn} \quad \text{for } i = 1, \dots, n, \quad \sum_{i=1}^n \alpha_i = 2. \end{aligned} \quad (14)$$

Here, $K(\mathbf{u}_i, \mathbf{u}_j)$ represents the Kernel function giving the dot product $\phi(\mathbf{u}_i) \cdot \phi(\mathbf{u}_j)$ in the high-dimensional feature space [27].

If $\delta > 1$, we obtain RSVC. The formulation reduces to the original SVC for $\delta = 1$. Also the values of α_i decide whether the pattern \mathbf{u}_i falls in the lower approximation, the boundary region, or outside the feature space rough sphere. From KKT conditions on equation (13), it can be observed that image of points with

- $\alpha_i = 0$ lie in the lower approximation.
- $0 < \alpha_i < \frac{1}{vn}$ form the hard support vectors (support vectors marking the boundary of the lower approximation).
- $\alpha_i = \frac{1}{vn}$ lie in the boundary region (patterns that may be shared by more than one cluster).
- $\frac{1}{vn} < \alpha_i < \frac{\delta}{vn}$ form the soft support vectors (support vectors marking the boundary of the upper approximation).
- $\alpha_i = \frac{\delta}{vn}$ lie outside the rough sphere (bounded support vectors).

Cluster Assignment

Once the dual problem is solved to find the α_i values, the clusters can be obtained using the following strategy. Let us define

$$\begin{aligned} R &= G(\mathbf{u}_i) : 0 < \alpha_i < \frac{1}{vn} \\ T &= G(\mathbf{u}_i) : \frac{1}{vn} < \alpha_i < \frac{\delta}{vn}, \end{aligned} \quad (15)$$

where $G(\mathbf{u}_i)$ is defined in equation (10).

From the above equations we can define the contours that enclose the lower approximation of clusters in data space as $\{\mathbf{u} : G(\mathbf{u}) = R\}$ and the contours that enclose the upper approximation of clusters in data space as $\{\mathbf{u} : G(\mathbf{u}) = T\}$. Now the soft clusters in data space are found using a strategy similar to the one used in SVC. Such an algorithm can be given as follows.

algorithm find_clusters

- As in SVC, find the adjacency matrix M by considering all pairs of points \mathbf{u}_i and \mathbf{u}_j whose images in feature space either belong to the lower approximation of the rough sphere or are hard support vectors and then looking at the image of the path that connects them as

$$M[i, j] = \begin{cases} 1 & \text{if } G(y) \leq R \quad \forall y \in [\mathbf{u}_i, \mathbf{u}_j] \\ 0 & \text{otherwise} \end{cases}$$

- Find connected components for the graph represented by M . Each connected component found gives the lower approximation of a cluster \mathbf{x}_i .
- Find the boundary regions as:

$$\begin{aligned} \mathbf{u}_i &\in \underline{A}(\mathbf{x}_i) \text{ and pattern } \mathbf{u}_k \notin \underline{A}(\mathbf{x}_j) \text{ for any cluster } j, \\ &\text{if } G(y) \leq T \quad \forall y \in [\mathbf{u}_i, \mathbf{u}_k] \text{ then } \mathbf{u}_k \in (\overline{A}(\mathbf{x}_i) - \underline{A}(\mathbf{x}_i)) \end{aligned}$$

Role of δ and v

From equation (14) it can be seen that the number of bounded support vectors is $n_{bsv} < 2\frac{vn}{\delta}$. For $\delta = 1$, $n_{bsv} < 2vn = v'n$, where $v' = 2v$. This corresponds to all the patterns \mathbf{u}_i with $\|\phi(\mathbf{u}_i) - \mu\|^2 > R^2$. Since $\delta > 1$ for RSVC, we can say that $\frac{v'}{\delta}$ is the upper bound on the fraction of points permitted to lie outside T and v' is the upper bound on the fraction of points permitted to lie outside R . Hence, v and

δ together give us control over the width of boundary region and the number of bounded support vectors. Therefore, we can choose the values of v and δ based on the percentage of the data we want to put in the soft core of the clusters and what percentage we want to treat as outliers.

46.6.3 Comparison of Rough K-Means and RSVC

As mentioned earlier, the rough clustering algorithms derived from GAs, k -means, and Kohonen are Euclidean-distance-based methods and RSVC is a kernel-based method. Hence we have given a theoretical and experimental comparison between these two approaches in this section. Since the Euclidean-distance-based rough clustering approaches are similar, we have chosen the most efficient of these, i.e., *rough k-means* (RKM) for a comparison with the RSVC. The experimental comparisons are made using two data sets.

The first data set was synthetically generated and is shown in Figure 46.5. It can be seen that the data are distributed in three possible clusters. Observe that some of the objects do not seem to belong to any one particular cluster.

Figure 46.6 shows the rough clustering obtained using the RKM algorithm. The value of *threshold* is 0.13. Dots mark the objects that belong to lower bounds, while the objects in the upper bounds are represented using asterisks. The corresponding results for RSVC are given in Figure 46.7. The parameters values were set as $v = 0.25$, $\delta = 1.25$, and $q = 5.3$.

Both algorithms create lower and upper bound representations of clusters. However, the lower bounds in the two algorithms are slightly different from each other. The lower bounds in RKM seem to be farther apart from each other than in RSVC. This difference can be explained based on the properties (C1)–(C3). Note that both algorithms satisfy properties (C1) and (C2). However, property (C3) – an object that does not belong to any lower bound must belong to two or more upper bounds – is obeyed by RKM but not by RSVC. This is why most of the boundary regions from RKM tend to fall between the lower bounds, thereby pulling the lower bounds further apart. In contrast, the boundary regions from RSVC tend to

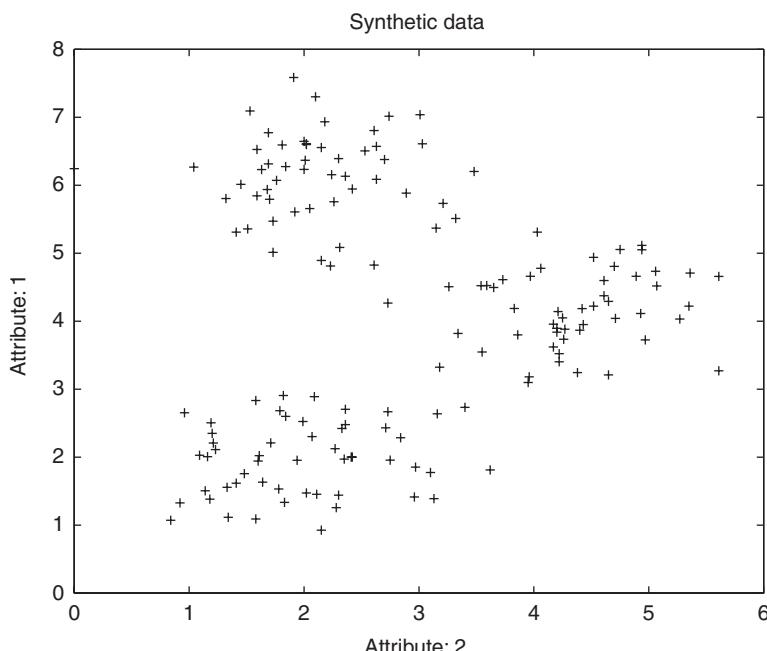


Figure 46.5 A synthetic data set

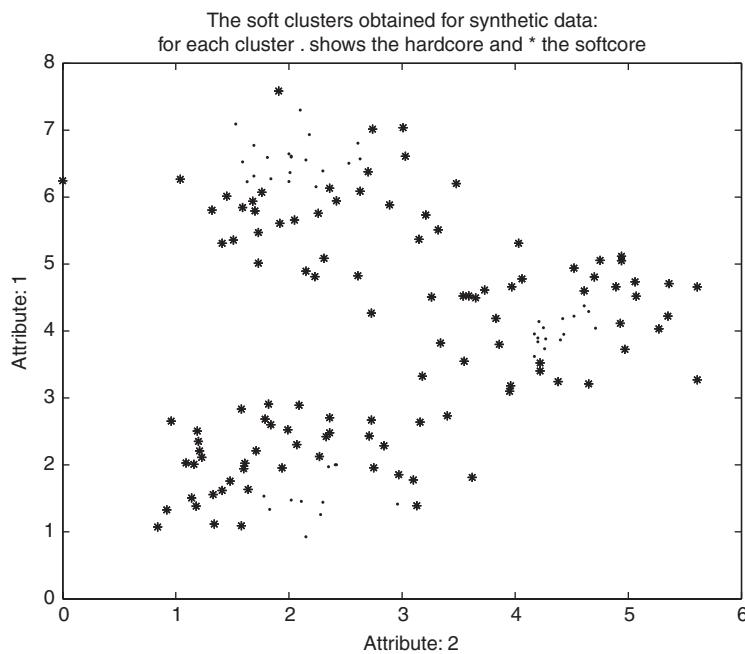


Figure 46.6 RKM clustering for the synthetic data set in Figure 46.5

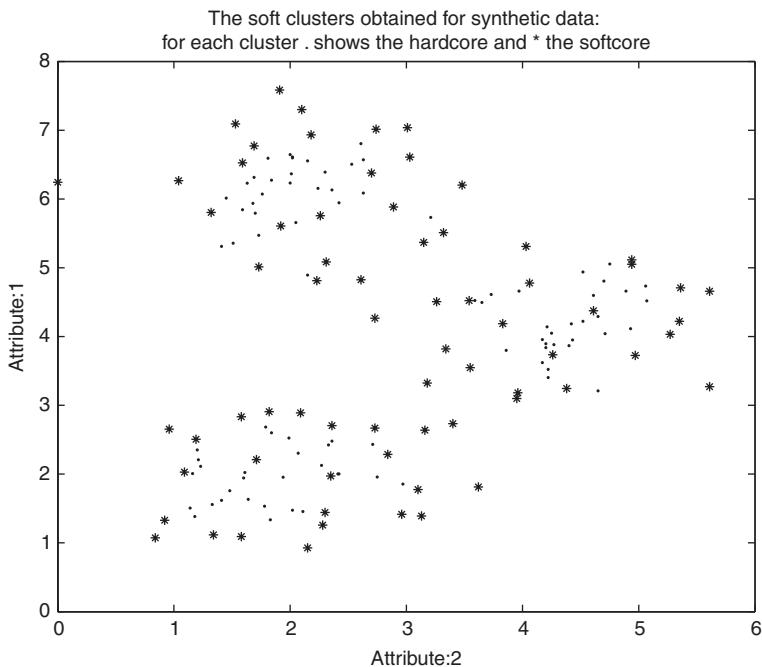


Figure 46.7 RSVC for the synthetic data set in Figure 46.5

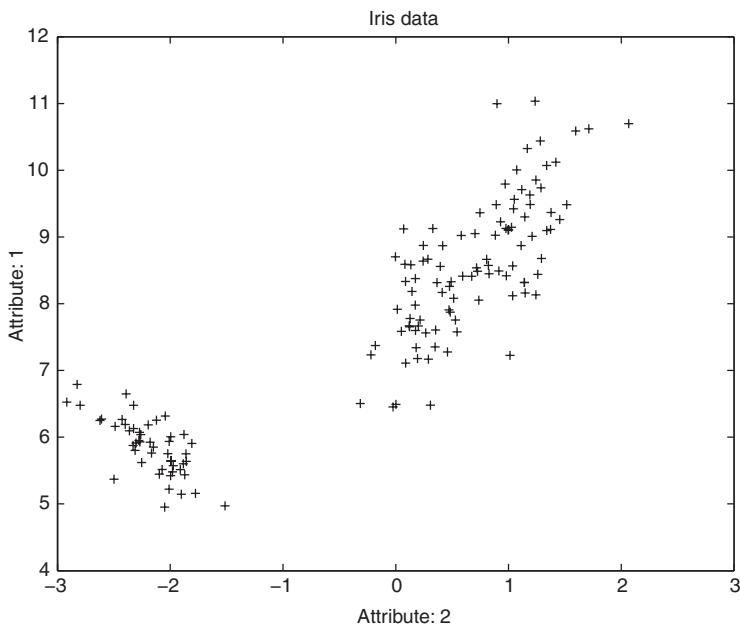


Figure 46.8 The well-known Iris data set

form a uniform-sized ring around the lower bound, allowing the lower bounds to be slightly closer to each other. One can argue the usefulness of the property (C3). Semantically, there is a possibility that an object in the upper bound of a cluster may not belong to that cluster. In such a case, the clustering algorithm should provide an alternate membership possibility by indicating any other clusters the object may belong to. On the other hand, insisting on property (C3) may result in clusters that are pulled further apart due to outliers or noisy data. Nevertheless, property (C3) is part of the conventional rough set theory. Therefore, one may have to regard the clusters created by RSVC more as interval sets than as rough sets.

The data points in synthetic data set are manipulated to highlight various features of clustering algorithms. Therefore, one should test the clustering algorithms on some real-world data. Both RKM and RSVC were tested for the Iris data set provided by one of the prominent statisticians R.A. Fisher. There are three types of Iris flowers. The data set contains 50 instances of each type, yielding a total of 150 objects. Each object is represented by four measurements, namely, sepal length, sepal width, petal length, and petal width. We chose two measurements identified using principal component analysis. The data are shown in Figure 46.8.

Figure 46.9 shows the rough clustering obtained using RKM algorithm. The value of $threshold = 0.175$. Dots mark the objects that belong to lower bounds, while the objects from upper bounds are represented using asterisks. The corresponding results for RSVC are given in Figure 46.10. The parameters values were set as $\nu = 0.25$, $\delta = 3.0$, and $q = 15.3$.

Both algorithms created three clusters, which seem to be formed along the diagonal. The bottom-left cluster is distinctly identifiable from the other two clusters. The top-right cluster is very close to the cluster in the center. Closer examination of Figure 46.9 reveals that there are a few objects – indicated by asterisks – that belong to the boundary region between these two clusters. Similar boundary region, albeit thinner, can also be found in Figure 46.10. The RKM clustering seems to have smaller lower bounds than does RSVC. However, the sizes of lower bounds can easily be changed by changing the parameters used in the clustering. We can also see that the lower bounds in RKM are a little farther apart than in RSVC, making it easier to see the boundary region between the top and center clusters. In this particular case, the fact that RKM obeys the property (C3) seems to be slightly advantageous. Such may not be the case if the data set were noisy.

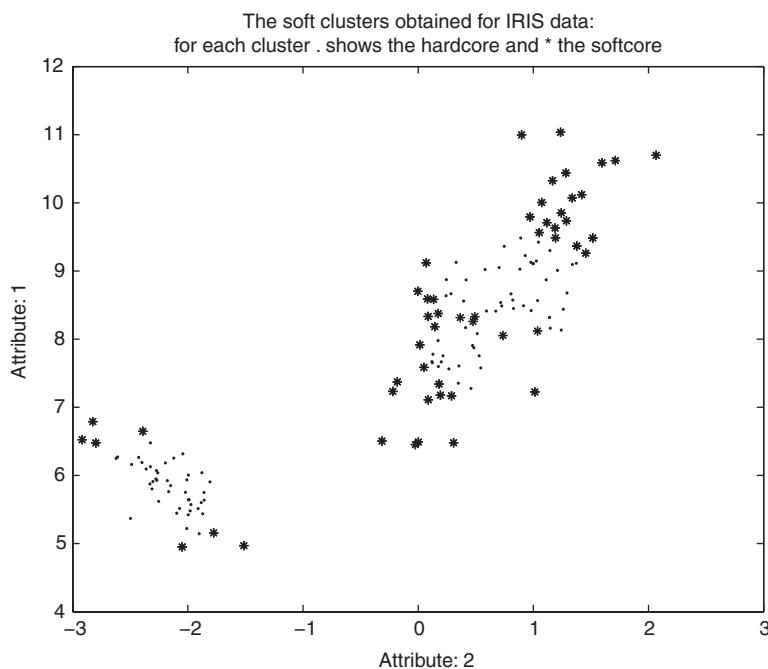


Figure 46.9 RKM clustering for the Iris data set in Figure 46.8

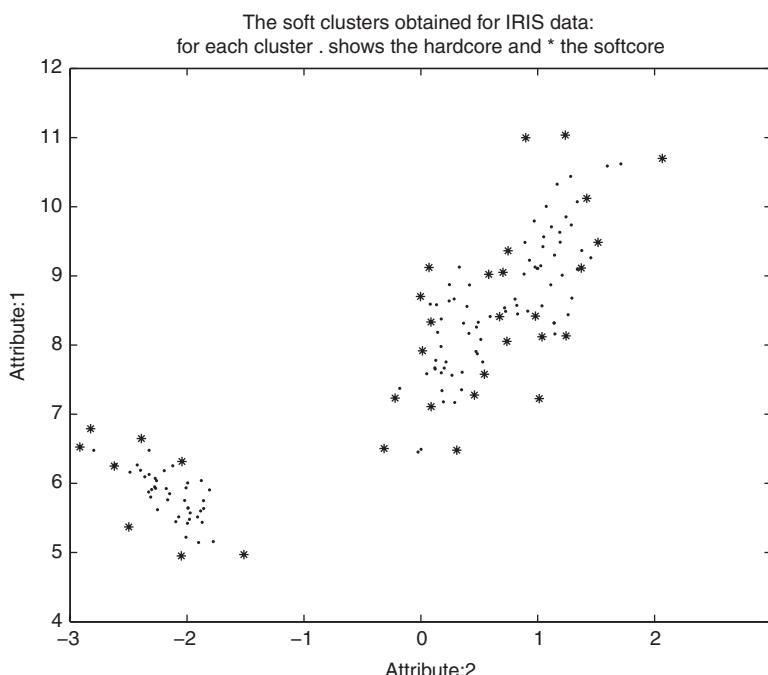


Figure 46.10 RSVC for the Iris data set in Figure 46.8

46.7 Extensions and Other Approaches

Rough clustering is gaining increasing attention from researchers. The rough k -means approach, in particular, has been a subject of further research. Peters [18, 19] discussed various deficiencies of Lingras and West's original proposal [24]. The first set of independently suggested alternatives by Peters is similar to the equation (7). Peters also suggest the use of ratios of distances as opposed to differences between distances similar to those used in the rough-set-based Kohonen algorithm described in [26]. The use of ratios is a better solution than differences. The differences vary based on the values in input vectors. The ratios, on the other hand, are not susceptible to the input values. Peters [19] have proposed additional significant modifications to rough k -means that improve the algorithm in a number of aspects. The refined rough k -means algorithm simplifies the calculations of the centroid by ensuring that lower bound of every cluster has at least one object. It also improves the quality of clusters as clusters with empty lower bound have a limited basis for its existence. Peters tested the refined rough k -means for various data sets. The experiments were used to analyze the convergence, dependency on the initial cluster assignment, and study of Davies–Boulden index and to show that the boundary region can be interpreted as a security zone as opposed to the unambiguous assignments of objects to clusters in conventional clustering. Despite the refinements, Peters concluded that there are additional areas in which the rough k -means needs further improvement, namely, in terms of selection of parameters. Mitra [16] describes how the selection of parameters can be made with the help of genetic algorithms and Davies–Boulden index. Mitra *et al.* [17] have further proposed a collaborative clustering that combines the advantages of rough k -means and fuzzy c -means approach. The proposed collaborative clustering is shown to be effective for practical clustering approaches.

Ho and Nguyen [15] have proposed a tolerance rough set model based variant of k -means algorithm to find overlapping clusters. This method was aimed at achieving non-hierarchical document clustering. More details about the approach can also be found in [42]. There are a number of other research efforts that combine rough set theory with clustering [12, 13, 20, 21]. However, these approaches tend to use conventional representation of clusters, which are created based on the properties of the rough set theory.

46.8 Feasibility for Large Data Sets

All the four approaches described in detail in this chapter have been shown to produce lower and upper bound representations of clusters. The GA-based approach was shown to successfully cluster highway sections [22]. The Kohonen and k -means approaches were used to cluster Web users as well as supermarket customers [3, 24–26]. Analysis of the results suggested semantic advantages in representing clusters based on lower and upper bounds. The resulting clusters from these first three approaches were found to be similar. The differences between these approaches are mostly related to the computational efficiency. The GA-based approach has the highest time requirements and was found to be infeasible for large number of objects. Both the k -means- and Kohonen-based approaches successfully managed to create rough set representations of clusters for as many as 50,000 supermarket customers. The k -means-based approach tended to converge faster than the Kohonen-based method. The RSVC approach has been demonstrated with a synthetic data set as well as with the widely used Iris data set. The time requirements for RSVC approach are comparable to GA-based approach.

The refined k -means algorithm [19] and the collaborative rough and fuzzy c -means approach [16] have been successfully used for data sets for more than 500,000 objects represented using 54 features.

46.9 Conclusion

This chapter describes how rough or interval sets representations of clusters make it possible for an object to belong to more than one cluster. These representations can be more descriptive than the conventional clustering and less verbose than fuzzy clustering. Four different approaches are presented which are modifications of genetic-algorithm-based clustering, k -means algorithm, the Kohonen self-organizing maps, and support vector clustering using the concept of lower and upper bounds. The first three approaches use

the conventional Euclidean distance for determining cluster memberships. The rough support vector clustering uses non-linear transformation of the input data space, making it possible to create irregular-shaped cluster boundaries.

The chapter presents theoretical and experimental comparisons of RKM and RSVC. The clusters obtained from RKM tend to be farther apart because they obey one rough set property insisting that any object that does not belong to any lower bound must belong to more than one upper bound. If we relax this property, it may be easier to deal with noise and outliers. However, since RSVC does not satisfy this property, it may be more appropriate to refer to the resulting clusters as interval sets. Further investigations on the effect of these rough set properties may provide us more insight into rough clustering.

The chapter also briefly discusses other rough-set-based clustering approaches including various modifications and extensions to the rough k -means approach. In addition, the feasibility of these approaches to large data sets is also discussed. While GA-based and RSVC approaches are not applicable to large data sets, rough- k -means-based clustering has been shown to work for data sets with more than 500,000 objects represented using more than 50 features.

Acknowledgment

The authors thank the Natural Sciences and Engineering Research Council of Canada for partial funding of this research.

References

- [1] A. Joshi and R. Krishnapuram. Robust fuzzy clustering methods to support web mining. In: *Proceedings of the Workshop on Data Mining and Knowledge Discovery*, SIGMOD '98, Vol. 15, June 2–4, 1998, Seattle, Washington, pp. 1–8.
- [2] P. Lingras and G. Adams. *Selection of Time-Series for Clustering Supermarket Customers*. Technical Report 2002_006. Department of Mathematics and Computing Science, Saint Mary's University, Halifax, N.S., Canada. 2002. http://cs.stmarys.ca/tech_reports/, accessed December 2007.
- [3] P. Lingras, R. Yan, and C. West. Fuzzy c-means clustering of web users for educational sites. In: *Proceedings of Sixteenth Conference of the Canadian Society of Computational Studies of Intelligence Advances in Artificial Intelligence Series*, 2671, Toronto, Springer, 2003, pp. 557–562.
- [4] W. Pedrycz and J. Waletzky. Fuzzy clustering with partial supervision. *IEEE Trans. Syst. Man Cybern.* 27(5) (1997) 787–795.
- [5] M. Banerjee, S. Mitra, and S.K. Pal. Rough fuzzy MLP: Knowledge encoding and classification. *IEEE Trans. Neural Netw.* 9(6) (1998) 1203–1216.
- [6] Y. Li, S.C.K. Shiu, S.K. Pal, and J.N.K. Liu. A rough set-based case-based reasoner for text categorization. *Int. J. Approx. Reason.* 41(2) (2006) 229–255.
- [7] S.H. Nguyen, T.T. Nguyen, and H.S. Nguyen. *Rough Set Approach to Sunspot Classification Problem*. RSFDGrC 2005, Regina, Canada, August 31–September 3, 2005. Advances in Artificial Intelligence Series 3641, Springer, New York, 2005, pp. 263–272.
- [8] Z. Pawlak. Rough sets. *Int. J. Inf. Comput. Sci.* 11 (1982) 145–172.
- [9] Z. Pawlak. Rough classification. *Int. J. Man-Mach. Stud.* 20 (1984) 469–483.
- [10] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, 1992.
- [11] S. Saha, C.A. Murthy, and S.K. Pal. Rough set based ensemble classifier for web page classification. *Fundam. Inf.* 76(1–2) (2007) 171–187.
- [12] S. Hirano and S. Tsumoto. Rough clustering and its application to medicine. *Inf. Sci.* 124 (2000) 125–137.
- [13] S. Hirano, X. Sun, and S. Tsumoto. Comparison of clustering methods for clinical databases. *Inf. Sci.* 159 (2004) 155–165.
- [14] S. Hirano and S. Tsumoto. On Constructing Clusters from Non-Euclidean Dissimilarity Matrix by Using Rough Clustering, JSAI Workshops, June 13–14, 2005, Kitakyushu City, Japan, pp. 5–16.
- [15] T.B. Ho and N.B. Nguyen. Nonhierarchical document clustering by a tolerance rough set model. *Int. J. Intell. Syst.* 17(2) (2002) 199–212.
- [16] S. Mitra. An evolutionary rough partitive clustering. *Pattern Recognit. Lett.* 25 (2004) 1439–1449.

- [17] S. Mitra, H. Bank, and W. Pedrycz. Rough-fuzzy collaborative clustering. *IEEE Trans. Syst. Man Cybern.* 36(4) (2006) 795–805.
- [18] G. Peters. Outliers in rough k-means clustering. In: *Proceeding of First International Conference on Pattern Recognition and Machine Intelligence* (Springer LNCS 3776), Kolkata, India, 2005, pp. 702–707.
- [19] G. Peters. Some refinements of rough k-means. *Pattern Recognit.* 39(8) (2006) 1481–1491.
- [20] J.F. Peters, A. Skowron, Z. Suraj, W. Rzasa, and M. Borkowski. Clustering: A rough set approach to constructing information granules. In: *Soft Computing and Distributed Processing, Proceedings of 6th International Conference*, June 24–25, 2002, Rzeszow, Poland, pp. 57–61.
- [21] K.E. Voges, N.K. Li. Pope, and M.R. Brown. Cluster analysis of marketing data: A comparison of k-means, rough set, and rough genetic approaches. In: H.A. Abbas, R.A. Sarker, and C.S. Newton (eds), *Heuristics and Optimization for Knowledge Discovery*. Idea Group Publishing, Hershey, PA, 2002, pp. 208–216.
- [22] P. Lingras. Unsupervised rough set classification using GAs. *J. Intell. Inf. Syst.* 16(3) (2001) 215–228.
- [23] P. Lingras. Rough set clustering for web mining. In: *Proceedings of 2002 IEEE International Conference on Fuzzy Systems*. May 12–17, 2002, Honolulu, Hawaii.
- [24] P. Lingras and C. West. Interval set clustering of web users with rough k-means. *J. Intell. Inf. Syst.* 23(1) (2004) 5–16.
- [25] P. Lingras, M. Hogo, M. Snorek, and B. Leonard. Clustering supermarket customers using rough set based kohonen networks. In: *Proceedings of Fourteenth International Symposium on Methodologies for Intelligent Systems, Lecture Notes in Artificial Intelligence Series*, 2871, Springer, Berlin, 2003, pp. 169–173.
- [26] P. Lingras, M. Hogo, and M. Snorek. Interval set clustering of web users using modified kohonen self-organizing maps based on the properties of rough sets. *Web Intell. Agent Syst. Int. J.* 2(3) (2004) 217–230.
- [27] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. Support vector clustering. *J. Mach. Learn. Res.* 2 (2001) 125–137.
- [28] R. Fletcher. *Practical Methods of Optimization*, 2nd ed. Wiley Interscience, New York, 2000.
- [29] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discovery* 2(2) (1998) 121–167.
- [30] S. Ashraf, S.K. Shevade, and N.M. Murty. Rough support vector clustering. *Pattern Recognit.* 38(10) (2005) 1779–1783.
- [31] Y.Y. Yao and T.Y. Lin. Generalization of rough sets using modal logic. *Intell. Automat. Soft Comput.* 2(2) (1996) 103–120.
- [32] Y.Y. Yao. Constructive and algebraic methods of the theory of rough sets. *Inf. Sci.* 109 (1998) 21–47.
- [33] L. Polkowski and A. Skowron. Rough mereology: A new paradigm for approximate reasoning. *Int. J. Approx. Reason.* 15(4) (1996) 333–365.
- [34] A. Skowron and J. Stepaniuk. Information granules in distributed environment. In: N. Zhong, A. Skowron, and S. Ohsuga (eds), *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, Lecture notes in Artificial Intelligence*, Springer-Verlag, Tokyo, 1711, 1999, pp. 357–365.
- [35] B.P. Buckles and F.E. Petry. *Genetic Algorithms*. IEEE Computer Press, Los Alamitos, CA, 1994.
- [36] J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.
- [37] M. Wall. *Galib, A C++ Library of Genetic Components*. 1993. <http://lancet.mit.edu/ga/>, accessed November 2007.
- [38] S.C. Sharma and A. Werner. Improved method of grouping provincewide permanent traffic counters. *Transport. Res. Record* 815 (1981) 13–18.
- [39] J.A. Hartigan and M.A. Wong. Algorithm AS136: A k-means clustering algorithm. *Appl. Stat.* 28 (1979) 100–108.
- [40] J. MacQueen. Some methods for classification and analysis of multivariate observations. In: *Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, June 21–July 18, 1965 and December 27, 1965–January 7, 1966, Berkeley, CA, 1967, pp. 281–297.
- [41] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 1988.
- [42] H.S. Nguyen. Rough document clustering and the internet. In: *Handbook on Granular Computing*. John Wiley & Sons, Hoboken, NJ, 2008, pp. 987–1005.

47

Rough Document Clustering and The Internet

Hung Son Nguyen and Tu Bao Ho

47.1 Introduction

Granular computing (GrC) was characterized as a common name of theories, methodologies, techniques, and tools that make use of granules (groups, classes, or clusters) in the process of problem solving [1]. From this point of view clustering can be treated as an information granulation process, which is also the main step of computing with words (CW) approach.

Rough set theory has been introduced by Pawlak [2] as a tool for concept approximation under uncertainty. The idea is to approximate the concept by two descriptive sets called *lower* and *upper approximations*. The main philosophy of rough set approach to concept approximation problem is based on minimizing the difference between upper and lower approximations (also called the *boundary region*). This simple, but brilliant idea, leads to many efficient applications of rough sets in machine learning, data mining, and also in granular computing.

The connection between rough set theory and granular computing was examined by many researchers. In [3–6] some particular semantics and interpretations of information granules were defined and some algorithms for constructing granules were given. Many clustering methods based on rough sets and other computational intelligence techniques were proposed including support vector machine (SVM) [7], genetic algorithm (GA) [8, 9], and modified self-organizing map (SOM) [10]. The rough-set-based clustering methods were applied to many real-life applications, e.g., medicine [11], Web user clustering [10, 12], and marketing [9].

This chapter presents the rough set approach to document clustering and its application in search engine technology, particularly, in the Web search result clustering problem. Let us explain the problem more precisely and present its current state of the art.

Two most popular approaches to facilitate searching for information on the Web are represented by Web search engine and Web directories. Web search engines¹ allow user to formulate a query, to which it responds using its index to return set of references to relevant Web documents (Web pages). Web directories² are human-made collection of references to Web documents organized as hierarchical structure of categories.

¹ E.g., Altavista (<http://www.altavista.com>), AllTheWeb (<http://www.alltheweb.com>), Google (<http://www.google.com>), HotBot (<http://www.hotbot.com>), and Lycos (<http://www.lycos.com>).

² E.g., Yahoo (<http://www.yahoo.com>) or Open Directory Project (<http://www.dmoz.org>).

Although the performance of search engines is improving every day, searching on the Web can be a tedious and time-consuming task because (1) search engines can index only a part of the ‘indexable Web,’ due to the huge size and highly dynamic nature of the Web, and (2) the user’s ‘intention behind the search’ is not clearly expressed by too general, short queries. As the effects of the two above, results returned by search engine can count from hundreds to hundreds of thousands of documents.

One approach to manage the large number of results is clustering. The concept arises from document clustering in information retrieval domain: *find a grouping for a set of documents so that documents belonging to the same cluster are similar and documents belonging to different cluster are dissimilar*. Search results clustering can thus be defined as *a process of automatical grouping search results into thematic groups*. However, in contrast to traditional document clustering, clustering of search results are done on the fly and locally on a limited set of results returned from the search engine. Clustering of search results can help user navigate through large set of documents more efficiently. By providing concise, accurate description of clusters, it lets user localize interesting document faster.

Despite being derived from document clustering, methods for clustering Web search results differ from its ancestors on numerous aspects. Most notably, document clustering algorithms are designed to work on relatively large collection of full-text document (or sometimes document abstract). In opposite, the algorithm for Web search results clustering is supposed to work on moderate-size (several hundreds of elements) set of text snippets (with length 10–20 words).

In document clustering, the main emphasis is put on the quality of clusters and the scalability to large number of documents, as it is usually used to process the whole document collection (e.g., for document retrieval on clustered collection). For Web search results clustering, apart from delivering good-quality clusters, it is also required to produce meaningful, concise description for cluster. Additionally, the algorithm must be extremely fast to process results online (as postprocessing search results before delivered to the user) and scalability with the increase of user requests (e.g., measures as number of processed requests per specified amount of time).

	Document clustering	Web search result clustering
Objects	Full-text documents or document abstracts	Short text snippets
Processing mode	Offline processing on a large collection	Online processing of moderate-size set of snippets
Quality measurement	Cluster quality	Cluster quality and cluster description meaningfulness
Computation requirement	Scalability with number of documents	Scalability with number of user requests

The earliest works on clustering results were done by Hearst and Pedersen on scatter/gather system [13], followed with application to Web documents and search results by Zamir and Etzioni [14] to create grouper based on novel algorithm suffix tree clustering. Inspired by their work, a Carrot framework was created by Weiss [15] to facilitate research on clustering search results. This has encouraged others to contribute new clustering algorithms under the Carrot framework like LINGO [16, 17] and AHC [18].

In this chapter, we proposed an approach to search results clustering based on tolerance rough set following the work on document clustering of Bao [19, 20]. The main problem that occurs in all mentioned works is based on the fact that many snippets remain unrelated because of their short representation (see Table 47.3). Tolerance classes are used to approximate concepts existed in documents and to enrich the vector representation of snippets. Set of documents sharing similar concepts are grouped together to form clusters. Concise, intelligible cluster labels are next derived from tolerance classes using special heuristic.

47.2 Rough Sets and Tolerance Rough Set Model

Rough set theory was originally developed [2, 21] as a tool for data analysis and classification. It has been successfully applied in various tasks, such as feature selection/extraction, rule synthesis, and classification [21]. In this chapter we will present fundamental concepts of rough sets with illustrative examples. Some extensions of rough set are described, concentrating on the use of rough set to synthesize approximations of concepts from data.

Consider a non-empty set of objects U called the universe. Suppose we want to define a concept over universe of objects U . Let us assume that our concept can be represented as subset X of U . The central point of rough set theory is the notion of set approximation: any set in U can be approximated by its *lower* and *upper approximation*.

47.2.1 Generalized Approximation Spaces

The classical rough set theory is based on equivalence relation that divides the universe of objects into disjoint classes. By definition, an equivalence relation $R \subseteq U \times U$ is required to be reflexive, symmetric, and transitive. Practically, for some applications, the requirement for equivalent relation has shown to be too strict. The nature of the concepts in many domains is imprecise and can be overlapped additionally. For example, let us consider a collection of scientific documents and keywords describing those documents. It is clear that each document can have several keywords and a keyword can be associated with many documents. Thus, in the universe of documents, keywords can form overlapping classes.

Skowron [22] has introduced a generalized tolerance space by relaxing the relation R to a tolerance relation, where transitivity property is not required. Formally, the generalized approximation space is defined as a quadruple $\mathcal{A} = (U, I, v, P)$, where

- U is a non-empty universe of objects.
- $I : U \rightarrow \mathcal{P}(U)$, where $\mathcal{P}(U)$ is a power set of U , is an *uncertainty function* satisfying conditions: (1) $x \in I(x)$ for $x \in U$, and (2) $y \in I(x) \iff x \in I(y)$ for any $x, y \in U$. Thus, the relation $x R y \iff y \in I(x)$ is a tolerance relation and $I(x)$ is a tolerance class of x .
- $v : \mathcal{P}(U) \times \mathcal{P}(U) \rightarrow [0, 1]$ is a *vague inclusion function*. Vague inclusion v measures the degree of inclusion between two sets. Vague inclusion must be *monotone* with respect to the second argument; i.e., if $Y \subseteq Z$, then $v(X, Y) \leq v(X, Z)$ for $X, Y, Z \subseteq U$.
- $P : I(U) \rightarrow \{0, 1\}$ is a *structurality function*.

Together with uncertainty function I , vague inclusion function v defines the *rough membership function* for $x \in U$, $X \subseteq U$ by $\mu_{I,v}(x, X) = v(I(x), X)$. Lower and upper approximations of any $X \subseteq U$ in \mathcal{A} , denoted by $\mathbf{L}_{\mathcal{A}}(X)$ and $\mathbf{U}_{\mathcal{A}}(X)$, are respectively defined as

$$\begin{aligned}\mathbf{L}_{\mathcal{A}}(X) &= \{x \in U : P(I(x)) = 1 \wedge v(I(x), X) = 1\}, \\ \mathbf{U}_{\mathcal{A}}(X) &= \{x \in U : P(I(x)) = 1 \wedge v(I(x), X) > 0\}.\end{aligned}$$

With definition given above, generalized approximation spaces can be used in any application where I , v , and P are appropriately determined.

47.2.2 Tolerance Rough Set Model

Tolerance rough set model (TRSM) was developed [19, 20] as basis to model documents and terms in information retrieval, text mining, etc. With its ability to deal with vagueness and fuzziness, tolerance rough set seems to be a promising tool to model relations between terms and documents. In many information retrieval problems, especially in document clustering, defining the similarity relation between document–document, term–term, or term–document is essential. In vector space model, it has been noticed [20] that a single document is usually represented by relatively few terms. This results in zero-valued

similarities, which decreases quality of clustering. The application of TRSM in document clustering was proposed as a way to enrich document and cluster representation with the hope of increasing clustering performance.

The idea is to capture conceptually related index terms into classes. For this purpose, the tolerance relation R is determined as the cooccurrence of index terms in all documents from D . The choice of cooccurrence of index terms to define tolerance relation is motivated by its meaningful interpretation of the semantic relation in context of IR and its relatively simple and efficient computation.

Let $D = \{d_1, \dots, d_N\}$ be a set of documents and $T = \{t_1, \dots, t_M\}$ a set of *index terms* for D . With the adoption of vector space model [23], each document d_i is represented by a weight vector $[w_{i1}, \dots, w_{iM}]$, where w_{ij} denotes the weight of term t_j in document d_i . TRSM is an approximation space $\mathcal{R} = (T, I_\theta, v, P)$ determined over the set of terms T as follows:

- **Uncertain function:** The parameterized uncertainty function I_θ is defined as

$$I_\theta(t_i) = \{t_j \mid f_D(t_i, t_j) \geq \theta\} \cup \{t_i\},$$

where $f_D(t_i, t_j)$ denotes the number of documents in D that contain both terms t_i and t_j . Clearly, the above function satisfies conditions of being reflexive: $t_i \in I_\theta(t_i)$ and symmetric: $t_j \in I_\theta(t_i) \iff t_i \in I_\theta(t_j)$ for any $t_i, t_j \in T$. Thus, the tolerance relation $\mathcal{I} \subseteq T \times T$ can be defined by means of function I_θ :

$$t_i \mathcal{I} t_j \iff t_j \in I_\theta(t_i),$$

where θ is a parameter set by an expert.

The set $I_\theta(t_i)$ is called the *tolerance class* of index term t_i .

- **Vague inclusion function:** To measure degree of inclusion of one set in another, the vague inclusion function is defined as

$$v(X, Y) = \frac{|X \cap Y|}{|X|}.$$

It is clear that this function is monotonous with respect to the second argument.

- **Structural function:** All tolerance classes of terms are considered as structural subsets: $P(I_\theta(t_i)) = 1$ for all $t_i \in T$.

The membership function μ for $t_i \in T$, $X \subseteq T$ is then defined as $\mu(t_i, X) = v(I_\theta(t_i), X) = \frac{|I_\theta(t_i) \cap X|}{|I_\theta(t_i)|}$ and the lower and upper approximations of any subset $X \subseteq T$ can be determined – with the obtained tolerance $\mathcal{R} = (T, I, v, P)$ – in the standard way

$$\mathbf{L}_{\mathcal{R}}(X) = \{t_i \in T \mid v(I_\theta(t_i), X) = 1\},$$

$$\mathbf{U}_{\mathcal{R}}(X) = \{t_i \in T \mid v(I_\theta(t_i), X) > 0\}.$$

47.2.3 Example

Consider a universe of unique terms extracted from a set of search result snippets returned from Google search engine for a ‘famous’ query: **jaguar**, which is frequently used as a test query in information retrieval because it is a polysemy, i.e., word that has several meanings, especially in the Web. The word jaguar can have the following meanings:

- jaguar as a cat (*panthera onca* – <http://dspace.dial.pipex.com/agarman/jaguar.htm>);
- jaguar as a Jaguar car;
- jaguar was a name for a game console made by Atari – <http://www.atari-jaguar64.de>;
- it is also a codename for Apple’s newest operating system MacOS X – <http://www.apple.com/macosx>.

Table 47.1 Tolerance classes of terms generated from 200 snippets return by Google search engine for a query ‘jaguar’ with $\theta = 9$

Term	Tolerance class	Document frequency
Atari	Atari, Jaguar	10
Mac	Mac, Jaguar, OS, X	12
onca	onca, Jaguar, Panthera	9
Jaguar	Atari, Mac, onca, Jaguar, club, Panthera, new, information, OS, site, Welcome, X, Cars	185
club	Jaguar, club	27
Panthera	onca, Jaguar, Panthera	9
new	Jaguar, new	29
information	Jaguar, information	9
OS	Mac, Jaguar, OS, X	15
site	Jaguar, site	19
Welcome	Jaguar, Welcome	21
X	Mac, Jaguar, OS, X	14
Cars	Jaguar, Cars	24

Tolerance classes are generated for threshold $\theta = 9$. It is interesting to observe (Table 47.1) that the generated classes do reveal different meanings of the word ‘jaguar’: a cat, a car, a game console, an operating system, and some more.

In context of information retrieval, a tolerance class represents a concept that is characterized by terms it contains. By varying the threshold θ , one can control the degree of relatedness of words in tolerance classes (or the preciseness of the concept represented by a tolerance class).

47.3 Applications of TRS Model in Text Mining

One interpretation of the given approximations can be as follows: if we treat X as a concept described vaguely by index terms it contains, then $\mathbf{U}_R(X)$ is the set of concepts that share some semantic meanings with X , while $\mathbf{L}_R(X)$ is a ‘core’ concept of X . Let us mention two basic applications of TRSM in text mining area.

47.3.1 Enriching Document Representation

In standard vector space model, a document is viewed as a *bag of words/terms*. This is articulated by assigning non-zero-weight values in document’s vector to terms that occur in document. With TRSM, the aim is to enrich representation of the document by taking into consideration not only terms actually occurring in document but also other related terms with similar meanings. A ‘richer’ representation of the document can be acquired by representing the document as a set of tolerance classes of terms it contains. This is achieved by simply representing the document with its upper approximation; i.e., the document $d_i \in D$ is represented by

$$d_i \rightarrow \mathbf{U}_R(d_i) = \{t_i \in T \mid v(I_\theta(t_i), d_i) > 0\}.$$

In fact, one can apply the enriched representation scheme to any collection of words, e.g., clusters. Moreover, composition of the upper approximation operator several times can return even more richer

representation of a document; i.e.,

$$d_i \rightarrow \mathbf{U}_R^k(d_i) = \underbrace{\mathbf{U}_R(\dots \mathbf{U}_R(d_i))}_{k \text{ times}}.$$

The use of upper approximation in similarity calculation to reduce the number of zero-valued similarities is the main advantage the TRSM-based algorithms claimed to have over traditional approaches. This makes the situation, in which two documents have a non-zero similarity although they do not share any terms, possible.

47.3.2 Extended Weighting Scheme

To assign weight values for document's vector, the TF*IDF weighting scheme is used. In order to employ approximations for document, the weighting scheme need to be extended to handle terms that occur in document's upper approximation but not in the document itself. The extended weighting scheme is defined from the standard TF*IDF by

$$w_{ij}^* = \begin{cases} (1 + \log f_{d_i}(t_j)) \log \frac{N}{f_D(t_j)} & \text{if } t_j \in d_i \\ 0 & \text{if } t_j \notin \mathbf{U}_R(d_i) \\ \min_{t_k \in d_i} w_{ik} \frac{\log \frac{N}{f_D(t_j)}}{1 + \log \frac{N}{f_D(t_j)}} & \text{otherwise.} \end{cases}$$

The extension ensures that each term occurring in upper approximation of d_i , but not in d_i has a weight smaller than the weight of any term in d_i . Normalization by vector's length is then applied to all document vectors:

$$w_{ij}^{new} = \frac{w_{ij}^*}{\sqrt{\sum_{t_k \in d_i} (w_{ij}^*)^2}}.$$

Table 47.2 presents an example of the enriched vector representation of a snippet and the extended weighting scheme.

47.4 Document Clustering Algorithms Based on TRSM

With the introduction of TRSM in [19], several document clustering algorithms based on that model were also introduced [19, 20]. The main novelty that TRSM brings into clustering algorithms is the way of representing clusters and document.

47.4.1 Cluster Representation

Determining cluster representation is a very important factor in partitioning-based clustering. Frequently, cluster is represented as a mean or median of all documents it contains. Sometimes, however, a representation not based on vector is needed as cluster description is directly derived from its representation. For example, cluster can be represented by most 'distinctive' terms from cluster's documents (e.g., most frequent terms or frequent terms in clusters but infrequent globally).

In [20], an approach to construct a *polythetic* representation is presented. Let R_k denote a representative for cluster k . The aim is to construct a set of index terms R_k , representing cluster C_k , so that

- each document d_i in C_k share some or many terms with R_k ;
- terms in R_k occur in most documents in C_k ;
- terms in R_k need not to be contained by every document in C_k .

Table 47.2 Example of snippet and its two vector representations

Title: EconPapers: Rough set bankruptcy prediction models versus auditor

Description: Rough set bankruptcy prediction models versus auditor signalling rates. *J. Forecast.* 22(8) (2003), 569–586. Thomas E. McKee. . . .

Original vector		Enriched vector	
Term	Weight	Term	Weight
auditor	0.567	auditor	0.564
bankruptcy	0.4218	bankruptcy	0.4196
signalling	0.2835	signalling	0.282
EconPapers	0.2835	EconPapers	0.282
rates	0.2835	rates	0.282
versus	0.223	versus	0.2218
issue	0.223	issue	0.2218
Journal	0.223	Journal	0.2218
MODEL	0.223	MODEL	0.2218
prediction	0.1772	prediction	0.1762
Vol	0.1709	Vol	0.1699
		applications	0.0809
		Computing	0.0643

The weighting for terms t_j in R_k is calculated as an averaged weight of all occurrences in documents of C_k :

$$w_{kj} = \frac{\sum_{d_i \in C_k} w_{ij}}{|\{d_i \in C_k \mid t_j \in d_i\}|}.$$

Let $f_{C_k}(t_j)$ be the number of documents in C_k that contain t_j . The above assumptions lead to the following rules to create cluster representatives:

To the initially empty representative set, terms that occur frequent enough (controlled by threshold σ) in documents within cluster are added. After this phase, for each document that is not yet ‘represented’ in representative set (i.e., document shares no terms with R_k), the strongest/heaviest term from that document is added to the cluster representatives.

Algorithm 1. Determine cluster representatives.

```

1:  $R_k = \emptyset$ 
2: for all  $d_i \in C_k$  and  $t_j \in d_i$  do
3:   if  $f_{C_k}(t_j)/|C_k| > \sigma$  then
4:      $R_k = R_k \cup t_j$ 
5:   end if
6: end for
7: if  $d_i \in C_k$  and  $d_i \cap R_k = \emptyset$  then
8:    $R_k = R_k \cup \text{argmax}_{t_j \in d_i} w_{ij}$ 
9: end if

```

47.4.2 TRSM-Based Clustering Algorithms

Both hierarchical and non-hierarchical clustering algorithms were proposed and evaluated with standard test collections and have shown some successes [19, 20].

The non-hierarchical document clustering algorithm based on TRSM is a variation of a K -means clustering algorithm with overlapping clusters. The modifications introduced are

- use of document's upper approximation when calculating document–cluster and document–document similarity,
- documents are soft assigned to cluster with associated membership value,
- use nearest neighbor to assign unclassified documents to cluster.

A hierarchical agglomerative clustering algorithm based on TRSM utilizes upper approximation to calculate cluster similarities in the merging step.

47.5 Case Study: The TRSM-Based Snippets Clustering Algorithm

The tolerance rough set clustering algorithm is based primarily on the k -means algorithm presented in [20]. By adapting k -means clustering method, the algorithm remain relatively quick (which is essential for online results postprocessing) while still maintaining good clusters quality. The usage of tolerance space and upper approximation to enrich interdocument and document–cluster relation allows the algorithm to discover subtle similarities not detected otherwise. As it has been mentioned, in search results clustering, the proper labeling of cluster is as important as cluster contents quality. Since the use of phrases in cluster label has been proved [14, 16] to be more effective than single words, TRC algorithm utilize n -g of words (phrases) retrieved from documents inside cluster as candidates for cluster description.

The TRC algorithm is composed of five phases (depicted in Figure 47.1):

1. **Documents preprocessing:** In TRC, the following standard preprocessing steps are performed on snippets: *text cleansing*, *text stemming*, and *stop-words elimination*.
2. **Documents representation building:** In this step, two main procedures are performed: *index term selection* and *term weighting*.
3. **Tolerance class generation:** The goal of the tolerance class generation is to determine for each term set of its related terms with regard to the tolerance relation – the tolerance class.
4. **Clustering:** Applying k -means clustering algorithm based on TRSM.
5. **Cluster labeling:** For labeling cluster we have decided to employ phrases because of its high descriptive power [14, 16]. We have adapted an algorithm for n -g generation from [24] to extract phrases from contents of each cluster. Most descriptive phrases are chosen to serve as labels for cluster.

The details of the last three steps are described in the following sections.

While document clustering deals with full-size documents, in clustering search results we have only set of small-size snippets.

47.5.1 Preprocessing

It is widely known [25] that preprocessing text data before feeding it into clustering algorithm is essentials and can have great impact on algorithm performance. In TRC, several preprocessing steps are performed on snippets:

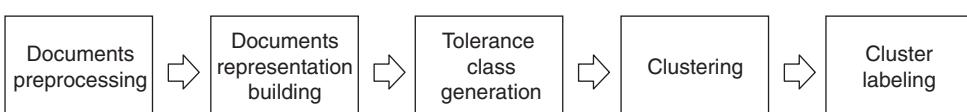


Figure 47.1 Phases of TRC algorithm

Text cleaning. In this step, text content of snippet is cleaned from unusable terms such as:

- non-letter characters like , \$, #
- HTML-related tags (e.g., = <a>, <p>) and entities (e.g., &, ")

Stemming. A version of Porter's stemmer [26] is used in this step to remove prefixes and suffixes, normalizing terms to its root form. This process can greatly reduce vocabulary of the collection without much semantic loss. The stemmed terms are linked to its original form, which are preserved to be used in subsequent phases (i.e., labels generation).

Stop-words elimination. A stop word itself does not bring any semantic meanings but in connection with other words can form meaningful phrases. Therefore, terms that occur in stop-word list are specially marked to be ignored from document index terms, but not removed (so it can be used in phrases extraction in label generation phase). Due to special nature of Web document, some words like 'web,' 'http,' and 'site' appear very frequently, thus a stop-word list adapted to Web vocabulary from [14] is used.

47.5.2 Document Corpus Building

As TRC utilizes vector space model for creating document-term matrix representing documents.

Index terms selection. Index terms are selected from all unique stemmed terms after stop-words elimination and with regard to the following rules:

- Digits and terms shorter than two characters are ignored.
- Terms contained in the query are ignored. (As we are operating on the top search results for a query, terms from query will occur in almost every snippet.)
- Minimum document frequency filtering – term that occurs in less given threshold (e.g., less than two snippets) are ignored as they will doubly contribute to document characterization.

Selected index terms used to characterize documents are enumerated and a document-term frequency matrix is built. Let N be a number of document search results snippets and M is the number of selected index terms. The document-term frequency matrix is defined as follows:

$$TF = [tf_{i,j}]_{N \times M},$$

where $tf_{i,j}$ is number of occurrences of term j in document j . Each row $TF[i]$ of TF matrix is a characterization of the i th document by means of term frequencies.

Term weighting. The TF*IDF term weighting scheme is applied for document-term matrix to create document-term weight matrix:

$$W = [w_{i,j}]_{N \times M},$$

where $w_{i,j}$ is weight of term j in i th document. Each row $W[i]$ in the W matrix represents a characterization of the i th document by means of weighted terms.

47.5.3 Tolerance Class Generation

This phase exists for the computational optimization purpose. It ensures that the calculation of the upper approximation for a set of terms can be done quickly.

Let us define *term cooccurrence matrix* as

$$TC = [tc_{x,y}]_{M \times M},$$

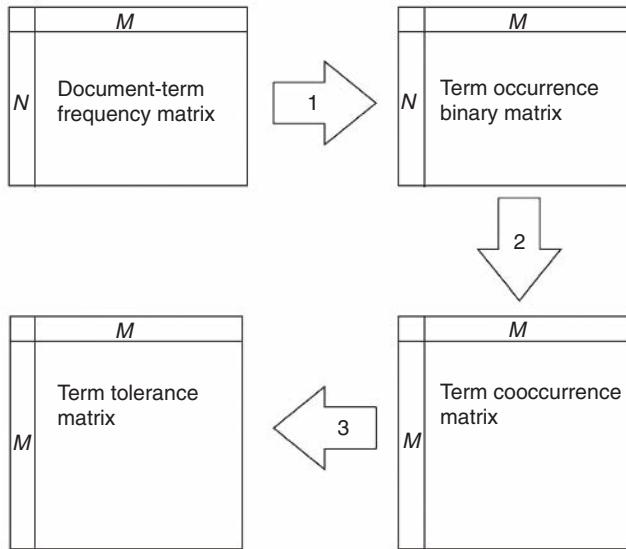


Figure 47.2 Process of generating tolerance classes

where $tc_{x,y}$ is a *cooccurrence frequency* of two terms x and y – number of documents in the collection in which terms x and y both occur. Let tolerance relation R between terms be defined as

$$xRy \iff tc_{x,y} > \theta,$$

where θ is called **cooccurrence threshold**. Having term cooccurrence matrix calculated we can define tolerance relation with different granularity by varying cooccurrence threshold.

Figure 47.2 presents the main steps of the tolerance class generation phase. The computation cost of Step 1 is $O(N \times M)$ and Steps 2 and 3 are both $O(M^2)$; altogether is $O(N \times M^2)$.

The detail implementation of this phase is presented in Algorithm 2.

47.5.4 *K*-Means Clustering

TRC adapts a variation of *K*-means algorithm for creating groups of similar snippets. Several main steps of the algorithm are described below (see pseudocode in Algorithm 3):

Algorithm 2. Tolerance class generation (Figure 47.2).

Input: TF – document-term frequency matrix, θ – cooccurrence threshold

Output: TOL – term tolerance binary matrix defining tolerance classes of term

1: Calculate a binary occurrence matrix OC based on document-term frequency matrix TF as follows:

$$OC = [oc_{i,j}]_{N \times M}, \text{ where}$$

$$oc_{i,j} = \begin{cases} 1 & \text{if } tf_{i,j} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Each column in OC is a bit vector representing term occurrence pattern in a document — bit is set if term occurs in a document.

2: Construct term cooccurrence matrix $COC = [coc_{x,y}]_{M \times M}$ as follows: for each pair of term x, y represented as pair of columns $OC[x], OC[y]$ – bit vectors – in the OC matrix

$$coc_{x,y} = \text{card}(OC^x \text{ AND } OC^y),$$

where AND is a binary AND between bit vectors and $card$ return cardinality – number of bits set – of a bit vector. $boc_{x,y}$ is the cooccurrence frequency of term x and y .

- 3: Given a cooccurrence threshold θ , a term tolerance binary matrix $TOL = [tol_{x,y}]_{M \times M}$ can be easily constructed by filtering out cells with values smaller than threshold θ :

$$tol_{x,y} = \begin{cases} 1 & \text{if } coc_{x,y} \geq \theta \\ 0 & \text{otherwise.} \end{cases}$$

Each row in the resulting matrix forms a bit vector defining a tolerance class for given term: $tol_{x,y}$ is set if terms x and y are in tolerance relation.

Algorithm 3. Clustering phase of TRC algorithm.

Input: D – set of N documents, K – number of clusters, δ – cluster similarity threshold

Output: K overlapping clusters of documents from D with associated membership value

- 1: Randomly select K document from D to serve as K initial cluster C_1, C_2, \dots, C_K .
- 2: **repeat**
- 3: **for** each $d_i \in D$ **do**
- 4: **for** each cluster $C_k, k = 1, \dots, K$ **do**
- 5: calculate the similarity between document's upper approximation and the cluster representatives $S(\mathbf{U}_R(d_i), R_k)$
- 6: **if** $S(\mathbf{U}_R(d_i), R_k) > \delta$ **then**
- 7: assign d_i to C_k with the similarity taken as cluster membership: $m(d_i, C_k) = S(\mathbf{U}_R(d_i), R_k)$
- 8: **end if**
- 9: **end for**
- 10: **end for**
- 11: **for** each cluster C_k **do**
- 12: **determine_cluster_representatives**(R_k)
- 13: **end for**
- 14: **until** stop_condition()
- 15: post-process unassigned documents
- 16: **if** necessary **determine_cluster_representatives**(R_k) **for** changed clusters C_k

Initial cluster forming. Selected documents serve as initial cluster representatives R_1, R_2, \dots, R_K .

Stop condition. The fact that the clustering algorithm is used as a postretrieval³ process puts a strict constraint on execution time, as users are not willing to wait more than a few seconds for a response. We decided to set a limit of **maximum number of iteration** for the K -means algorithm. Due to the nature of quick convergence of K -means algorithm, this limit allows us to reduce the response time of the clustering engine with insignificant loss in clustering quality.

Determining cluster representatives. Cluster representatives is determined as described in Section 47.4.1.

Nearest-neighbor assignment. As a result of the restriction set by **cluster similarity threshold**, after all iterations there may exist document that has not been assigned to any cluster. In TRC, there are two possible options:

- Create a special ‘others’ cluster with unassigned documents as proposed by [16].
- Assign these documents to their nearest cluster.

For the latter option, we have decided to assign document to its nearest-neighbor cluster.

Cluster label generation. As already pointed out, when evaluating clustering algorithms for search results, the quality of cluster label is as much important as the quality of the cluster itself. For labeling cluster we have decided to employ phrases because of its high descriptive power [14, 16].

³ Results returned from search engine are processed on the fly by the clustering algorithm and presented to the user.

We have adapted an algorithm for n -g generation from [24] to extract phrases from contents of each cluster. Most descriptive phrases are chosen to serve as labels for cluster. Descriptiveness of phrases is evaluated by taking into consideration the following criteria:

- frequency of the phrase in the whole collection,
- frequency of the phrase inside a cluster,
- length of the phrase, measured as number of words it is made from.

Following the intuition of TD*IDF scheme, we hypothesize that phrases that are relatively infrequent in the whole collection but occur frequently in clusters will be good candidate for cluster's label. We also prefer long phrases over shorter one.

47.6 Experimental Results

Due to aforementioned lack of standard collection for testing Web search results clustering, we had to build a small test collection. For this purpose, we have defined a set of queries for which results were collected from major search engine Google to form test data collection.

Algorithm 1. Assignment of document to its nearest-neighbor cluster.

```

for each unassigned document  $d_u$  do
    Find the nearest-neighbor document  $NN(d_u)$  with non-zero similarity;
    Among clusters which  $NN(d_u)$  belongs to, choose the one  $C_k$  that  $NN(d_u)$  has strongest membership
    with;
    Assign  $d_u$  to  $C_k$  and calculate its cluster membership as  $m(d_u, C_k) = m(NN(d_u), C_k) \cdot$ 
     $S(\mathbf{U}_R(d_u), \mathbf{U}_R(NN(d_u)))$ ;
end for
```

47.6.1 Test Queries

The test set is composed of queries representing subjects of various degree of specificity in order to test algorithm behavior on data with different vocabulary characteristic.

The first three queries represent very general concepts and are frequently used as a test by authors [14, 15] of search results clustering algorithm. Next three queries are more specific subjects but broad enough to have interesting subclasses. Last three queries are about relatively specific topics and were chosen to test algorithm on highly cohesive vocabulary. In Table 47.3, figures in the second column are the *approximated number of results* for a query retrieved from Google. Looking at these figures gives us some additional intuition about generality (or specificity) of concepts represented by query, or just popularity of each subject on the Web.

Table 47.3 shows some statistics of snippet collection retrieved from Google search engine for set of test queries. Both stemming and stop-word list were used to filter out unnecessary terms. Additionally, minimum document frequency filter was used to remove terms that occur in less than two documents. Thus, the indexing vocabulary could be reduced by about 70%. On average, any snippet is indexed by 6 to 10 terms, which is about 2–3% of total vocabulary. Worth noticed in the results of term filtering, some document may be left represented by none terms.

47.6.2 Interdocument Similarity Enrichment

The main purpose of using upper approximation in our TRC algorithm is to enhance the association between documents, i.e., to increase the similarity between related documents. To measure that enhancement, we compare the *density* of similarity functions created by standard document representation and the one using upper approximation (for all collections of snippets). In our experiments, the

Table 47.3 Queries used to generate test collection and characteristics of retrieved snippets from Google for those queries

Query	Results count	Specificity	Snippets	Terms	Terms per snippet		
					Avg.	Min	Max
Java	23,400,000	Low	200	332	7.280	0	15
Clinton	4,27,0000	Low	200	337	7.305	0	13
Jaguar	2,580,000	Low	200	339	6.595	0	13
‘Data mining’	1,080,000	Medium	200	319	8.815	0	16
Wifi	864,000	Medium	200	340	6.915	1	19
Clustering	810,000	Medium	195	319	7.456	0	16
Voip	916,000	High	200	355	8.235	0	17
‘Rough sets’	748	High	146	244	9.089	1	18
‘Search results clustering’	105	High	68	149	12.279	2	19

similarity between two documents d_i and d_j , also called *interdocument similarity*, is calculated by using cosine similarity measure (see [27]), and denoted by $\text{sim}(d_i, d_j)$. The density of a given similarity function $\text{sim} : D \times D \rightarrow [0, 1]$ over a collection D of documents, is calculated as a number of pairs $(d_i, d_j) \in D \times D$ of documents, such that $\text{sim}(d_i, d_j) > t$, where $t \in [0, 1]$ is called a *similarity level*.

For a given collection D of snippets and the similarity level t , the *relative density improvement of similarity function*, denoted by $\text{improve}_t(D)$, is measured by

$$\frac{\text{dense}_t(\text{sim}_{\text{TRSM}}) - \text{dense}_t(\text{sim})}{\text{dense}_t(\text{sim})},$$

where $\text{dense}_t(\text{sim})$ and $\text{dense}_t(\text{sim}_{\text{TRSM}})$ are densities of two similarity functions defined by two document representations: the standard and the one based on TRSM. In Figure 47.3 (up), the *relative density improvement* of similarity function for tested snippet collections in different similarity levels is presented.

It can be observed that the enrichment of upper approximation had indeed improved interdocument similarities for all queries. The level of improvement varies with different queries, depending on the cooccurrence threshold. Some queries like ‘java,’ ‘clinton,’ ‘voip,’ achieved significantly better level of improvement than others (‘jaguar,’ ‘clustering’). It is promising that improvement has happened in all similarity levels. (The improvement in level $t = 0.67$ was significantly improved.) This is very important for clustering quality as this could form better clusters. The representation enrichment technique results in improvement of the clustering process as it is presented in Figure 47.3 (bottom).

47.6.3 Comparison with Other Approaches

The TRC algorithm was implemented entirely in Java programming language, as a component within the *Carrot²* framework [28]. *Carrot²* is an open-source, data-processing framework developed by Weiss [29]. It enables and ease experimentation of processing and visualization of search results. The *Carrot²* architecture is based on a set of loosely coupled, but cooperating, components that exchanges information with each other using XML messages (see Figure 47.4).

Such implementation of TRC makes the comparison of TRC with other algorithms like LINGO [16], AHC [18], and STC [15] possible, because all algorithms (included our own TRC) were developed within the *Carrot²* framework [29]. Figure 47.5 presents the results of the mentioned algorithms for the query ‘data mining.’ One can see that TRC found several new interesting clusters, e.g., ‘student notes’ or ‘high effective strategy,’ which are not identified other methods.

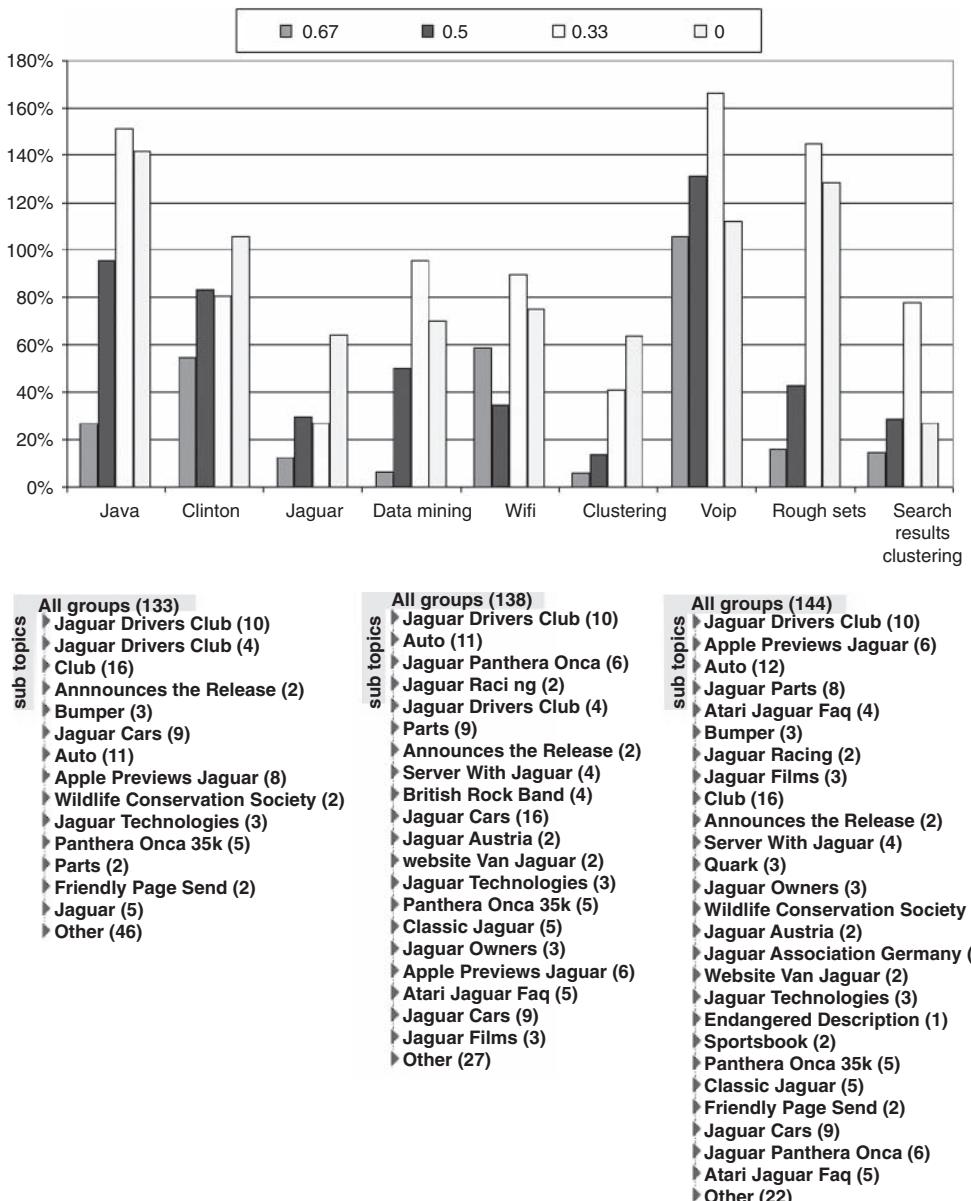


Figure 47.3 Top: Relative improvement of interdocument similarity matrix measure with cooccurrence threshold = 5 and various similarity levels $t = 0, 0.33, 0.5, 0.67$. Bottom: Example of clustering results produced by TRC for query 'jaguar' with different similarity thresholds

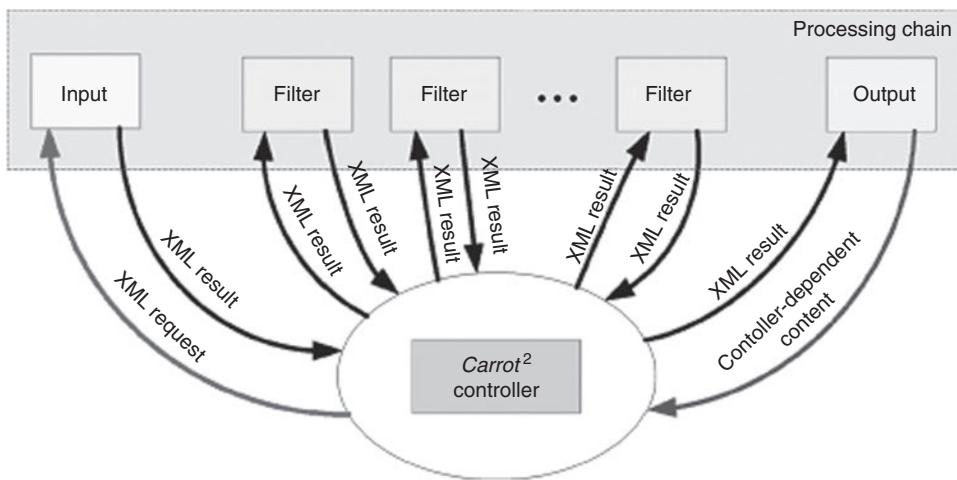


Figure 47.4 Data flow in *Carrot²* framework

47.7 Conclusion

This chapter was thought as a proof-of-concept demonstration of tolerance rough set application to Web mining domain. We wanted to investigate how rough set theory and its ideas, like approximations of concept, could be practically applied in a task of search results clustering. The result is a design of a TRC – a tolerance rough set clustering algorithm for Web search results and implementation of the proposed solution within an open-source framework, *Carrot²*.

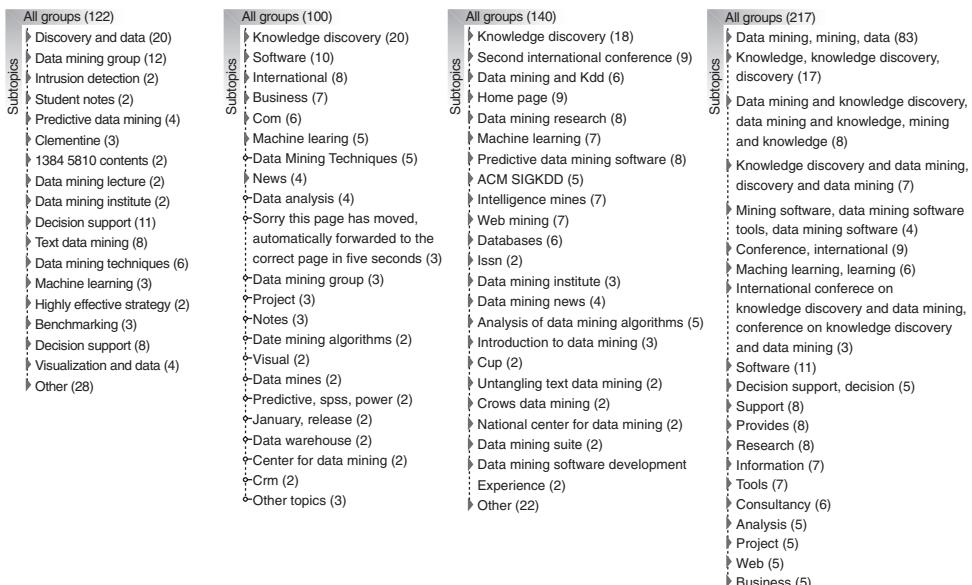


Figure 47.5 Comparison of results for a query 'data mining' produced by different algorithm (from left to right): TRC, LINGO, AHC, and STC. All outputs were produced using *Carrot²* visualization component

The experiment we have carried showed that tolerance rough set and upper approximation it offers can indeed improve the representations, which have positive effects on clustering quality. The results are promising and encourage further work on its application. This research is a step forward to make the application of computing with words (CW) on the Internet possible.

Acknowledgment

The research has been partially supported by the grant N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland and by the grant Innovative Economy Operational Programme 2007–2013 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland.

References

- [1] A. Bargiela and W. Pedrycz. *Granular Computing. An Introduction. The International Series in Engineering and Computer Science*, Vol. 717. Kluwer Academic Publishers, Boston, MA, 2002.
- [2] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer, Dordrecht, 1991.
- [3] Z. Pawlak. Granularity of knowledge indiscernibility and rough sets. In: *Proceedings of the 1998 IEEE International Conference on Computational Intelligence*, Vol. 1, Anchorage, AK, May 04–09, 1998, pp. 106–110.
- [4] L.T. Polkowski and A. Skowron. Towards adaptive calculus of granules. In: *Proceedings of the FUZZ-IEEE International Conference, 1998 IEEE World Congress on Computational Intelligence (WCCI'98)*, Anchorage, Alaska, USA, May 5–9, 1998, pp. 111–116.
- [5] H.S. Nguyen, A. Skowron, and J. Stepaniuk. Granular computing: a rough set approach. *Comput. Intell. Int. J.* 17(3) (2001) 514–544.
- [6] J.F. Peters, A. Skowron, Z. Suraj, W. Rzasa, and M. Borkowski. Clustering: A rough set approach to constructing information granules. In: *Soft Computing and Distributed Processing, Proceedings of 6th International Conference, SCDP 2002*, IOS Press, Amsterdam, The Netherlands, pp. 57–61.
- [7] S. Ashraf, S.K. Shevade, and N.M. Murty. Rough support vector clustering. *Pattern Recognit.* 38(10) (2005) 1779–1783.
- [8] P. Lingras. Unsupervised rough set classification using GAs. *J. Intell. Inf. Syst.* 16(3) (2001) 215–228.
- [9] K.E. Voges, N.K.L.I. Pope, and M.R. Brown. Cluster analysis of marketing data: A comparison of k-means, rough set, and rough genetic approaches. In: H.A. Abbas, R.A. Sarker, and C.S. Newton (eds), *Heuristics and Optimization for Knowledge Discovery*. Idea Group Publishing, Hershey, PA, pp. 208–216.
- [10] P. Lingras, M. Hogo, and M. Snorek. Interval set clustering of web users using modified Kohonen self-organizing maps based on the properties of rough sets. *Web Intell. Agent Syst. Int. J.* 2(3) (2004) 217–230.
- [11] S. Hirano and S. Tsumoto. Rough clustering and its application to medicine. *J. Inf. Sci.* 124 (2000) 125–137.
- [12] P. Lingras and C. West. Interval set clustering of web users with rough k-means. *J. Intell. Inf. Syst.* 23(1) (2004) 5–16.
- [13] M.A. Hearst and J.O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In: *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, Zürich, CH, 1996, pp. 76–84.
- [14] O. Zamir and O. Etzioni. Grouper: A dynamic clustering interface to web search results. *Comput. Netw. (Amsterdam, Netherlands: 1999)* 31(11–16) (1999) 1361–1374.
- [15] D. Weiss. *A Clustering Interface for Web Search Results in Polish and English*. Master's Thesis. Poznan University of Technology, Poland, June 2001.
- [16] S. Osinski. *An Algorithm for Clustering of Web Search Result*. Master's Thesis, Poznan University of Technology, Poland, June 2003.
- [17] S. Osinski and D. Weiss. A concept-driven algorithm for clustering search results. *IEEE Intell. Syst.* 20(3) (2005) 48–54.
- [18] M. Wroblewski. *A Hierarchical WWW Pages Clustering Algorithm Based on the Vector Space Model*. Master's Thesis. Poznan University of Technology, Poland, July 2003.
- [19] S. Kawasaki, N.B. Nguyen, and T.B. Ho. Hierarchical document clustering based on tolerance rough set model. In: D.A. Zighed, H.J. Komorowski, and J.M. Zytkow (eds), *Principles of Data Mining and Knowledge Discovery*,

- 4th European Conference, PKDD 2000, Lyon, France, September 13–16, 2000, Proceedings (2000). Vol. 1910 of *Lecture Notes in Computer Science*. Springer-Verlag, London, UK, pp. 458–463.
- [20] T.B. Ho and N.B. Nguyen. Nonhierarchical document clustering based on a tolerance rough set model. *Int. J. Intell. Syst.* 17(2) (2002) 199–212.
 - [21] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron. Rough sets: A tutorial. In: S. Pal and A. Skowron (eds), *Rough Fuzzy Hybridization*. Springer-Verlag, Berlin, Heidelberg, 1998, pp. 3–98.
 - [22] A. Skowron and J. Stepaniuk. Tolerance approximation spaces. *Fundam. Inf.* 27(2–3) (1996) 245–253.
 - [23] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*, 1st ed. Addison-Wesley Longman Publishing Co. Inc., Boston, MA, 1999.
 - [24] F.A. Smadja. From n-grams to collocations: An evaluation of xtract. In: *29th Annual Meeting of the Association for Computational Linguistics, June 18–21, 1991, University of California, Berkeley, California, USA, Proceedings*, 1991, pp. 279–284.
 - [25] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*, 2nd ed. Morgan Kaufmann, San Francisco, CA, 2006.
 - [26] M.F. Porter. An algorithm for suffix stripping. In: P.W. Karen and S. Jones (eds), *Readings in Information Retrieval*, Morgan Kaufmann, San Francisco, 1997, pp. 130–137.
 - [27] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., Reading, MA, 1989.
 - [28] C.L. Ngo and H.S. Nguyen. A method of web search result clustering based on rough sets. In: A. Skowron, R. Agrawal, M. Luck, T. Yamaguchi, P. Morizet-Mahoudeaux, J. Liu, and N. Zhong (eds), *2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2005)*, September 19–22, 2005, Compiègne, France, IEEE Computer Society, Los Alamitos, CA, pp. 673–679.
 - [29] S. Osinski and D. Weiss. Carrot²: Design of a flexible and efficient web information retrieval framework. In: P.S. Szczępaniak, J. Kacprzyk, and A. Niewiadomski (eds), *AWIC* (2005), Vol. 3528 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Heidelberg, 2005, pp. 439–444.

48

Rough and Granular Case-Based Reasoning

Simon C.K. Shiu, Sankar K. Pal, and Yan Li

48.1 Introduction

Case-based reasoning (CBR) is a reasoning methodology that is based on prior experience and examples. Generally, a CBR reasoner will be presented with a problem, and it then searches its memory of past cases (stored in the case base) and attempts to find a case or multiple cases that most closely match the current case. In most situations, the found cases need to be adapted to meet the requirement of the new problems. CBR systems have been widely used in many applications: design, planning, prediction and classification, knowledge inference and evaluation, and many others [1].

The problem-solving quality of a CBR system is the average quality of the proposed solutions, which can be usually described by accuracy and the required adaptation effort. The accuracy is the percentage of the problems which can be successfully solved. The required adaptation effort is the cost for modifying the proposed solutions derived from the retrieved case(s) to solve the problems. In other words, the basic qualification of a CBR system is whether the proposed solution can be used to solve the new problem. The existence of redundant and noisy cases will cause problem in the case retrieval; i.e., the wrong case(s) may be retrieved even though the correct case is contained in the case base. As a result, the accuracy will decrease and the required adaptation effort will increase.

Secondly, the problem-solving efficiency is also important, which can be defined as the average time for solving a problem. With increasing storage of cases (i.e., solved problems), the case bases tend to become larger and larger, which will slow down the case retrieval speed.

Finally, the completeness of a case base is a measure of its problem-solving coverage of all the potential problems. It can be measured using the concept of competence [2–6], i.e., the range of problems that the case base can successfully solve. The competence of a case base can be described by two important competence properties: the coverage set and the reachability set. The coverage set of a case is the set of all target problems that this case can be used to solve. On the other hand, the reachability set of a target problem is the set of all cases that can be used to solve it. Cases with large coverage sets are more important because they can solve many other problems and therefore should solve many of the future target problems. Cases with small reachability sets are more important because they represent regions of the target problem space that are difficult to solve.

The two performance criteria of accuracy and competence are closely related. Generally speaking, if more cases can be successfully solved by a case base, the accuracy will increase. Therefore, for a given

case base, larger competence achieves higher accuracy. Thus, the desirable characteristics of case bases are: small in size, large in competence, and contain no redundancy and noises.

48.1.1 Information Granules and Cases

The use of information granules allows flexible encoding of case characteristics as real numbers, linguistic terms, fuzzy numbers, and fuzzy complex objects. These case features can also be organized systematically as fuzzy categories (i.e., clusters) or fuzzy granules. In traditional case representation, case knowledge is encoded as crisp concepts. However, in many practical situations, when specifying a query case, it is often difficult to articulate the feature values precisely. If the cases can be organized in conceptually overlapping categories, retrieval could be implemented via a classification and similarity regime which encodes feature characteristics in a mixed fuzzy and crisp format. Use of the fuzzy concept significantly improves the flexibility and expressiveness with which case knowledge can be represented. Further references in fuzzy CBR can be found in [7–14].

The theory of rough sets [15–17] provides a new mathematical framework for analysis of data that are imprecise, vague, and uncertain. The main notion of rough sets is that if objects are similar or indiscernible in a given information system, the same set of information can be used to characterize them. Therefore, there may exist some objects that cannot be classified with certainty as members of the set or of its complement. In a rough-set-theoretic approach, each vague concept is characterized by a pair of precise concepts (called the lower and upper approximations of the vague concept). Rough set approaches deal primarily with the classification of data and synthesizing approximations of concepts. These approaches are also used to construct models that represent the underlying domain theory from a set of data. In CBR system development, it is often necessary to determine some properties from the case data to partition the cases into subsets. However, in real-life situations, it is sometimes impossible to define a classifying concept in a precise (crisp) manner. For example, given a new case, it may not be possible to know which class it belongs to. The best knowledge derived from past cases may only give us enough information to say that this new case belongs to a boundary between certain cases, which may consist of various possible solutions. The formulation of these lower- and upper-set approximations can be generalized to some arbitrary level of precision which forms the basis of rough concept approximations.

Furthermore, the techniques in generating information granules can be applied to select and/or generate the prototypical cases for building CBR system. This set of prototypical cases will then be indexed and retrieved at later stages of the CBR reasoning tasks. Therefore, given a case base, if one can identify that only one element of the equivalence class is needed to represent the entire class, enormous storage space will be saved. The other possible criterion to consider for reduction is to keep only those case features that preserve the indiscernibility relation. Since the attributes rejected are redundant, their removal will not affect the task of case retrieval based on feature similarities. Therefore, selection and generation of cases can be regarded as the two important phases in building good case base for a CBR system. Whereas case selection deals with selecting informative prototypes from the data, case generation concerns the construction of cases that need not necessarily include all the data points given. Thus, the key concepts here are those of information granule and reducts. An information granule formalizes the concept of finite-precision representation of objects in real-life situations and reducts represent the core of an information system in a granular universe. Hence, rough set theory is a natural choice for case selection in domains that are data rich, which contain uncertainties and allow tolerance for imprecision. Additionally, rough sets have the capability to handle complex objects thereby extending the applicability of rough CBR system.

48.1.2 Reasoning Using Cases

The first step in solving a new problem using a CBR system is retrieving the most similar case(s) from the case base. Assuming that similar problems have similar solutions, the case(s) retrieved are used to derive the solution for the new problem. Usually, the past solution needs adjustment to fit the new situation. The process of fixing (i.e., adapting) the old solution is called case adaptation. The process of ‘retrieving’ and ‘adapting’ cases is genuinely referred to as *reasoning using cases*. As motioned before, one of the most important assumptions in CBR is that similar experiences can guide future reasoning, problem solving,

and learning. Most of today's case retrieval algorithms are based on this assumption. However, there are an increasing number of arguments about using this simple feature-based similarity as an explanation of human thinking and categorization. This is because similarity cannot be studied in an isolated manner, and there are many possible assumptions and constraints that could affect measurements of similarity among cases. Its meaning always depends on the underlying context of a particular application and does not convey a fixed characteristic that applies to any comparative context. There are broadly two major approaches. The first is based on the computation of distance between cases, where the most similar case is determined by evaluation of a similarity measure (i.e., metric). The second approach is related more to the representation/indexing structures of the cases. The indexing structure can be traversed to search for a similar case. Some typical similarity measures used in CBR systems are weighted Euclidean distances, Hamming and Levenshtein distances, cosine coefficient, ratio of common and different factors, and the k-nearest neighbor principle.

For adaptation knowledge, the traditional approach is by interviewing domain experts and coding the task-specific adaptation knowledge manually into the CBR system. This knowledge may be represented as a decision table, semantic tree, or IF–THEN rules. Alternatively, the adaptation knowledge can be learned from the cases using machine learning techniques. Through learning we generate specialized heuristics that relate the differences in the input specification (i.e., problem attributes) to the differences in the output specifications (i.e., solution attributes). These heuristics can be used to determine the amount of adaptation that is suitable. After the adaptation, it is desirable to check if the solution really works for that particular problem on hand. At this point, there is also a need to consider what action is to be taken if this check determines that the solution proposed is unlikely to be successful. At this stage, the CBR system will now enter a learning phase. Learning may occur in a number of ways. The addition of a new problem, its solution, and the outcome to the case base is a common method. Alternatively, learning may be needed to modify the criteria for case retrieval or indexing, or even the adaptation process itself.

48.1.3 Organization of the Chapter

The central focus of CBR system is the concept of 'case knowledge' (i.e., information granules) and the techniques of how to extract 'case knowledge.' This 'case knowledge' can be considered as something in between from specific cases to generalize abstract model. For example, (i) after removal of irrelevant features and selection of representative cases from the original data, this collection can be regarded as the 'case knowledge'; (ii) after generalizing a number of cases into a prototypical case, and a selection of a set of prototypical cases can also be regarded as the 'case knowledge.'

In this chapter, we will illustrate how this can be done. Before that, we would like to summarize the case knowledge extraction processes in Figure 48.1.

For removing redundancy and noises, reducing sizes, and preserving the competence, we consider the following four tasks: (1) feature reduction (FR), (2) learning similarity measures, (3) case selection (CS) and case generation (CG), and (4) competence model development.

The first two tasks aim to reduce the redundancy contained in the features and thus avoid the negative effect of the non-informative features. Based on the reduced feature set and learned similarity measures, we can remove those redundant and noisy cases, thus resulting a better data clustering performance. Since the dimensionality of the case bases has been reduced, the problem-solving efficiency is also improved.

The next two tasks, i.e., CS and CG, are performed to obtain a smaller set of cases. While CS is to identify and remove redundant and noisy cases, CG generates new prototypical cases by merging multiple cases into one case and extracting the central cases from different clusters.

Finally, in order to assess the completeness of the case bases, we develop competence model to describe and predict the range of problems that the CBR systems can successfully solve. Due to the page limits in this chapter, we will not explain the details of the competence model development.

48.2 Fast Rough-Set-Based Feature Reduction

FR is the first task of case knowledge extraction; its purpose is to remove the non-informative features and facilitate the task of case selection. We presents a novel and fast approach for FR, which is developed

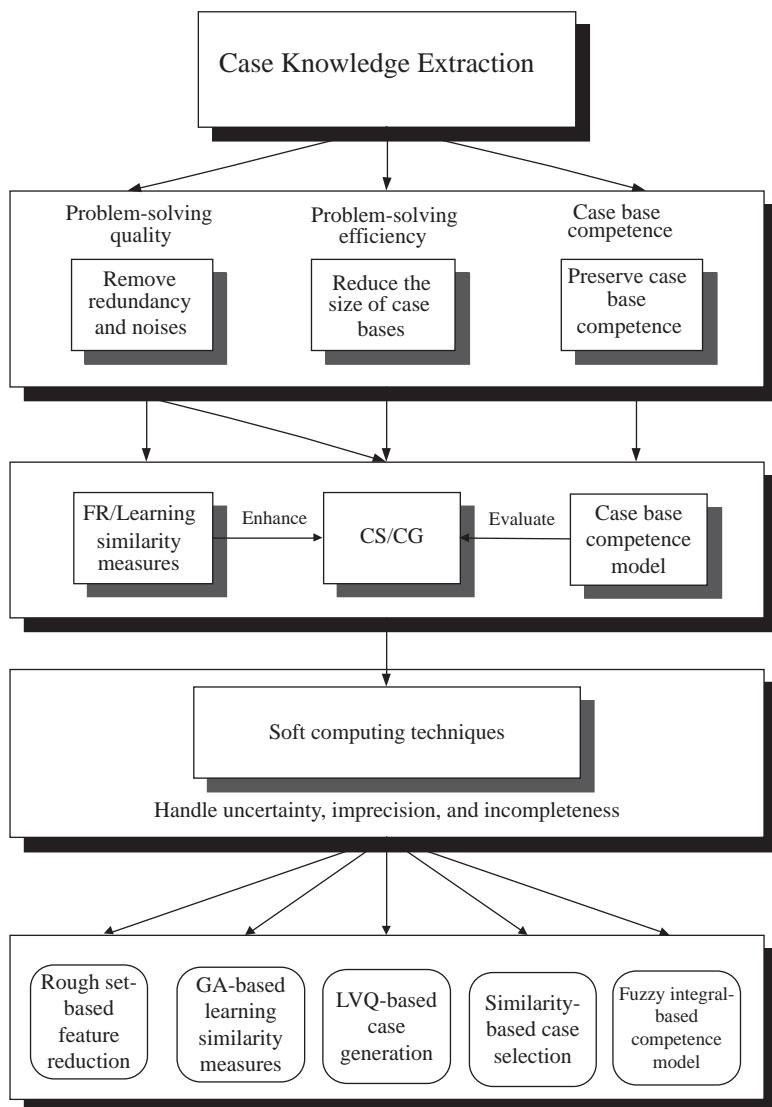


Figure 48.1 The methodology of case knowledge extraction

based on the relative attribute dependency among features to compute the approximate reduct instead of crisp reduct. The approximate reduct is considered as a generalization of the crisp reduct, which can be found quickly. Some fundamental concepts, such as dispensable/indispensable attributes, reduct and core, are also modified.

48.2.1 Basic Concepts

An information system and the corresponding decision table are denoted by $IS = (U, A, f)$ and $DT = (U, A \cup \{d\}, f)$, respectively. The conditional attribute set $C = A$, and the decision attribute set $D = \{d\}$. The concept of reduct is defined on the basis of positive region.

Definition 1. (Reduct). A subattribute set $B \subseteq A$ is called a reduct of A if it is a set of indispensable attributes in the information system IS and $IND(B) = IND(A)$.

Traditionally, the reducts are obtained through the computation of discernibility matrix. The discernibility matrix of an IS completely depicts the identification capability of the system and all reducts of the system are therefore hidden in some discernibility function induced by the discernibility matrix [18]. Based on these definitions, it requires a considerable effort for the discernibility function-based reduct generation methods to compute the discernibility matrix.

First, the discernibility matrix DM of the decision table DT is generated in step 1 (see below Algorithm: Generate Reduct Based on Discernibility Matrix).

Since DM is symmetric and the elements $dm_{ii} = \emptyset$ for $i = 1, 2, \dots, n$, it can be represented only by the elements in the lower or upper triangle of the matrix.

Second, one of the reduct of the feature set A , denoted by $REDU$, is generated in step 2. It can be further divided into several substeps:

1. The $CORE$ of DT is obtained as such a set of dm_{ij} that each dm_{ij} contains a single attribute which can identify at least two cases; that is, $CORE = \{dm_{ij} \in DM \mid \text{card}(dm_{ij}) = 1\}$.
2. $REDU$ is initialized to $CORE$.
3. One of the attributes in A is added iteratively to $REDU$ until the intersection of $REDU$ and each dm_{ij} ($1 \leq i, j \leq N$) is not empty.

The substep (1) is based on a proposition of the concept of core [18]:

$$CORE(A) = \{a \in A : dm_{ij} = \{a\} \text{ for some } i, j\}.$$

Proof. Let $B = \{a \in A : dm_{ij} = \{a\} \text{ for some } i, j\}$. It needs to show that $CORE(A) = B$. The proof is divided into two parts:

(\subseteq) Let $a \in CORE(A)$. Then $IND(A) \subseteq IND(A - \{a\})$, so there exist x_i and x_j which are indiscernible with respect to $A - \{a\}$ but discernible by a . Hence $dm_{ij} = \{a\}$.

(\supseteq) If $a \in B$, then for some i and j we have $dm_{ij} = \{a\}$. Hence, a is indispensable in A . ■

The reduct computation in substep (3) can be explained as follows. Since any attribute in dm_{ij} can distinguish cases i and j , the attributes in $REDU$ can also discern the two cases if the intersection of $REDU$ and dm_{ij} is not empty. When the iterations of adding attributes to $REDU$ stop, the discernibility power of the set of elements in $REDU$ (i.e., a subset of the original attributes) is the same as that of the original set of conditional attributes A .

Here we should mention that $REDU$ is not necessarily the minimal set of attributes that preserves the identification capability of the original information system.

Algorithm: Generate Reduct Based on Discernibility Matrix

Step 1. Create the discernibility matrix $DM = [dm_{ij}]$, $i, j = 1, 2, \dots, n$.

Step 2. Generate one reduct.

Let A denote the set of original attributes.

$CORE = \{dm_{ij} \in DM \mid \text{card}(dm_{ij}) = 1\}$;

$REDU = CORE$;

$A = A - REDU$;

While ($A \neq \emptyset$) do

If ($REDU \cap dm_{ij} \in DM \neq \emptyset$ for every i and j)}, stop;

Else

{Randomly select one attribute $a \in A$

```

Add  $a$  to  $REDU$ :  $REDU = REDU \cup \{a\}$ ;
 $A = A - \{a\}$ ;
}

```

In this reduct generation algorithm, step 1 requires $O(n^2)$ computations to create the discernibility matrix. This is because that DM has n^2 elements of the form dm_{ij} and the number of steps for computing of any dm_{ij} is bounded by a constant. In step 2, $REDU$ has m elements at maximum, and DM has n^2 elements. Therefore, step 2 needs $O(n^2 \times m)$ computations to obtain the intersection of $REDU$ and each dm_{ij} . Thus, the complexity of this algorithm is $O(n^2 \times m)$, which is rather high with large number of cases and attributes.

To address the problem of computational complexity, Han *et al.* [19] have developed a reduct computation approach based on the concept of relative attribute dependency. Given a subset of condition attributes, B , the relative attribute dependency is a ratio between the number of distinct rows in the decision table corresponding to B only and the number of distinct rows in the decision table corresponding to B together with the decision attributes, i.e., $B \cup \{d\}$. The larger the relative attribute dependency value (i.e., close to 1), the more useful is the subset of condition attributes B in discriminating the decision attribute values. If this value equals to 1, each distinct row in the decision table corresponding to B maps to a distinct decision attribute value. Some further concepts [19] are defined as

Definition 2. (Projection). Let $P \subseteq A \cup D$, where $D = \{d\}$. The projection of U on P , denoted by $\Pi_P(U)$, is a subtable of U and is constructed as follows:

1. Remove attributes $A \cup D - P$.
2. Merge all indiscernible rows.

Definition 3. (Consistent Decision Table). A decision table DT or U is consistent when $\forall x, y \in U$, if $f_D(x) \neq f_D(y)$, then $\exists a \in A$ such that $f_a(x) \neq f_a(y)$.

Definition 4. (Relative Dependency Degree). Let $B \subseteq A$, A be the set of conditional attributes. D is the set of decision attributes. The relative dependency degree of B w.r.t. D is defined as δ_B^D , $\delta_B^D = \frac{|\Pi_B(U)|}{|\Pi_{B \cup D}(U)|}$, where $|\Pi_X(U)|$ is the number of equivalence classes in $U / IND(X)$.

The relative dependency degree δ_B^D implies how well subset B discerns the objects in U relative to the original attribute set A . It can be computed by counting the number of equivalence classes induced by B and $B \cup D$, i.e., the distinct rows in the projections of U on B and $B \cup D$.

Based on the definition of the relative dependency degree, we define the dispensable and indispensable attributes as follows:

Definition 5. (Dispensable and Indispensable Attributes). An attribute $a \in A$ is said to be dispensable in A w.r.t. D , if $\delta_{A-\{a\}}^D = \delta_A^D$; otherwise, a is indispensable in A w.r.t. D .

According to Definitions 3 and 4, we can obtain Lemma 1.

Lemma 1. $\forall B \subseteq A$, $\Pi_{B \cup D}(U)$ is consistent if and only if $|\Pi_B(U)| = |\Pi_{B \cup D}(U)|$.

Proof. We need to show (1) if $\Pi_{B \cup D}(U)$ is consistent, then $|\Pi_B(U)| = |\Pi_{B \cup D}(U)|$; and (2) if $|\Pi_B(U)| = |\Pi_{B \cup D}(U)|$, then $\Pi_{B \cup D}(U)$ is consistent. ■

1. We assume that $\Pi_{B \cup D}(U)$ is consistent. According to the definition of consistent decision table (Definition 3), if $xIND(B)y$, ($x \in U, y \in U$), then $xIND(B \cup D)y$. Therefore, the number of equivalence classes of $\Pi_{B \cup D}(U)$ and $\Pi_B(U)$ is equal; i.e., $|\Pi_B(U)| = |\Pi_{B \cup D}(U)|$.
2. This part of proof is by contradiction. Suppose $|\Pi_B(U)| = |\Pi_{B \cup D}(U)|$ and $\Pi_{B \cup D}(U)$ is inconsistent. Then there exists at least two objects x and y , having the same condition attribute values but different decision attribute value. That is, x and y belong to one equivalence class with respect to B , and

belong to two different equivalence classes with respect to $B \cup D$. We have $|\Pi_B(U)| < |\Pi_{B \cup D}(U)|$. This contradicts to $|\Pi_B(U)| = |\Pi_{B \cup D}(U)|$. Therefore, $\Pi_{B \cup D}(U)$ must be consistent when $|\Pi_B(U)| = |\Pi_{B \cup D}(U)|$.

It is easy to prove that, if U is consistent, then $\delta_A^D = \frac{|\Pi_A(U)|}{|\Pi_{A \cup D}(U)|} = 1$; i.e., $|\Pi_A(U)| = |\Pi_{A \cup D}(U)|$.

Lemma 2. If U is consistent, then $\forall B \subset A$, $POS_B(D) = POS_A(D)$ if and only if $|\Pi_B(U)| = |\Pi_{B \cup D}(U)|$.

Proof [19]. U is consistent indicates that $POS_A(D) = U$ and that $|\Pi_B(U)| = |\Pi_{B \cup D}(U)|$ means that $\Pi_{B \cup D}(U)$ is consistent according to Lemma 1. It can easily be inferred that $\Pi_{B \cup D}(U)$ is consistent if and only if $POS_B(D) = U$.

Based on Definitions 3 and 5 and Lemmas 1 and 2, Theorem 1 can be induced.

Theorem 1. If U is consistent, $B \subseteq A$ is a reduct of A w.r.t. D , if and only if $\delta_B^D = \delta_A^D = 1$ and for $\forall Q \subset B$, $\delta_Q^D \neq \delta_A^D$.

Proof [19]. According to Definition 4, $\delta_B^D = \delta_A^D = 1$ means that $|\Pi_B(U)| = |\Pi_{B \cup D}(U)|$, if and only if, by Lemma 2, $POS_B(D) = POS_A(D)$. Similarly, for $\forall Q \subset B$, $\delta_Q^D \neq \delta_A^D$ if and only if $POS_Q(D) \neq POS_A(D)$.

In order to compute the reduct quickly, we use Definitions 4 and 5 (relative dependency degree, dispensable and indispensable attributes) and Theorem 1. Theorem 1 gives the necessary and sufficient conditions for reduct computation and implies that the reduct can be generated by only counting the distinct rows in some projections.

48.2.2 Fast Rough-Set-Based FR Algorithms

In Theorem 1, U is always assumed to be consistent, which is not necessarily true in real-life applications. In this section, we relax this condition to find approximate reducts rather than the exact reducts. The use of a relative dependency degree in reduct computation is extended to inconsistent information systems. Some new concepts, such as the β -dispensable attribute, β -indispensable attribute, β -reduct (i.e., approximate reduct), and β -core are introduced to modify the traditional concepts in rough set theory. The parameter β is used as the consistency measurement to evaluate the goodness of the subset of attributes currently under consideration. It can also determine the number of attributes which will be selected in the generated approximate reduct. These are explained as follows.

Definition 6. (β -Dispensable Attribute and β -Indispensable Attribute). If $a \in A$ is an attribute that satisfies $\delta_{A-\{a\}}^D \geq \beta \cdot \delta_A^D$, a is called a β -dispensable attribute in A . Otherwise, a is called a β -indispensable attribute.

The parameter β , $\beta \in [0, 1]$, is called the consistency measurement.

Definition 7. (β -Reduct/Approximate Reduct and β -Core). B is called a β -reduct or approximate reduct of conditional attribute set A if B is the minimal subset of A such that $\delta_B^D \geq \beta \cdot \delta_A^D$. The β -core of A is the set of β -indispensable attributes.

The relationship between β -reduct and β -core is similar to the relationship between the traditional reduct and core, which is described in Theorem 2.

Theorem 2. β -core can be computed as the intersection of all approximate reducts; i.e., β -core = $\cap_i \text{reduct}_i$, where reduct_i is the i th approximate reduct.

Proof. The proof is divided into two parts.

1. For every attribute $a \in \beta\text{-core}$, a is a β -indispensable attribute; i.e., $\delta_{A-\{a\}}^D < \beta \cdot \delta_A^D$. According to Definition 7, an approximate reduct implies that $a \in \cap_i \text{reduct}_i$. This can be proved using the method of contradiction as follows:
If $\exists i, a \notin \text{reduct}_i$, then $\text{reduct}_i \subseteq A - \{a\}$ and $\delta_{\text{reduct}_i}^D < \delta_{A-\{a\}}^D < \beta \cdot \delta_A^D$. This result contradicts the assumption that reduct_i is an approximate reduct. Therefore, $a \in \cap_i \text{reduct}_i$, and then $A \subseteq B$ holds.
2. Let an attribute $a \in \cap_i \text{reduct}_i$. If we assume $a \notin \beta\text{-core}$, that is, a is a dispensable attribute, then $\exists i$, such that $a \notin \text{reduct}_i$. This is not possible since $a \in \cap_i \text{reduct}_i$. Therefore, $a \in \beta\text{-core}$.

This completes the proof. ■

The consistency measurement β represents how consistent the subdecision table (with respect to the considered subset of attributes) is relative to the original decision table (with respect to the original attribute set). It also reflects the relationship of the approximate reduct and the exact reduct. The larger the value of β , the more similar is the approximate reduct to the exact reduct computed using the traditional discernibility function-based methods. If $\beta = 1$ (i.e., attains its maximum), the two reducts are equal (according to Theorem 1). The reduct computation is implemented by counting the distinct rows in the subdecision tables of some subattribute sets. β controls the end condition of the algorithm and therefore controls the size of reduced feature set. Based on Definitions 6 and 7, the first rough-set-based FR algorithm in our developed approach is given in Figure 48.2.

In some domains, the order for selecting attributes in the reduct must be considered carefully. For example, when dealing with text documents, there are hundreds or thousands of keywords which are all regarded as attributes. If the order is randomly selected or if one simply makes use of the order in which keywords appear in a text document, the most informative attributes may not be selected initially during reduct computation. Therefore, the end condition $\delta_R^D > \beta$ in Algorithm 1 cannot be satisfied quickly. It should also be borne in mind that the final attribute set may consist of many non-informative features. This issue is addressed by computing the significance value of each attribute. These significance values are used to guide the attribute selection sequence. Details are given in Algorithm 2 (see Figure 48.3).

Feature Reduction Algorithm 1

```

Input:  $U$  – the entire set of objects;
 $A$  – the entire condition attribute set;
 $D$  – the decision attribute set.
Output:  $R$  – the approximate reduct of  $A$ 

Step 1 Initialize  $R = \emptyset$  (empty set).

Step 2 Compute the approximate reduct.
  While ( $A$  is not empty)
    Compute  $\delta_R^D$ ;
    If ( $\delta_R^D > \beta$ )
      Return  $R$  and stop;
    Otherwise
       $R = R \cup \{q\}$ ;
       $A = A - R$ ;
  Step 3 Output  $R$ .

```

Figure 48.2 Feature reduction algorithm 1

Feature Reduction Algorithm 2

The inputs and output are the same as that in algorithm 1.

Step 1 Initialize $R = \emptyset$;

Step 2 For each $a \in A$

- Compute the significance of a ;*
- Add the most significant one, q , to R :*
- $R = R \cup \{q\}$;*
- $A = A - \{q\}$;*

Step 3 For current R

- Compute the relative dependency degree δ_R^D ;*

Step 4 While (A is not empty)

- If $\delta_R^D > \beta$, return R and stop;*
- Otherwise, go to step 2 and then step 3.*

Step 5 Output R

Figure 48.3 Feature reduction algorithm 2

Notice that the significance of an attribute can be evaluated in many ways using different evaluation criteria such as information gain (IG), frequency of occurrence (often used in text documents), and dependency factors (in rough-set-based methods). The computation complexities of the feature reduction Algorithms 1 and 2 are $O(n \times m)$, where m is the number of attributes in $A \cup D$ and n is the number of objects in U . In the FR algorithms, we consider m subsets of attributes by adding one attribute in each iteration until $\delta_R^D > \beta$, and n computations are required in each iteration to calculate the different rows.

48.2.3 Experimental Results

In this section, we test our proposed FR algorithms mainly on classification problems and provide their comparisons with KPCA. The experiments used four real-life data sets: (1) House-votes-84 database [20]; (2) Text document sets (Texts 1-8), it is composed of eight text document sets randomly sampled from Reuters21578 [7]; (3) Mushroom Database [20]; and (4) Multiple Features [20]. The experimental results demonstrate not only a reduced storage requirement but also an improvement in the classification accuracy with fewer features in the generated reduct. Notice that Storage = |Reduced feature set| / |Original feature set|, where $|.|$ is the cardinality of set $(.)$. The accuracy is the classification accuracy when using the reduced feature set.

1. House-votes-84.

This data set is tested using four splits: randomly selecting 20, 30, 40, and 50% of the original data, as the testing data; the corresponding left data are used as the training data. The four splits are denoted as Split 1–4. Table 48.1 shows the reduced storage requirement and the classification accuracy with different β values.

Table 48.1 provides three observations: (i) the features could not be reduced with $\beta = 1$; (ii) the classification accuracy is improved after the rough-set-based feature reduction with almost all of the used β values; (iii) the accuracy attains most of its maximums for the four splits when $\beta = 0.95$.

In Table 48.2, P0 represents the original accuracy with the whole data set, while P(FR) denotes the accuracy with the reduced feature set after applying the rough-set-based feature reduction algorithm 1.

Table 48.1 Storage and accuracy with different β values on house-votes-84

β	Split 1		Split 2		Split 3		Split 4	
	Storage (%)	Accuracy (%)	Storage (%)	Accuracy (%)	Storage (%)	Accuracy (%)	Storage (%)	Accuracy (%)
1.00	100.00	93.10	100.00	92.31	100.00	93.68	100.00	94.01
0.90	43.75	96.55	43.75	94.62	43.75	95.98	43.75	95.85
0.95	56.25	97.70	50.00	94.62	50.00	95.98	56.25	96.31
0.96	63.50	94.25	66.25	95.38	62.50	94.83	62.50	94.47
0.97	68.75	94.25	62.50	94.62	62.50	94.83	68.75	92.63

Table 48.2 Storage and accuracy using $\beta = 0.95$

Split	P0 (%)	P(FR) (%)	Storage
1	93.10	97.70	56.25
2	92.31	94.62	50.00
3	93.68	95.98	50.00
4	94.01	96.31	56.25
Avg.	93.28	96.15	53.13

2. Text data sets.

The fast rough-set-based FR algorithm 2 was applied to the text data sets. The most distinct characteristic of the text domain is its high dimensionality. We randomly select 80% documents in each text data set as the training data and the remaining 20% is used as the testing data.

Initially, each word (term) occurred in the text data is considered as a feature, and therefore there are often hundreds or thousands of feature terms in a text data set. Before the FR algorithm is performed, we preprocess the term feature set to filter the stop words or very low-frequent words. Stop words are extremely common words which appear in almost every document, such as ‘a,’ ‘the,’ and ‘with.’ These words are often considered to contribute little useful information in classifying the text documents. On the other hand, low-frequency words – words which occur just once or twice – are filtered. The words which remain are considered to be the original feature terms.

To facilitate the rough-set based feature term reduction, each text document is represented using a term vector with respect to the acquired original feature terms. Assume that there are m terms in the set of original feature terms. A given document DOC can be described by an m -dimensional term vector $[t_1, t_2, \dots, t_m]$, where t_k is a Boolean variable which is given by

$$t_k = \begin{cases} 1 & \text{if } DOC \text{ contain term } k \\ 0 & \text{if } DOC \text{ does not contain term } k \end{cases} \quad k = 1, 2, \dots, m. \quad (1)$$

Each document is represented by an m -dimension vector. An example of a document vector is

$$D = [t_1, t_2, \dots, t_m] \quad t_k \in [0, 1], k = 1, 2, \dots, m, \quad (2)$$

where t_k is the normalized weight of feature term k in document DOC . t_k is computed by two steps: weight computation and weight normalization.

Step 1: Weight computation. Compute the weight of each feature term in each document using term frequency-inverted document frequency ($tf-idf$).

$$w_k = -\log(N_k/N)f_k, \quad k = 1, 2, \dots, m,$$

where

w_k is the weight of term k ;

N_k is the number of documents containing term k ;

N is the total number of documents;

f_k is the frequency of term k .

Note that w_j is the weight of the j th term in the whole set of text documents. Here, in order to reduce the computational load, the term weight for each term in each document is not computed.

Step 2: Weight Normalization. Let w_{\max} denote the maximal weight. w_k is normalized to be

$$w_k = w_k/w_{\max}.$$

That is, for each k in equation (2), $t_k = w_k$.

Table 48.3 Reduced storage and improved accuracy when applying $\beta = 1$ to text data

Text data set	P0 (%)	P(FR) (%)	Storage (%)
Text1	62.50	75.00	12.50
Text2	62.50	62.50	9.05
Text3	33.33	33.33	28.48
Text4	37.93	41.38	10.51
Text5	56.25	75.00	9.59
Text6	77.78	77.78	31.53
Text7	51.19	50.00	7.81
Text8	72.79	69.39	3.37
<i>Avg.</i>	56.78	60.55	14.11

In the FR algorithm 2, the significance of each feature $t_k(k = 1, 2, \dots, m)$ is evaluated by its term frequency-inverted document frequency, i.e., w_k , which is positively proportional to the frequency of occurrence of this feature and inverse proportional to the number of documents which contain this term.

The experimental results in Table 48.3 shows that the storage requirements for all the eight data sets fall significantly, and the accuracy using reduced feature set is preserved for Text2, Text3, and Text6 and even improves for Text1 and Texts4-5. For Texts7-8, the accuracy decreases a little due to the reduction of features. Since the accuracy attains its maximum when $\beta = 1$, here β is set to be 1.

3. Mushroom data.

We randomly select 80% data as the training data and the left 20% as the testing data. It is shown that there are five features in the generated reduct. Therefore, the storage requirement with respect to the feature set is 22.73% of the original feature set. Here, $\beta = 1$. The classification accuracy is not affected after feature reduction. For this data set, $P0 = P(FR) = 1$.

Summary: After applying the fast rough-set-based FR method to House-votes data and texts 1–8, the feature set is substantially reduced and the classification accuracy is preserved or even improved. Tables 48.1–48.3 show that the improvement in classification accuracy is 3.06% for House-votes-84 and 3.77% for the text data sets. The size of the feature set decreases from the original 100% to 53.13% for house-votes-84, 14.11% for text data sets, and 22.70% for mushroom data.

48.3 Learning Similarity Measure of Nominal Features

This section tackles the second task for case knowledge granules extraction, i.e., learning similarity measures of nominal features. The purpose of this task is to discover the relationship among different feature values to emphasize the importance of critical features. Nominal feature is one type of symbolic features; its feature values are completely unordered. The most often used similarity metrics for symbolic features is the Hamming metric and its variances which always assume that, if two nominal feature's values are equal, the similarity is defined as 1; otherwise, the similarity is defined as zero. This similarity computation is coarse grained and may affect the quality of the retrieved cases and also the problem-solving accuracy. Here, we extend the similarity values from $\{0, 1\}$ to $[0, 1]$ using a GA-based supervised method for the learning of similarities of nominal feature values. Here we assume that there are only limited feature values in the domain of each nominal feature. Theoretically, for a given nominal feature, the similarity of each pair of feature values is required to be computed to determine the similarity measure of this feature. That is, if there are n elements in the domain of a feature, $n \cdot (n - 1)/2$ similarity values need to be learned, which may require substantial computational effort. In practice, it is not necessary to determine so many similarity values. In the given classification problem, if two different nominal feature values certainly lead to different class labels, the similarity

between these two nominal values is assumed as zero. The GA-based method is then used to learn the similarity values of other nominal feature values. The learned similarity values are expected to improve the classification accuracy and can be used to analyze the importance of each feature in the given CBR classifier.

48.3.1 Using GA to Learn Similarity Measure for Nominal Features

Let there be a case base consisting of N cases e_1, e_2, \dots, e_N , m nominal features f_1, f_2, \dots, f_m . The domain of each feature has limited elements represented by $D_i = \{v_{i1}, v_{i2}, \dots, v_{m,l_i}\}$, $i = 1, 2, \dots, m$; v_{ij} is a nominal value and l_i is the number of different values in the domain of the i th nominal feature. There are $L_i = l_i \cdot (l_i - 1)/2$ similarity values that should be learned at most for the i th feature. In the following, we discuss the encoding rule, fitness function, and the constructed GA algorithm in detail.

Encoding rule:

Each chromosome is encoded as a string consisting of m parts corresponding to the m features. A chromosome c takes the form shown in Figure 48.4. For the i th part, there are L_i genes represented by L_i decimals: $s_{ip} \in [0, 1]$, $(1 \leq p \leq L_i)$, representing the similarity measure for the i th feature. The initial values of s_{ij} ($j = 1, 2, \dots, L_i$) are randomly generated for $i = 1, 2, \dots, m$.

Fitness function:

In the GA-based learning process, the fitness function of a chromosome c is the corresponding classification accuracy using the similarity values indicated in c . Based on case retrieval, the class label of an unseen case can be determined by the majority of its k nearest neighbors. The classification accuracy is the ratio of the number of correctly classified cases, N_{Corr}^c , over the whole number of unseen cases, N_{Total}^c . The fitness function is then defined as $\text{fitness}(c) = N_{Corr}^c / N_{Total}^c$.

The GA algorithm:

- Initialize the population of the chromosomes. A population set is represented by $\{c_1, c_2, \dots, c_P\}$, where P is the size of the population. Each chromosome is encoded as in Figure 48.4. Each gene is a randomly initialized to be a decimal in $[0, 1]$, representing the similarity value between two nominal values in the domain of each nominal feature.
- Selection and crossover. Here the selection probability is set as 1 and the whole set of population is considered to be the mating pool. These settings let the modal to be closer to a random search. In each generation, two chromosomes are randomly selected to perform crossover. The cutting point for crossover is randomly generated and the genes in the two chromosomes that lie behind the cutting point are exchanged to produce an offspring.
- Mutation. Let the mutation probability be p_{muta} . Randomly select one gene g (with value v_g) in the newly generated offspring string and convert the value v_g to $(1 - v_g)$. If v_g represents the degree of how similar of two feature values, then $(1 - v_g)$ represents their degree of dissimilarity.
- End condition. Repeat (a)–(c) until the number of generations attains a predefined threshold.

s_{11}	s_{12}	...	s_{1L_1}	s_{21}	s_{22}	...	s_{2L_2}	...	s_{m1}	s_{m2}	...	s_{mL_m}
0.25	0.10	...	0.83	0.62	0.58	...	0.35	...	0.74	0.40	...	0.56

Figure 48.4 A chromosome c

Here we provide some discussions about parameter control in GA. Crossover probability is how often crossover will be performed. If there is no crossover, offspring are exact copies of parents. If crossover probability is 100%, then all offspring are made by crossover. Crossover is made in hope that new chromosomes will contain good parts of old chromosomes and therefore the new chromosomes will be better. The crossover operator can be divided into two types: 1-point and multipoint crossover. 1-point is used in the traditional GA, where two mating chromosomes are each cut once at corresponding points, and the segments following the cuts are exchanged. In a 2-point or multipoint crossover, chromosomes are seen as loops formed by joining the ends together, rather than as linear strings. Researchers now agree that 2-point crossover produces better results than 1-point crossover [21]. However, if a strict interpretation of the schema theorem is imposed then operators which use many crossover points should be avoided because they can cause extreme disruption to schema [22].

Mutation probability is how often parts of chromosome will be mutated. If there is no mutation, offspring are generated immediately after crossover (or directly copied) without any change. If mutation probability is 100%, whole chromosome is changed. Mutation generally prevents the GA from falling into local extremes. Mutation should not occur very often, because then GA will in fact change to random search. Recommendations are often results of empiric studies of GAs that were often performed on binary encoding only. Crossover rate should be high generally, about 80–95%. On the other side, mutation rate should be very low. Best rates seem to be about 0.5–1%.

48.3.2 Simulation Results and Analysis

The Balloons database from the UCI repository [20] is used in the simulations to show the effectiveness of learning similarity measure using the GA-based learning approach. This data set consists of 16 cases and 4 nominal features (three conditional features and one class label). There are two nominal values in the domain of each conditional feature. Some example cases are shown in Table 48.4. It is found that there are four similarity values which need to be learned: $\text{sim}(\text{Yellow}, \text{Purple})$ (Color), $\text{sim}(\text{Small}, \text{Large})$ (Size), $\text{sim}(\text{Stretch}, \text{Dip})$ (Act), and $\text{sim}(\text{Adult}, \text{Child})$ (Age).

Six cases are firstly selected as the training data and the remaining 10 cases are used as testing cases. The original classification accuracy based on the majority voting principle is 0.70. In the learning process of the GA algorithm, the mutation probability is also set as 0.05.

Table 48.5 shows the learned similarity values for the four nominal features. With these similarity values, the accuracy increases from the original 0.70 to 0.97. The distances of similarity values to {0, 1} for ‘act’ and ‘age’ are the smallest compared with that of other features. Therefore, the features ‘act’ and ‘age’ are the most critical features to make the classification decisions. In fact, only using these two features, all the testing cases can be correctly classified based on the majority voting principle. In contrast, with the other two features ‘color’ and ‘size,’ five out of ten cases are classified to the wrong classes.

48.4 Case Selection Methods for Case Knowledge Granules Extraction

The task of CS is discussed in the context of CBR classifiers, which can be defined as CBR systems that are built for the classification problem – to determine whether or not an object is a member of a class or which of several classes it may belong to. To build a CBR classifier, the cases stored in the case base

Table 48.4 Example cases in Balloons database

ID	Color	Size	Act	Age	Inflated
1	Yellow	Small	Stretch	Adult	T
2	Purple	Large	Dip	Child	F

Table 48.5 Learned similarity values on balloons database (original accuracy = 0.70)

Number of Generations	sim(Yellow, Purple)	sim(Small, Large)	sim(Stretch, Dip)	sim(Adult, Child)	Accuracy
100	0.55	0.43	0.69	0.28	1.0
500	0.56	0.86	0.83	0.38	0.9
1,000	0.57	0.55	0.79	0.24	0.9
5,000	0.63	0.51	0.29	0.03	1.0
10,000	0.65	0.31	0.84	0.25	1.0
20,000	0.70	0.33	0.79	0.23	1.0
Avg.	0.61	0.50	0.71	0.24	0.97
$d^*, \{0, 1\}$	0.39	0.36	0.23	0.23	—

Note: * denotes the features, color, size, act, and age, respectively.

are used as training data and the unseen cases are used as testing data. Through combining the FR and CS processes, we present a novel and fast approach to extract case knowledge granules for building both efficient and competent CBR classifiers.

Like FR, CS is economical. The main objective of the CS process developed in this research is to extract case knowledge through identifying and removing redundant and noisy cases. In the context of CBR, a redundant case can be defined as follows. If two cases are the same (i.e., case duplication) or if one case subsumes another case, one of the cases duplicated or subsumed cases are considered to be redundant. They can be removed from the case base without affecting the overall problem-solving ability of the CBR system. The meaning of subsumption is as follows: Given two cases e_p and e_q , when case e_p subsumes case e_q , case e_p can be used to solve more problems than e_q . In this case, e_q is said to be redundant. On the other hand, the definition of noisy cases is very much dependent on how we interpret the data distribution regions, and their association with the class labels. According to Brighton and Mellish [23], there are two broad categories of class structures: the classes are defined by (1) homogenous regions' or (2) heterogeneous regions. In this research, we consider only the first category of data distribution. Based on the assumption that similar problems should have similar solutions, we define noisy cases as those that are very similar in their problem specifications yet propose different (or conflicting) solutions.

CS schemes are traditionally based on the k -NN principle, e.g., the condensed nearest neighbour rule (CNN) [24] and the Wilson editing method [25]. There are several variations of the CNN and Wilson editing method [26–28]. Based on the assumption that similar problems should have similar solutions, these methods examine the k -nearest neighbours of each case and then identify and remove noisy cases. This group of methods are referred to as k -NN based CS methods.

Some CS strategies are derived from the area of case base maintenance (CBM), which includes policies of revising the organization and content of the case bases to facilitate the future reasoning of CBR systems. The concepts of case coverage and reachability [2–6] are used to reduce redundant cases and thus build the case knowledge bases. As mentioned earlier, *coverage* of a case is the set of target problems (i.e., cases) that this case can be used to solve. The *reachability* of a target problem (i.e., a case) is the set of all cases that can be used to solve the target problem. The larger the coverage and the smaller the reachability of a case, the more important this case in the CBR system. Thus, these two concepts can be used to identify redundant cases through examining the problem-solving ability of each case. Some such algorithms are developed in [2–6, 29, 30]. This research constructs and compares different case selection approaches based on the similarity measure and the concepts of case coverage and reachability, which are closely related to the k -NN-based methods.

Case generation (CG) is an alternative approach of CS for reducing the size of the case base. New cases (or called prototypes) can be generated instead of selecting a subset of cases from the original case base. New cases, thus generated, have lower dimension than that in the original case base, for example, the fuzzy rough method in [11, 12] generated cases of variable dimensions of lower size. On the other

hand, the support vectors produced by SVM [31] or SVM ensemble [32, 33] can also be considered as cases selected as a subset of the original case base.

48.4.1 Case Selection Approach

In this section, we present four CS algorithms that are based on the similarity measure but that use of the case similarity in different ways. Algorithm 1 first selects cases having a large coverage and then, if the two cases have a similar coverage, selects the one with the smaller reachability set. CS algorithm 2 directly selects cases according to measurements of case similarity. The CS algorithms 3–4 are formed by incorporating the k -NN principle into CS algorithm 1 and CS algorithm 2, respectively.

Each of the four CS approaches has its own rationale. For CS algorithm 1, the similarity concept is used to compute a case's coverage and reachability values, which can be interpreted as a measurement of its significance with respect to all other cases. A case is considered to be important if it ‘covers’ many similar cases (with a similarity value greater than a threshold α) all belonging to the same class. Here α is the similarity threshold between a particular case and its nearest boundary case. Since the cluster centers (cases) often have large coverage sets and the boundary cases have small coverage sets, this CS algorithm tends to select the cluster centers and remove the boundary cases.

CS algorithm 2 assumes that redundant cases can be found in densely populated clusters in the case base, with the similarity measure being used to describe the local density around a case. The more densely populated the cluster, the more redundant cases should be removed. A threshold can then be set up to determine the number of cases which should be deleted. Assume e_p is a case which has been already selected. A case e_q is considered to be redundant and should be removed if the similarity of e_p and e_q is greater than the given threshold and the classification label of e_p is the same as that of e_q . As they tend to have different class labels from their neighbour cases, boundary cases will not be removed. Therefore, a number of representative interior cases and the boundary cases are preserved. This algorithm is fast, and it is suitable for case bases with high densities.

It is observed that, however, both CS algorithms 1 and 2 are vulnerable to noisy cases. The noisy cases mislead the computations of case coverage and reachability in the first CS algorithm, and they are often recognized as boundary cases which play important role in the second CS algorithm. In order to solve this problem, the k -NN principle is incorporated into the CS algorithms 1 and 2 to first detect and remove noisy cases, thereby forming algorithms 3 and 4. Here, the similarity concept is used to compute the k -nearest neighbors of each case. Based on the assumption that similar cases should have similar solutions, noisy cases are defined as cases having different class labels from the majority voting of their k -nearest neighbors. After the noisy cases are removed, the CS algorithms 1 and 2 are applied to remove the redundant cases. In this way, both noisy and redundant cases can be deleted from the case base.

Before providing a detailed description of the four CS algorithms, we shall define some related concepts. Assume there is a case base CB , the condition attribute set is A , and the decision attribute set is D . *Coverage* of a case is the set of target problems (i.e., cases) that this case can be used to solve, while *reachability* of a target problem (i.e., a case) is the set of all cases that can be used to solve the target problem.

Definition 8. *The coverage set of a case e is defined as*

$$\text{CoverageSet}(e) = \{e' \mid e' \in CB, e' \text{ can be solved by } e\}.$$

Definition 9. *The reachability set of a case e is defined as*

$$\text{ReachabilitySet}(e) = \{e' \mid e' \in CB, e \text{ can be solved by } e'\}.$$

Notice that, in different situations, the meaning that a case can ‘solve’ another case is different. In this chapter, we redefine the two concepts more explicitly as follows.

Definition 10. *Coverage set of a case e is redefined as*

$$\text{Cover}(e) = \{e' \mid e' \in CB, \text{sim}(e, e') > \alpha, d(e) = d(e')\},$$

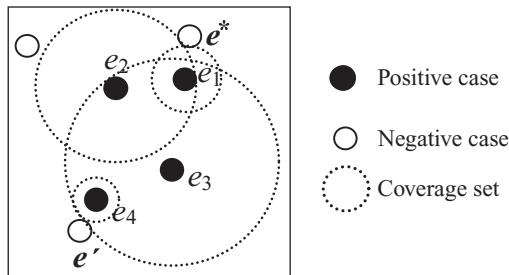


Figure 48.5 The coverage set and reachability set

where α is the similarity computed between case e and its nearest boundary case (the cases which have different class label of e); d is the decision attribute in D .

Here the coverage set of a case e is the set of cases which fall in the disk centred at e with radius α . We assume there is only one decision attribute d . It is straightforward to extend the definition to a situation with multiple decision attributes.

Definition 11. Reachability set of a case can be derived from the Definition 10:

$$\text{Reach}(e) = \{e' \mid e' \in CB, e \text{ can be covered by } e'\}.$$

These definitions are illustrated in Figure 48.5, where e^* is the nearest boundary case of cases e_1 and e_2 and e' is the boundary case of e_3 and e_4 . The dotted circle centered at a case represents the coverage set of this case. According to Definitions 10 and 11, we have

$$\begin{aligned} \text{Cover}(e_1) &= \{e_1\}, \text{Cover}(e_2) = \{e_1, e_2\}, \text{Cover}(e_3) = \{e_1, e_2, e_3, e_4\}, \text{Cover}(e_4) = \{e_4\}; \text{ and} \\ \text{Reach}(e_1) &= \{e_1, e_2, e_3\}, \text{Reach}(e_2) = \{e_2, e_3\}, \text{Reach}(e_3) = \{e_3\}, \text{Reach}(e_4) = \{e_3, e_4\}. \end{aligned}$$

The implication of the concepts of case coverage and reachability is that the larger the coverage set of a case, the more significant the case because it can correctly classify more cases based on the k -NN principle. In contrast, the larger the reachability set of a case, the less important the case in the case base because it can be reached by more existing cases.

In the example shown in Figure 48.5, case e_3 is the most important case due to its largest coverage set and then follows case e_2 . Notice that case e_1 and e_4 have the same size of coverage set but the reachability set of e_4 is smaller, e_4 is considered to be more critical. One focus of this research is the preservation of the competence (the number of cases the case base can cover) of the case bases. We attempt to build an algorithm (see Figure 48.6) for selecting a subset of cases that would preserve the overall competence as compared to the original entire case base.

Since the algorithm involves the computation of coverage set and reachability set for each case in the original case base, the computation complexity of this algorithm is $O(m \times n^2)$, where m is the number of condition attributes in A and n is the number of cases in the case base. This algorithm must make three passes of the case base: the first to compute the similarity; the second to search the boundary case with the largest similarity α for each case; and the third pass is to find the nearest neighbors with a similarity with the current case that is larger than α . Case selection algorithm 2, shown in Figure 48.7, addresses this problem, requiring only one pass of the case base to compute the similarity between each two cases.

Algorithm 2 is totally similarity based. If the similarity between a case e^* and the current case e is larger than a given threshold η and they are with the same class label, e^* will be considered as redundant and eliminated from the case base. This algorithm is suitable to the case bases with high density cases, while the CS algorithm 1 can be used on both sparse and dense cases. Notice that the larger the parameter η , the more cases are selected by this algorithm. The value of η can be determined either by the predefined size of the selected case base or by the required classification accuracy.

Case Selection Algorithm 1

Input: CB – the entire case base;
 A – the entire condition attribute set;
 D – the decision attribute set.
Output: S – the selected subset of cases.

- Step 1 Initialize* $S = \emptyset$ (empty set).
- Step 2 For every case* $e, e \in CB$,
Compute the coverage set and reachability set of e .
- Step 3 Select the case which has the maximum coverage set.*
Ties are broken by selecting the case with smallest reachability set.
- Step 4 The process stop when the selected cases cover the whole original case base* CB .

Figure 48.6 Case selection algorithm 1

Based on the concepts of coverage and reachability, case selection algorithm 1 could remove not only the redundant cases but also the noisy cases due to the small coverage sets of the noisy cases. However, the effectiveness of CS is still degraded by the existence of noisy cases. Cases located near the noisy cases tend to have smaller coverage sets than do other cases. As a result, cases close to noisy cases would be selected less often, which may lead to the loss of important information. Case selection algorithm 2 tends to eliminate redundant cases but was not able to effectively deal with noisy cases. A noisy case e^* may be regarded as a boundary case. Since its class label could not be predicted by the cases which satisfy $\text{sim}(e, e^*) > \eta$, it could not be removed. This would result in the preservation of noisy cases and the selection of an unsatisfactory case base.

To tackle the mentioned problems with case selection algorithm 1 and algorithm 2, the k -NN principle is incorporated to delete both the noisy cases and the redundant cases. Based on the similarity computation between cases, the k -NN principle is firstly used to find out the noisy cases. A case is said to be a noisy case if it cannot correctly classified by the majority of its k -nearest neighbors. Notice that, when the value

Case Selection Algorithm 2

Input: CB – the entire case base;
 A – the entire attribute set;
 D – the decision attribute set.
Output: S – the selected subset of cases.

- Step 1 Initialize resulted subset case base* $S = CB$.
- Step 2 For each case* e in CB ,
Compute the similarity between e *and all the cases in* CB .
- If* $\text{sim}(e, e) > \eta$, and $d(e') = d(e)$,
remove e' *from* S , $S = S - \{e'\}$;
- Step 3 Output* S .

Figure 48.7 Case selection algorithm 2

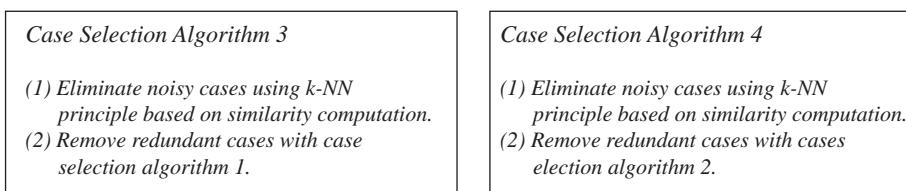


Figure 48.8 Case selection algorithms 3 and 4

k increases, the possibility of a case being a noise decreases and vice versa. In this section, k is equal to the small odd number, 3. After the noisy case removal, the case selection algorithms 1 and 2 are then applied to further eliminate the redundant cases. The CS methods which incorporate the k -NN principle in the CS algorithms 1 and 2 are given as case selection algorithms 3 and 4 (see Figure 48.8).

48.4.2 Combining Feature Reduction and Case Selection

In most existing CS methods, as a first step, one computes the similarity between cases using all features involved and then the similarities are used to compute k -nearest neighbors, case coverage sets, and reachability sets. The feature importance can be determined using three different methods: all the feature weights are equal; or the feature weights are determined in advance with domain knowledge; or the feature weights are learned by training some models. Each method, however, has some limitations which offer challenges to both FR and CS.

When all the feature weights are equal, and feature importance is consequently not considered, the computed similarities may be misleading. This will result in wrongly computed k -nearest neighbors, case coverage, and reachability sets. This will in turn directly affect the quality of the cases selected using our proposed CS algorithms.

The second and third methods are also problematic to determine feature importance. When the feature weights must be determined in advance using required domain knowledge, the knowledge is obtained either by interviewing experts – which is labour intensive – or is extracted from the cases – which adds to the burden of training. Similarly, when feature weights must be learnt using models such as neural networks or decision trees, the burden of training is again not trivial, and even after training these models, the case representation is then in the form of a trained neural network or a number of rules, which is not convenient for directly retrieving similar cases from a case base for the unseen cases.

We address these problems by combining the fast rough-set-based FR approach with the CS algorithms. Feature importance is taken into account through reduct generation. The features in the reduct are regarded as the most important, while other features are considered to be irrelevant. Reduct computation does not require any domain knowledge and the computational complexity is only linear with respect to the number of attributes and cases. After combining the FR method and CS algorithms, the case representation is still the same as that of the original case base. This form of knowledge representation is easier to understand and more convenient for retrieving unseen cases. Furthermore, since only the features in the reduct are involved in the computations in the CS algorithms, the running time for case selection is also reduced.

For the CBR classifiers, there are three main benefits from combining FR with CS: (1) classification accuracy can be preserved or even improved by removing non-informative features and redundant and noisy cases; (2) storage requirements are reduced by deleting irrelevant features and redundant cases; (3) the classification decision response time can be reduced because fewer features and cases will be examined when an unseen case occurs.

In this work, we propose two ways to combine FR and CS based on different definitions of a ‘best’ subfeature set (approximate reduct) R^* . The first method – called an ‘open loop’ – applies the FR and CS sequentially and only once. The best approximate reduct is identified after applying FR alone. In construct, the second method can be regarded as a ‘close loop,’ which integrates FR and CS in an

RFRCS1 (Rough set-based Feature Reduction and Case Selection method):

Step 1 Initialize $P = \emptyset$, $Accr = \emptyset$, and $\beta = 1$. (P will store the reduced case bases after FR;
 $Accr$ will store the corresponding classification accuracies using these reduced case bases.)

Step 2 While ($\beta > 0$)

Implement Feature Reduction Algorithm; (Output the generated approximate reduct R)

$$P \leftarrow P \cup \Pi_R(U);$$

Implement unseen case classification using $\Pi_R(U)$; (Output the current accuracy, a)

$$Accr \leftarrow Accr \cup \{a\};$$

$$\beta = \beta - \lambda;$$

Step 3 Find a^* , $a^* = \max\{a \in Accr\}$; and find the corresponding R^* .

Step 4 Output the reduced final case base corresponding to R^* , denoted by $CB^* = \Pi_{R^*}(U)$.

Step 5 Let CB^* be the input original case base. Apply the case selection algorithms 1-4.

Figure 48.9 The RFRCS1 algorithm

interactive manner, determining the best approximate reduct after applying both FR and CS approaches. The interaction of FR and CS is reflected in the identification of the suitable β value.

In the first, ‘open loop,’ method, the ‘best’ approximate reduct R^* is defined as the approximate reduct which can achieve the highest accuracy after applying only the FR process. Such a best approximate reduct can be generated by iteratively tuning the value of the consistency measurement β . For example, we start from the exact reduct with $\beta = 1$, and in each iteration reduce β using a given parameter $\lambda = 0.01$. When the classification accuracy attains its maximum after applying FR alone, the approximate reduct is selected as R^* . In the following CS process, R^* is used to detect redundant and noisy cases.

In the second, ‘close loop,’ method, the ‘best’ subfeature set is defined as the approximate reduct which can achieve the highest accuracy after applying both FR and CS. R^* is determined much as in the first method. The value of consistency measurement β is modified with step length λ until it attains its maximum classification accuracy. Theoretically speaking, the ‘best’ approximation reduct found using the second method is not necessarily the same as that found using the first method. The two combination methods are described as follows (see Figures 48.9–48.10):

Obviously, the second combination method, RFRCS2, requires more computational effort because the ‘best’ approximate reduct depends on both FR and CS processes. For this reason, we mainly use the RFRCS1 method to test the performance of FR-CS combinations.

48.4.3 Experimental Results

In this section, we test our proposed CS algorithms, the combinations of the rough-set-based FR and CS, and provide comparisons with KPCA and SVMs techniques.

To demonstrate their effectiveness, we use three main evaluation indices: storage requirement, classification accuracy, and classification speed. For FR and CS processes, storage requirement has different meanings. The storage in FR is the percentage of preserved features after reducing features; in CS,

RFRCS2:

Step 1 Initialize $FCB = \emptyset$, $Accr = \emptyset$ and $\beta = 1$. (FCB will store the final reduced case bases after both FR and CS.)

Step 2 While ($\beta > 0$)

Implement Feature Reduction Algorithm; (*Output the reduced feature set R*)

Implement Case Selection Algorithms 1-4, where
the original input case base $CB = \Pi_R(U)$; (*Output final reduced case base FCB*.)

Implement unseen case classification using FCB ; (*Output the current accuracy, a*)
 $Accr \leftarrow Accr \cup \{a\}$;
 $\beta = \beta - \lambda$;

Step 3 Find $a^*, a^* = \max\{a \in Accr\}$; and find the corresponding R^* .

Step 4 Output the reduced final case base corresponding to R^* ,
denoted by $FCB^* = \Pi_{R^*}(U)$.

Figure 48.10 The RFRCS2 algorithm

storage is the percentage of selected cases. The classification accuracy is the percentage of the unseen cases which can be correctly classify.

1. For the *House-votes-84* data, four splits are generated by randomly selecting 20, 30, 40, and 50% as the testing data; the corresponding left data are used as the training data.
2. For the *Text data* and *Mushroom data*, we randomly select 80% documents in each text data set as the training data and the remaining 20% is used as the testing data.
3. For the *Multiple Features* data, we use the training/testing data sets in the original database, which has 1000 training samples and 1000 testing samples.

Tables 48.6 and 48.7 demonstrate the reduced storage and improved accuracy when using different CS algorithms. $P(W)$, $P(1)$, $P(2)$, and $P(4)$ represent the classification accuracy using Wilson editing, case selection algorithms 1, 2, and 4, respectively. Notice that the results of Algorithm 3 are very similar to those of Algorithm 1. Due to space limitations, they are not included in Tables 48.6 and 48.7 and related

Table 48.6 Case selection using the house-votes-84 data set

Split	P0 (%)	P (W) (%)	Storage	P(1) (%)	Storage	P(2) (%)	Storage	P(4) (%)	Storage
1	93.10	95.40	92.82	93.10	74.43	93.10	81.03	95.40	73.85
2	92.31	94.62	93.11	92.31	84.26	92.31	81.97	94.62	75.08
3	93.68	95.40	92.34	94.83	67.43	93.68	85.44	95.40	77.78
4	94.01	95.39	91.74	94.93	83.9	94.01	85.78	95.39	77.52
Avg.	93.28	95.20	92.50	93.79	77.40	93.28	83.56	95.20	76.06

Table 48.7 Case selection using text data sets

Data	P0 (%)	P(W) (%)	Storage	P(1) (%)	Storage	P(2) (%)	Storage	P(4) (%)	Storage
Text1	62.50	75.00	40.43	62.50	87.23	62.50	89.36	75.00	38.30
Text2	62.50	75.00	17.65	62.50	54.90	75.00	41.18	75.00	17.65
Text3	33.33	66.67	75.34	33.33	90.41	33.33	72.60	66.67	67.12
Text4	37.93	31.03	82.86	34.48	82.86	37.93	74.29	31.03	80.00
Text5	56.25	56.25	73.68	56.25	94.74	56.25	71.05	56.25	71.05
Text6	77.78	33.33	8.99	77.78	14.82	77.78	10.09	33.33	6.56
Text7	51.19	40.48	22.97	44.05	54.05	51.19	43.24	40.48	22.97
Text8	72.79	71.09	32.20	72.11	44.00	68.37	25.20	72.45	18.40
Avg.	56.78	56.11	44.27	55.38	65.38	57.79	53.38	56.28	40.26

results in the following sections. Here ‘storage’ means the proportion of cases which are selected in the final case base. In case selection algorithm 2, the parameter $\eta = 0.99$.

Table 48.6 (the house-votes data) shows that after case selection, all the CS algorithms were able to reduce cases while preserve or even improve classification accuracy. The Wilson editing and case selection algorithm 4 attain greatest accuracy, while Algorithm 4 has more powerful capability to reduce useless cases than other algorithms do. Table 48.7 shows the results for the text data sets. Algorithm 2 is most accurate. Algorithm 4 produced the smallest reduced case base with respect to the number of cases.

To summarize, both Tables 48.6 and 48.7 show results for Algorithm 4 that are satisfactory in terms of both classification accuracy and storage requirements after the case selection.

Some experiments using RFRCS1 and RFRCS2 were conducted to show the positive impact of the proposed rough-set-based FR method on the CS algorithms. The two main evaluation measurements are still storage and accuracy. Comparisons are made based on the k -NN classifier, using different CS algorithms in combination with FR. Here, k is set to a small odd number, 3.

Here, let $P(F + W)$, denote the classification accuracy of the combination of the rough-set-based FR and Wilson editing, and $P(F + 1)$, $P(F + 2)$, $P(F + 3)$, $P(F + 4)$ that of case selection algorithms 1 to 4. The final reduced case base is the case base containing the reduced feature set and the selected cases. The results of $P(3)$ and $P(F + 3)$ are similar to those of $P(1)$ and $P(F + 1)$ and are therefore not shown. Since the combination method RFRCS2 requires a greater computational effort, in this section we mainly conduct the experiments using the algorithm RFRCS1.

RFRCS1: Storage requirement and classification accuracy

1. House-votes-84.

Table 48.8 shows the results when using RFRCS1. On this data set, RFRCS1 incorporates the proposed fast rough-set-based FR approach into the CS algorithms. Obviously, the combined algorithms are

Table 48.8 Applying RFRCS1 to house-votes-84 ($\beta = 0.95$)

Split	P(W) (%)	P($F + W$) (%)	P(1) (%)	P($F + 1$) (%)	P(2) (%)	P($F + 2$) (%)	P(4) (%)	P($F + 4$) (%)
1	95.40	97.70	93.10	94.25	93.10	97.70	95.40	97.70
2	94.62	96.15	92.31	94.62	92.31	94.62	94.62	96.15
3	95.40	97.13	94.83	95.98	93.68	95.98	95.40	97.13
4	95.39	96.77	94.93	95.39	94.01	96.31	95.39	96.77
Avg.	95.20	96.94	93.79	95.06	93.28	96.15	95.20	96.94
	+ 1.74		+ 1.27		+ 2.87		+ 1.74	

Table 48.9 Applying RFRCS1 to text data sets ($\beta = 1$)

Data	$P(W)$ (%)	$P(F + W)$ (%)	$P(1)$ (%)	$P(F + 1)$ (%)	$P(2)$ (%)	$P(F + 2)$ (%)	$P(4)$ (%)	$P(F + 4)$ (%)
Text1	75.00	87.50	62.50	75.00	62.50	75.00	75.00	87.50
Text2	75.00	75.00	62.50	62.50	75.00	62.50	75.00	75.00
Text3	66.67	66.67	33.33	33.33	33.33	33.33	66.67	66.67
Text4	31.03	44.83	34.48	44.83	37.93	41.38	31.03	44.83
Text5	56.25	68.75	56.25	75.00	56.25	75.00	56.25	68.75
Text6	33.33	44.44	77.78	77.78	77.78	77.78	33.33	44.44
Text7	40.48	41.67	44.05	45.24	51.19	53.57	40.48	41.67
Text8	71.09	70.75	72.11	68.71	68.37	63.27	72.45	70.41
Avg.	56.11	62.45	55.38	60.30	57.79	60.23	56.28	62.41
	+6.34		+4.92		+2.44		+6.13	

more accurate and require less storage space than the approaches that make use of individual CS algorithms alone. Here, $\beta = 0.95$.

Algorithms $(F + W)$ and $(F + 4)$ are shown to be most accurate. The $(F + 4)$ algorithm also has the best classification accuracy and the most reduced storage requirement. This is because the Algorithm 4 is able to reduce the number of cases more effectively than algorithm that uses Wilson editing (Table 48.6). We can conclude that the fast rough-set-based FR approach using case selection algorithm 4 is superior to other FR and CS algorithms used either individually or in combination.

2. Text data sets.

This section examines the impact of FR on CS using the text data sets. Table 48.9 displays the text data set results. They are similar to those for the house-votes data set, except that the improvement in accuracy is much greater after incorporating FR to CS.

3. Mushroom data.

Table 48.10 shows the experimental results after applying RFRCS1 to the mushroom data set. Only the results of the case selection algorithm 1 and its combination with FR are contained in the table. This is because that, except the case selection algorithm 1, none of other algorithms were able to remove cases from the original case base. This is because that the Mushroom data are sparse and the CS algorithms 2 and 4 are suitable to the highly dense data. The classification accuracy of FR approach was the same as the original accuracy using the entire case base, 1. Therefore, $P_0 = P(FR) = P(W) = P(2) = P(4) = 1$. Table 48.10 shows the impact of feature reduction on case selection algorithm 1. On average, the classification accuracy after applying the combination of FR and the CS algorithm 1 increases by 9.3% from $P(1) = 89.6\%$ to $P(F + 1) = 98.9\%$. Here, the storage is with respect to cases instead of that of features. It is the percentage of cases which need to be stored in the final reduced case base after applying the algorithm 1. For the FR in algorithm $(F + 1)$, β is set to be 1. There are

Table 48.10 Applying RFRCS1 to mushroom data

Split	$P(1)$ (%)	$P(F + 1)$ (%)	Storage (%)
1	87.00	100.00	25.50
2	89.33	98.67	7.43
3	90.50	99.50	11.33
4	91.60	97.60	10.00
Avg.	89.61	98.94	13.57
	+9.33%		13.57

five features in the generated reduct so the storage requirement with respect to the feature set is 22.7% of the original feature set.

To conclude, when the combination method RFRCS1 is applied, the results of almost all of the data sets and of all of the proposed CS algorithms are positive. The classification accuracy and storage space requirement show a notable improvement. In almost all the testing, the combination of proposed rough set based FR approach with the CS algorithm 4, denoted by $(F + 4)$, is the most promising algorithm in terms of both classification accuracy and storage requirement.

RFRCS1: Classification Efficiency

This section describes some experiments carried out to determine the efficiency of case retrieval or unseen case classification after reducing both features and cases. The testing is also based on the algorithm RFRCS1.

Table 48.11 shows the average T_{FR} , T_{CS} , T_0 , T , and T_s using the three data sets. T_{FR} and T_{CS} are the average time cost in the FR process and the case selection method 4, respectively. T_0 is the average time needed to classify one unseen case using the entire original data sets. T is the average time needed to classify one unseen case using the reduced data set RFRCS1. $T_s = T_0 - T$. Since there are much fewer features and cases in the reduced data sets than those in the entire data sets, the case retrieval time of CBR classifiers using the reduced data sets will be much less than the time using the entire data sets. For this reason, T_s describes the amount of time that is saved for an unseen case classification due to this data compression. T_{FR} , T_{CS} , T_0 , T , and T_s are represented in seconds. The efficiency of case classification is improved using the reduced data sets. Although the average saved time of identifying only one unseen case is not notable, it could be significant using all the testing cases. For example, for the house-votes-84 data, the total saved time for predicting the class labels for all the 217 testing cases is $(0.03 \times 217) \approx 6.51$ s.

RFRCS2: Storage requirement and classification accuracy

Previously, we performed some experiments using the FR and CS combination method RFRCS1. The ‘best’ β and approximate reduct were determined through only in the FR process. Here the RFRCS2 is applied to real-life data, where the most suitable β is obtained when the final accuracy attains its maximum after both FR and CS. The experimental results show that the best β values found in RFRCS2 are not necessarily the same as those in RFRCS1. Compared with RFRCS1, using this kind of combination of FR with CS, the classification accuracy is shown to be further improved and/or the storage space could be further reduced.

48.5 Rough LVQ-Based Case Generation

We present a case generation approach which integrates fuzzy sets, rough sets, and learning vector quantization (LVQ). If the feature values of the cases are numerical, fuzzy sets are firstly used to discretize the feature spaces. Secondly, the fast rough-set-based FR is incorporated to identify the significant features. Finally, the representative cases (prototypes) are then generated through LVQ learning process on the case bases after FR. As a result, a few of prototypes are generated as the representative cases of the original case base. These prototypes can be considered as the extracted case knowledge granules which can improve the problem-solving efficiency and enhance the understanding of the case base.

Table 48.11 Speed of case classification using RFRCS1

Data sets	T_{FR}	T_{CS}	T_0	T	T_s
House-votes-84	0.343	0.004	0.10	0.07	0.03
Text data	5.597	0.020	1.58	0.02	1.56
Mushroom data	0.600	0.008	1.14	0.93	0.21

Three real-life data are used in the experiments to demonstrate the effectiveness of this case generation approach. Several evaluation indices, such as classification accuracy, the storage space, case retrieval time, and clustering performance in terms of intrasimilarity and intersimilarity, are used in these testing.

48.5.1 Introduction

Similar to the task of CS, CG is to extract the most representative cases from a given case base, which can build a new case base with smaller number of cases (i.e., a case knowledge base). The cases generated by CG process are not necessarily the data points of the given case base. The case representation form of these produced prototypical cases may be different from that of the original cases. For example, the support vectors which generated by SVMs and the rules learned by the decision trees, where some features may not exist in the newly extracted prototypical cases. Some other related work to CG includes [34–36] which generates cases through merging the cases in the same class or modifying the original cases. A rough fuzzy CG technique is proposed in [11, 12] which identifies the cluster granules as the newly generated cases. The rough-set-based FR methods developed in Section 48.2 are all built on the basis of indiscernibility relation. If the attribute values are continuous, the feature space needs to be discretized to define the indiscernibility relations and equivalence classes on different subset of attribute sets.

Fuzzy sets are used for the discretization by partitioning each attribute into three levels: ‘low’ (L), ‘medium’ (M), and ‘high’ (H). Finer partitions may lead to better accuracy at the cost of higher computational load. The use of fuzzy sets has several advantages over the traditional ‘hard’ discretizations, such as handling the overlapped clusters and linguistic representation of data [11].

Triangular membership functions are used to define the fuzzy sets: L, M, and H. There are three parameters C_L , C_M , and C_H for each attribute which should be determined beforehand. They are considered as the centers of the three fuzzy sets. Here the center of fuzzy set M for a given attribute a is the average value of all the values occurring in the domain of a .

Assume V_a is the domain of attribute a , then $C_M = \frac{\sum_{y \in V_a} y}{|V_a|}$, where $|*$ is the cardinality of set *. C_L and C_H are computed as

$$\begin{aligned} C_L &= (C_M - \text{Min}_a)/2, \\ C_H &= (\text{Max}_a - C_M)/2, \end{aligned}$$

where $\text{Min}_a = \min\{y | y \in V_a\}$ and $\text{Max}_a = \max\{y | y \in V_a\}$.

The membership functions are illustrated in Figure 48.11.

More formally, the membership functions for a given attribute a can be formulated as

$$\mu_L(x) = \begin{cases} 1, & \text{Min}_a \leq x \leq C_L \\ \frac{C_M - x}{C_M - C_L}, & C_L < x \leq C_M \\ 0, & x > C_M \end{cases}$$

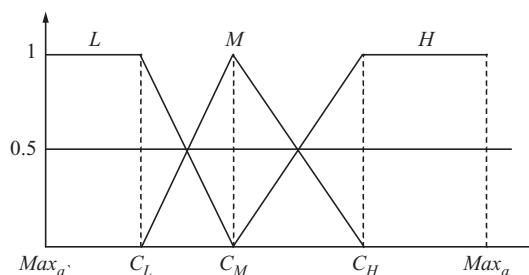


Figure 48.11 Membership functions of L, M, and H for attribute a

$$\mu_M(x) = \begin{cases} 0, & x \leq C_L \\ \frac{x - C_L}{C_M - C_L}, & C_L < x \leq C_M \\ \frac{C_H - x}{C_H - C_M}, & C_M < x \leq C_H \\ 1, & x > C_H \end{cases}$$

$$\mu_H(x) = \begin{cases} 0, & x \leq C_M \\ \frac{x - C_M}{C_H - C_M}, & C_M < x \leq C_H \\ 1, & x > C_H \end{cases}$$

where $\mu_*(x)$ is the membership value of case x to fuzzy set*.

48.5.2 Rough LVQ-Based Case Generation

LVQ derives from the self-organizing map (SOM), which is an unsupervised learning and robust to handle noisy and outlier data. The SOM can serve as a clustering tool of high-dimensional data. For classification problems, supervised learning LVQ should be superior to SOM since the information of classification results is incorporated to guide the learning process. LVQ is more robust to redundant features and cases and more insensitive to the learning rate. As Kohonen pointed out in [37], LVQ in stead of SOM should be used in decision and classification processes. This is the reason that LVQ is applied in case selection for building compact case base for CBR classifiers.

The basic idea of LVQ (see Figure 48.12) is the same as that of SOM, which is simple yet effective. It defines a mapping from high-dimensional input data space onto a regular two-dimensional array of nodes called competitive layer. Every node i of the competitive layer is associated with an m -dimensional vector $v_i = [v_{i1}, v_{i2}, \dots, v_{im}]$, where m denotes the dimension of the cases called reference vectors. The basic assumption here is that the nodes near to the same input vector should locate near to each other. Given an input vector, the most similar node in the competitive layer can be found as the winning node. Other nearby nodes for the input vector can also be found through similarity computation. Based on the mentioned assumption, the winning node and those nearby nodes should locate near to the input vector. The class information is also incorporated in the learning process. At each learning step, if the winning node and those nearby nodes are in the same class of input vector, the distances among these nodes are reduced; otherwise, these nodes are kept intact. This is different from the unsupervised learning process of SOM, where the winning node and those in its neighborhood will move toward each other even they are not in the same class. The amount of decrease in distance is determined by the given learning rate. As a result, after the learning with the reference vectors, LVQ converges to a stable structure and the final weight vectors are the cluster centres. These weight vectors are considered as the generated prototypes which can represent the entire case base.

Although LVQ has similar advantages of SOM, such as the robustness with noise and missing information, it does not mean that the data preprocessing is not required before the learning process. Since the

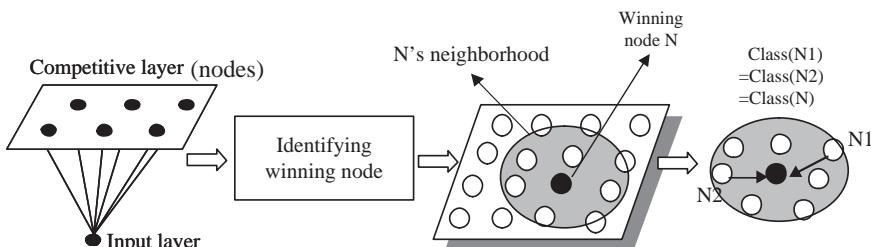


Figure 48.12 Outline of learning vector quantization

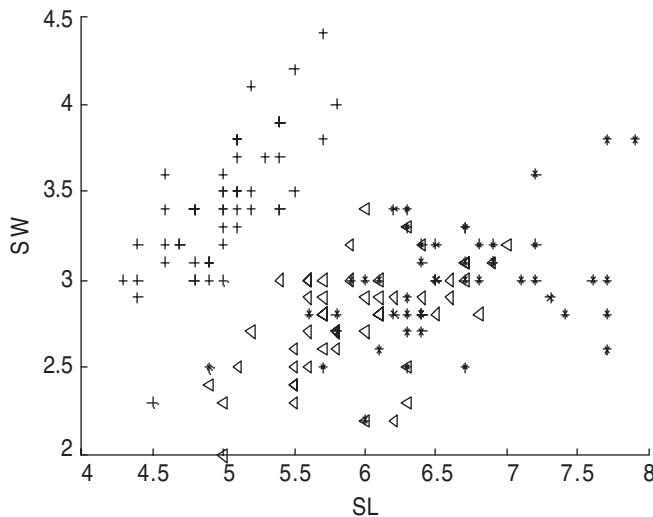


Figure 48.13 Iris data on SL and SW

basic assumption of LVQ is that similar feature values should lead to similar classification results, the similarity computation is critical in the learning process. Feature selection is one of the most important preparations for LVQ which can achieve better clustering and similarity computation results.

Different subset of features will result different data distribution and clusters. Take the Iris data [20] for example. Figures 48.13 and 48.14 shows the two dimensional Iris data on two different subset of features: {PW, PL} and {SW, SL}.

Based on the two subsets of features, LVQ is applied to learn three prototypes for the Iris data. The generated representative cases are shown in Tables 48.12 and 48.13.

It shows that different subset of attributes can affect the LVQ learning process and different prototypes are generated. According to the classification accuracy, the feature set of {PL, PW} is better than {SL, SW}.

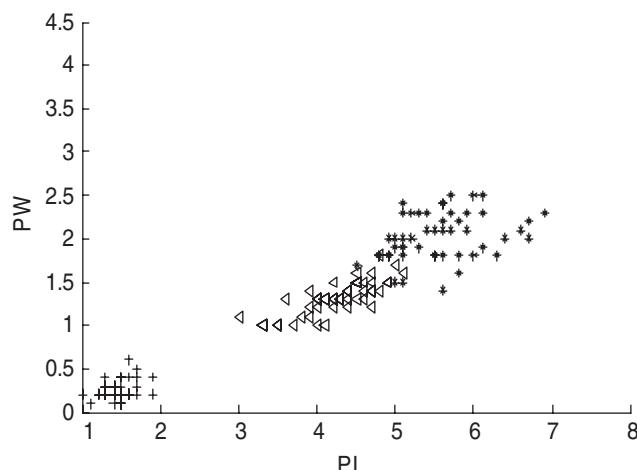


Figure 48.14 Iris data on PL and PW

Table 48.12 Prototypes extracted using PL and PW

Prototypes	SL	SW	PL	PW	Class label
P1	0.619	0.777	0.224	0.099	1
P2	0.685	0.613	0.589	0.528	2
P3	0.766	0.587	0.737	0.779	3
Classification accuracy using P1, P2, and P3: 0.98.					

The feature selection is addressed using the approximate reduct-based FR method. LVQ is then applied to generate representative cases for the entire case base. Here the learning rate α is given in advance, and only the distance between the winning node and the given input vector is updated in each learning step. The number of weight vectors is determined as the number of classes in the given case base. The learning process is ended with a fixed number of iterations T , say, 5000 in this chapter. Assume the given case base has n cases which represented by m features, and there are c classes. R is the approximate reduct computed by the feature selection process. The LVQ algorithm is given as follows:

LVQ-Based Case Generation Algorithm

Step 1. Initialize c weight vectors $[v_1, v_2, \dots, v_c]$ by randomly selecting one case from each class.

Step 2. Generate prototypes through LVQ.

```

 $t \leftarrow 1;$ 
While ( $t \leq T$ )
  for  $k = 1$  to  $n$ 
     $x \in U, x_k \leftarrow x, U \leftarrow U - \{x_k\};$ 
  1. Compute the distances  $D = \{\|x_k - v_{i,t-1}\|_R : 1 \leq i \leq c\};$ 
  2. Select  $v_{win,t-1} = \arg\{v_{i,t-1} : \|x_k - v_{i,t-1}\|_R = \min\{d \in D\}\};$ 
  3. If  $Class(v_{win,t-1}) = Class(x_k)$ 
    Update  $v_{win,t} = v_{i,t-1} + \alpha(x_k - v_{win,t-1});$ 
  4. Output  $V = [v_{1,T-1}, v_{2,T-1}, \dots, v_{c,T-1}]$ .

```

The output vectors are not the data points in the given case base, but modified during the learning process based on the provided information by the data. They are considered to be the generated prototypes which represent the entire case base. Each prototype can be used to describe the corresponding class and regarded as the cluster center.

48.5.3 Experimental Results

To illustrate the effectiveness of the developed rough LVQ case selection method, we describe here some results on three real-life data from UCI Machine Learning Repository [20]. These databases are Iris data,

Table 48.13 Prototypes extracted using SL and SW

Prototypes	SL	SW	PL	PW	Class label
P1	0.649	0.842	0.211	0.094	1
P2	0.712	0.550	0.572	0.212	2
P3	0.980	0.840	1.096	1.566	3
Classification accuracy using P1, P2, and P3: 0.80.					

Table 48.14 The characteristics of three UCI databases

Data set	Number of cases	Number of features	Category of features
Iris	150	4	Numerical
Glass	214	10	Numerical
Pima	768	8	Numerical

Glass data, and Pima data, whose characteristics are listed in Table 48.14. In all the experiments, 80% cases in each database are randomly selected for training and the remaining 20% cases are used for testing.

In this section, four indices are used to evaluate the rough LVQ case generation method. The *classification accuracy* is one of the important factors to be considered for building classifiers. On the other hand, the efficiency of CBR classifiers in terms of *case retrieval time* should not be neglected. The *storage space* and clustering performance (in terms of *intrasmilarity* and *intersimilarity*) are also tested in this section. Based on these evaluation indices, comparisons are made between our developed method and others such as basic SOM, basic LVQ, and random case selection methods.

The rough-set-based FR is firstly used to find the approximate reduct of the given case bases. In the experiments of this section, the parameter β is determined during the testing through populating the points in the interval [0.5, 1.0]. Initially, β is set to be 0.5. In each step, the β value increase at a constant rate 0.01 and this value is used in the feature selection process and being tested. The steps stop when β attains 1. The β value which can achieve the highest classification accuracy is selected as the suitable β . Based on the generated subset of features, the LVQ learning is then applied for extracting representative cases as the prototypes of the entire case base. The learning rates for the three data sets are $\alpha = 0.8$ (Iris data), $\alpha = 0.8$ (Glass data), and $\alpha = 0.5$ (Pima data). In the following sections, each evaluation index is tested to show the effectiveness of the rough LVQ case generation approach.

The used accuracies here are defined as

$$\text{Accuracy}_{\text{Test}} = \frac{|\{x, x \text{ can be correctly classified}, x \in \text{Testdata}\}|}{|\text{Testdata}|},$$

$$\text{Accuracy}_{\text{All}} = \frac{|\{x, x \text{ can be correctly classified}, x \in \text{Entiredata}\}|}{|\text{Entiredata}|},$$

where $|*$ is the cardinality of set $*$; *Testdata* is the set of cases for testing; *Entiredata* is the set of cases in the whole data set. To be more specifically, ‘ x can be correctly classified’ means that x can be correctly classified by the extracted prototypes.

If the training cases are used for classify the testing cases, the classification accuracies on the three databases are 0.980 (Iris), 0.977 (Glass), and 0.662 (Pima). These accuracy values are called the original classification accuracies. The experimental results of using the generated prototypes are demonstrated in Table 48.15. It is observed that after the case generation, the original accuracies are preserved and even

Table 48.15 Comparisons of classification accuracy using different case generation methods

Methods	Iris data		Glass data		Pima data	
	Accuracy _{Test}	Accuracy _{All}	Accuracy _{Test}	Accuracy _{All}	Accuracy _{Test}	Accuracy _{All}
Random	0.760	0.746	0.860	0.864	0.597	0.660
SOM	0.920	0.953	0.930	0.925	0.688	0.730
LVQ	0.980	0.953	0.930	0.935	0.708	0.743
Rough LVQ	1.000	0.960	0.930	0.935	0.714	0.740

Table 48.16 Reduced storage and saved case retrieval time

Data set	Reduced features (%)	Reduced cases (%)	Saved time of case retrieval (s)
Iris	50	97.0	0.600
Glass	60	98.8	0.989
Pima	50	99.6	0.924

improved. The rough LVQ method can achieve the highest classification accuracy in most of the testing. The basic LVQ method performs better than the other methods: random and SOM.

Due to both the feature selection and case selection processes, the storage space with respect to the features and cases is reduced substantially. Subsequently, the average case retrieval time will decrease. These results are shown in Table 48.16, where

$$\begin{aligned} \text{Reduce features} &= \left(1 - \frac{|\text{Selected features}|}{|\text{Original features}|} \right) \times 100\%, \\ \text{Reduced cases} &= \left(1 - \frac{|\text{Prototypes}|}{|\text{Entire data}|} \right) \times 100\%, \\ \text{Saved time of case retrieval} &= (t_{\text{train}} - t_p), \end{aligned}$$

where $|*|$ is the number of elements in set $*$; *selected features* is the set of features that are selected by the rough set-based method; *original features* is the set of features in the original database; *Prototypes* is the set of extracted representative cases; t_{train} is the case retrieval time to classify the testing cases using the training cases; t_p is the case retrieval time to classify the testing cases using the extracted prototypes; The unit of time is second.

From Table 48.16, the storage requirements of features and cases are reduced dramatically. For example, the percentage of reduced features is 60% for Glass data, and the percentage of reduced cases is 99.6% for Pima data. The case retrieval time also decreases because that there are much fewer features and cases after applying the rough LVQ-based case selection method.

Intrasimilarity and intersimilarity are two important indices to reflect the clustering performance. They are used in this section to prove that the developed rough LVQ-based approach can achieve better clustering than using random selected prototypes.

Since the similarity between two cases is inverse proportional to the distance between them, we use interdistance and intradistance to describe the inter-similarity and intra-similarity. These distances can be directly computed based on the numerical feature values. Assume there are K classes for a given case base, C_1, C_2, \dots, C_K . The intradistance and interdistance of the case base are defined as

$$\begin{aligned} \text{Intradistance} &= \sum_{x,y \in C_i} d(x, y), \\ \text{Interdistance} &= \sum_{x \in C_i, y \in C_j} d(x, y), i, j = 1, 2, \dots, K, i \neq j. \end{aligned}$$

$$\text{Ratio} = \text{Interdistance}/\text{Intradistance}.$$

The lower the intradistance and the higher the interdistance, the better is the clustering performance. Therefore, it is obvious that the higher the ratio between the interdistance and the intradistance, the better is the clustering performance.

The results are shown in Table 48.17. Rough LVQ method demonstrates higher ratio values and therefore achieves better clustering result.

Table 48.17 Interdistance and interdistance: Comparisons between the random and rough LVQ methods

Data set	Methods	Interdistance	Intradistance	Ratio
Iris	Random	1284.52	102.13	12.577
	Rough LVQ	1155.39	51.99	22.223
Glass	Random	8640.20	4567.84	1.892
	Rough LVQ	7847.37	3238.99	2.423
Pima	Random	56462.83	54529.05	1.035
	Rough LVQ	28011.95	25163.45	1.113

48.6 Summary and Conclusion

In this chapter, we have developed some soft computing techniques to extract case knowledge granules in the development of CBR systems. They reduce the size of the case base through removing the redundancy and noises, as well as preserve the problem-solving ability in terms of competence. The built techniques include rough-set-based feature reduction, GA-based supervised algorithm for learning similarity measures, case selection based on case coverage and reachability and NN principle, rough LVQ-based case generation. Among these techniques, the rough-set-based FR and CS are the most important methods, which identify and remove both irrelevant features and noisy and redundant cases. Thus, the size of the case bases is reduced and therefore the efficiency is improved, and simultaneously the accuracy and competence of the case bases are preserved or even improved. The combination of the proposed FR and CS methods are tested. The experimental results are very promising, and support our objective of trying to develop a compact and competent CBR system through case knowledge extraction.

Acknowledgments

This research project is supported by the Hong Kong Polytechnic University research grants G-T643 and A-PD55.

References

- [1] D.W. Aha, C. Marling, and I. Watson (eds). Special issue on case-based reasoning. *Knowl. Eng. Rev.* 20(3) (2005), 201–328.
- [2] B. Smyth and M. Keane. Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada, August 20–25, 1995, pp. 377–382.
- [3] B. Smyth and E. McKenna. Modeling the competence of case-bases. In: *Proceedings of the Fourth European Workshop on Case Based Reasoning (EWCBR-98)*, Dublin, Ireland, September 23–25, 1998, pp. 208–220.
- [4] B. Smyth. Case-base maintenance. In: *Proceedings of the Eleventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-98)*, Vol. 2, Castellón, Spain, June 1–4, 1998, pp. 507–516.
- [5] B. Smyth and E. McKenna. Building compact competent case bases. In: *Proceedings of the Third International Conference on Case-based Reasoning (ICCBR-99)*, Monastery Seeon, Munich, Germany, July 27–30, 1999, pp. 329–342.
- [6] B. Smyth and E. McKenna. Footprint-based retrieval. In: *Proceedings of the Third International Conference on Case-based Reasoning (ICCBR-99)*, Monastery Seeon, Munich, Germany, July 27–30, 1999, pp. 343–357.
- [7] D.D. Lewis. Reuters-21578 text categorization test collection distribution 1.0. <http://www.research.att.com/~lewis>, 1999, accessed February 2006.
- [8] Y. Li, S.C.K. Shiu, and S.K. Pal. Combining feature reduction and case selection in building CBR classifiers. In: S.K. Pal, D. Aha, and K.M. Gupta (eds), *Case-Based Reasoning in Knowledge Discovery and Data Mining*. John Wiley and Sons, NJ, 2006.

- [9] Y. Li, S.C.K. Shiu, and S.K. Pal. Combining feature reduction and case selection in building CBR classifiers. *IEEE Trans. Knowl. Data Eng.* 18(3) (2006) 415–429.
- [10] Y. Li, S.C.K. Shiu, S.K. Pal, and J.N.K. Liu. A rough set-based case-based reasoner for text categorization. *Int. J. Approx. Reason.* 41(2) (2006) 229–255. Special issue on Advances in fuzzy sets and rough sets, edited by Francesco Masulli and Alfredo Petrosino.
- [11] S.K. Pal and S.C.K. Shiu. *Foundations of Soft Case-Based Reasoning*. John Wiley, New York, 2004.
- [12] S.K. Pal and P. Mitra. Case generation using rough sets with fuzzy representation. *IEEE Trans. Knowl. Data Eng.* 16(3) (2004) 292–300.
- [13] S.C.K. Shiu, X.Z. Wang, and D.S. Yeung. Neural-fuzzy approach for maintaining case-bases. In: S.K. Pal, T. Dillon, and D. Yeung (eds), *Soft Computing in Case Based Reasoning*. Springer-Verlag, London, 2001, pp. 259–273.
- [14] S.C.K. Shiu, C.H. Sun, X.Z. Wang, and D.S. Yeung. Transferring case knowledge to adaptation knowledge: An approach for case-base maintenance. *Comput. Intell. Int. J.* 17(2) (2001) 295–314. Special Issue editors: David Leake, Barry Smyth, David Wilson and Qiang Yang.
- [15] Z. Pawlak. Rough sets. *Int. J. Comput. Inf. Sci.* 11 (1982) 341–356.
- [16] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory, Knowledge Engineering and Problem Solving*, Vol. 9. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.
- [17] Z. Pawlak and A. Skowron. Rudiments of rough sets. *Inf. Sci.* 177 (2007) 3–27.
- [18] A. Skowron and C. Rauszer. The discernibility matrices and functions in information systems. In: K. Slowinski (ed.), *Intelligent Decision Support-Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer, Dordrecht, 1992, pp. 331–362.
- [19] J. Han, X. Hu, and T.Y. Lin. Feature subset selection based on relative dependency between attributes. In: *Proceedings of the Fourth International Conference of Rough Sets and Current Trends in Computing (RSCTC-04)*, Uppsala, Sweden, June 1–5, 2004, pp. 176–185.
- [20] S. Hettich, C.L. Blake, and C.J. Merz. *UCI Machine Learning Databases*. Department of Information and Computer Science <http://www.ics.uci.edu/~mlearn/MLRepository.html>. Irvine, CA: University of California, accessed March 2006.
- [21] D. Beasley, D.R. Bull, and R.R. Martin. *An Overview of Genetic Algorithms Part 1: Fundamentals*. Technical report. University of Purdue, 1993.
- [22] W. Darrell. *A Genetic Algorithm Tutorial*. Technical Report, CS-93-103. Colorado State University, 1993.
- [23] H. Brighton and C. Mellish. Advances in instance selection for instance-based learning algorithms. *Data Min. Knowl. Discovery* 6(2) (2002) 153–172.
- [24] P.E. Hart. The condensed nearest neighbor rule. *Inst. Electr. Electron. Eng. Trans. Inf. Theory* 14 (1968) 515–516.
- [25] D.R. Wilson and L. Dennis. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. Syst. Man Cybern.* 2(3) (1972) 408–421.
- [26] G.W. Gates. The reduced nearest neighbor rule. *IEEE Trans. Inf. Theory* 18(3) (1972) 431–433.
- [27] G.L. Ritter, H.B. Woodruff, S.R. Lowry, and T.L. Isenhour. An algorithm for the selective nearest neighbor decision rule. *IEEE Trans. Inf. Theory* 21(6) (1975) 665–669.
- [28] I. Tomek. An experiment with the edited nearest-neighbor rule. *IEEE Trans. Syst. Man Cybern.* 6(6) (1976) 448–452.
- [29] G. Cao, S.C.K. Shiu, and X.Z. Wang. A fuzzy-rough approach for the maintenance of distributed case-based reasoning systems. *Soft Comput.* 7(8) (2003) 491–499.
- [30] K. Racine and Q. Yang. Maintaining unstructured case bases. In: *Proceedings of the Second International Conference on Case-based Reasoning (ICCBR-97)*, Providence, Rhode Island, July 25–27, 1997, pp. 553–564.
- [31] V.N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
- [32] D. Kim and C. Kim. Forecasting time series with genetic fuzzy predictor ensemble. *IEEE Trans. Fuzzy Syst.* 5(4) (1997) 523–535.
- [33] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1999.
- [34] C.L. Chang. Finding prototypes for nearest neighbor classifiers. *IEEE Trans. Comput.* C-23(11) (1974) 1179–1184.
- [35] P. Domingos. Rule induction and instance-based learning: A unified approach. In: *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada, August 20–25, 1995, pp. 1226–1232.
- [36] S. Salzberg. A nearest hyperrectangle learning method. *Mach. Learn.* 6(3) (1991) 251–276.
- [37] T. Kohonen. *Self-Organization and Associative Memory*. Springer-verlag, New York, 1988.

49

Granulation in Analogy-Based Classification

Arkadiusz Wojna

49.1 Introduction

To construct the knowledge about the world and to make decisions people use several reasoning methods. One of them consists in searching for similarities between concepts and facts and drawing conclusions by analogy. This reasoning paradigm has been successfully adopted in computer science. Research inspired by this paradigm established the new model of machine learning called analogy-based reasoning [1]. This model is strongly related to case-based reasoning [2, 3]. By analogy to natural human memory it assumes that a set of memorized examples is given and reasoning about a new fact is based on similar (analogous) facts from this set.

Analogy-based reasoning is more time consuming than other inductive methods. However, the advance of hardware technology and the development of indexing methods for memorized examples [4–21] have made possible the application of analogy-based reasoning to real-life problems.

Selection of a similarity measure among memorized examples is an important component of this approach that strongly affects the quality of reasoning. To construct a similarity measure and to compare examples we need to assume a certain fixed structure of examples. Most of data are collected in relational form: the examples are described by vectors of attribute values. Therefore, in the chapter we assume this original structure of data.

To construct a metric one can use both general mathematical properties of the domains of attribute values and specific information encoded in the memorized data. Diversity of attribute domain properties resulted in a variety of metrics proposed. Domains with numerical values provide rich algebraic information: they have the structure of linear order and provide the absolute difference as a natural metric. Such an information is often enough to construct an accurate metric for numerical attributes. The city-block (Manhattan) and Euclidean metrics are the most widely used examples [22–24]. In domains with nominal values the situation is different: the identity relation is the only relation available for a metric constructor. The first idea was to use the Hamming metric [22, 23] but its accuracy was not enough for practical applications. For this type of attributes the metric definition needs to use more information contained in data. This need resulted in the definition of value difference metric (VDM) [25] and many its variants [26–29]. Successful applications of metrics for nominal attributes induced from information in data encouraged researchers to construct analogous metrics for numerical attributes: Interpolated value difference metric (IVDM), windowed value difference metric (WVDM) [30, 31], and density-based value difference metric (DBVDM) [21].

The notions of similarity and metric can be used for classification problems. K nearest neighbors (k -nn) [21, 23, 29, 31–33] is the most popular analogy-based method used for this problem. Given a similarity measure in a space of objects an object from this space belongs to the concept if the majority of the k nearest examples belongs to the concept. In the chapter we show that on the basis of the notion of k nearest neighbors one can define a certain granulation in the space of objects. This allows us to formulate analogy-based methods as functions on granules and use them to reason about granules.

Concept learning by the methods from granular computing is expressed often with the use of rule language. We provide additional relation between analogy-based reasoning and granular computing by generalization of rule language to the metric-based form [21, 34]. Metric is the fundamental tool for analogy-based methods and they can be easily and effectively enhanced by such metric-based rules. The chapter shows that such combined rule-based and analogy-based approach can again be formulated in the language of functions and union and intersection operations on granules. This makes it possible to apply this combination to data classification problems.

The idea of combining rule-based and analogy-based approaches appeared also in some other methods.

RISE [28] combines rules with metrics in the way opposite to the idea of metric-based generalization of rules. It generalizes the notion of a metric in such a way that a metric can measure a distance between single objects and rules. Then it induces a set of rules and the rule nearest to a given object determines whether the object belongs to the concept or not.

Anpron [35, 36] and *DeEPsNN* [37] implement certain hybrid models.

Anpron induces a set of rules as the concept representation. For a given object to be classified it selects the matching rule from the rule set. But before the system decides about the objects it looks at exceptions to the rule in case library. If there are any exceptions to the rule, the system computes distances between the object and the exceptions. If it finds an exception close enough to the object, the decision is made on the basis of the exception. Otherwise the rule is used to make decision.

DeEPsNN works similar. It checks whether in a certain neighborhood of a given object to be classified there are any examples. If so 3-nn is applied, otherwise a set of rules is used to make a decision about this object.

In the chapter we build the correspondence between the notions of analogy and granulation and show how some analogy-based learning methods can be expressed in terms of operations on granules. The presented learning methods are based on simple analogy definitions which can be insufficient in problems with more structured domains. In such domains the choice of the appropriate similarity measure is difficult. An interesting solution to this problem has been proposed by Melanie Mitchell [38, 39]. She investigated the problem of guessing string transformation given another exemplary transformation. She proposed the model in which the learning system is provided a predefined set of elementary analogies related to the problem domain and the system learns from given question and example which elementary analogies are important and how to combine them together to define a final complex analogy. This model can be adapted to the world of granular computations. It provides solutions for building a system that selects the elementary granulations relevant for the problem and constructs the appropriate complex function from an available set of operations on granules.

The chapter is organized in the following way:

Section 49.2 introduces the reader to the classification problem and to the idea of learning from data. In particular it shows that learning from data is learning a function on elementary granulation.

In Section 49.3 we introduce the notion of metric and discuss its properties and possible enhancements for metrics. We also show that metric induced from data defines a distance measure between elementary granules.

Section 49.4 presents the basic model of analogy-based learning: the k nearest neighbors (k -nn) method in which a metric is its fundamental component. We describe how the notion of the k nearest neighbors defines certain granulation and show that k -nn can be defined as a function on this granulation.

Section 49.5 provides a review of the most popular metrics induced from data: the city-block (Manhattan) metric for numerical attributes and the Hamming metric and VDM for nominal attributes.

Section 49.6 describes the combination of k -nn and rules. It shows how the combined model can be formulated in terms of functions and union and intersection operations on granules. Section 49.7 builds additional bridge between analogy and rules. It provides metric-based extension of rules and shows

that such generalized rules define granulation and can also be used in the combined model of k -nn and rules.

Section 49.8 finalizes the chapter with some conclusions.

49.2 Learning a Concept from Examples

We assume that the target concept is defined over a universe of objects U^∞ . The concept to be learned is represented by a decision function $dec : U^\infty \rightarrow V_{dec}$. In the chapter we consider the situation when the decision is discrete and finite $V_{dec} = \{d_1, \dots, d_m\}$. The value $dec(x) \in V_{dec}$ for an object $x \in U^\infty$ represents the category of the concept that the object x belongs to. The decision function defines the partition $\{Class(d_j) : d_j \in V_{dec}\}$ of the universe U^∞ into the decision classes:

$$Class(d_j) = \{y \in U^\infty : dec(y) = d_j\}.$$

In the chapter we investigate the problem of decision learning from a set of examples. This problem relates to the situation when the target decision function $dec : U^\infty \rightarrow V_{dec}$ and the partition into decision classes are unknown. Instead of this there is a finite set of examples $U \subseteq U^\infty$ provided, and the decision values $dec(x)$ are available for the objects $x \in U$ only. The task is to provide an algorithmic method that learns a function (hypothesis) $h : U^\infty \rightarrow V_{dec}$, approximating the real decision function dec given only this set of examples U .

The objects from the universe U^∞ are real objects. In the chapter we assume that they are described by a set of n attributes $A = \{a_1, \dots, a_n\}$. Each real object $x \in U^\infty$ is represented by the object that is a vector of values (x_1, \dots, x_n) . Each value x_i is the value of the attribute a_i on this real object x . Each attribute $a_i \in A$ has its domain of values V_i and for each object representation (x_1, \dots, x_n) the values of the attributes belong to the corresponding domains: $x_i \in V_i$ for all $1 \leq i \leq n$. In other words, the space of object representations is defined as the product $V_1 \times \dots \times V_n$. The type of an attribute a_i is either numerical, if its values are comparable and can be represented by numbers $V_i \subseteq \mathbb{R}$ (e.g., age, temperature, and height), or nominal, if its values are incomparable, i.e., if there is no linear order on V_i (e.g., color, sex, and shape).

It is easy to learn a function that assigns the appropriate decision for each object in a set of examples $x \in U$. However, in most of decision learning problems a training set U is only a small sample of possible objects that can occur in real application and it is important to learn a hypothesis h that recognizes correctly as many objects as possible. The most desirable situation is to learn the hypothesis that is accurately the target function: $h(x) = dec(x)$ for all $x \in U^\infty$. Therefore the quality of a learning method depends on its ability to generalize information from examples rather than on its accuracy in the set of examples.

The information available about each exemplary object $x \in U$ is restricted to the vector of attribute values (x_1, \dots, x_n) and the decision value $dec(x)$. This feature defines the indiscernibility relation in the universe U^∞ : $IND = \{(x, x') \in U^\infty \times U^\infty : \forall a_i \in A \ x_i = x'_i\}$. The indiscernibility relation IND is an equivalence relation in the universe of objects U^∞ . The equivalence class of an object $x \in U^\infty$ in the relation IND is defined by $[x] = \{x' \in U : x \text{IND} x'\}$. Each equivalence class contains the objects that are indiscernible by the values of the attributes from the set A . The set of all equivalence classes defines a partition that is called the elementary granulation of the universe U^∞ :

$$\mathbb{X} = \{[x] : x \in U^\infty\}.$$

Each equivalence class $[x] \in \mathbb{X}$ is called an elementary granule in the universe U^∞ and corresponds to a certain vector of attribute values $(x_1, \dots, x_n) \in V_1 \times \dots \times V_n$:

$$[x] = \{x' \in U : \forall a_i \in A \ x'_i = x_i\}.$$

Since in the considered problem of concept learning the only information about new objects to be classified is the vector of attribute values (x_1, \dots, x_n) the objects from the universe U^∞ with the same value vector are indiscernible. Therefore searching for a hypothesis $h : U^\infty \rightarrow V_{dec}$ approximating the

real function $dec : U^\infty \rightarrow V_{dec}$ is restricted to searching for a hypothesis that is a function in the space of object representations: $h : V_1 \times \dots \times V_n \rightarrow V_{dec}$. This kind of hypothesis can be identified with the function $h^{\mathbb{X}} : \mathbb{X} \rightarrow V_{dec}$ in the set of elementary granules of the universe U^∞ :

$$h^{\mathbb{X}}([x]) = h(x_1, \dots, x_n).$$

This formula shows that searching for a solution to a classification problem is searching for a certain function on elementary granules.

49.3 Metric in the Universe of Objects

We assume that in the universe of objects U^∞ , a distance function $\rho : U^\infty \times U^\infty \rightarrow \mathbb{R}$ is defined. The distance function ρ is assumed to satisfy the axioms of a pseudometric; i.e., for any objects $x, y, z \in U^\infty$,

1. $\rho(x, y) \geq 0$ (positivity),
2. $\rho(x, x) = 0$ (reflexivity),
3. $\rho(x, y) = \rho(y, x)$ (symmetry),
4. $\rho(x, y) + \rho(y, z) \geq \rho(x, z)$ (triangular inequality).

The distance function ρ models the relation of similarity between objects. The properties of symmetry and triangular inequality are not necessary to model similarity but they are fundamental for the efficiency of the indexing and searching methods [7, 8, 11, 14, 17, 20]. Sometimes the definition of a distance function satisfies the strict positivity: $x \neq y \Rightarrow \rho(x, y) > 0$. However, the strict positivity is not used by the distance-based learning algorithms and a number of important distance measures like VDM [25], IVDM, WVDM [30], and DBVDM [21] do not satisfy this property.

Since the only information about an object x to be classified is the vector of its attribute values (x_1, \dots, x_n) , we assume that this vector can only be used in any metric definition. Each metric ρ based only on attribute values induces a certain metric $\rho^{\mathbb{X}} : \mathbb{X}^2 \rightarrow \mathbb{R}$ in the set of elementary granules \mathbb{X} :

$$\rho^{\mathbb{X}}([x], [y]) = \rho(x, y).$$

In the l_p -norm based metric the distance between two objects x represented by (x_1, \dots, x_n) and y represented by (y_1, \dots, y_n) is defined by

$$\rho(x, y) = \left(\sum_{i=1}^n \rho_i(x_i, y_i)^p \right)^{\frac{1}{p}}, \quad (1)$$

where the metrics $\rho_i : V_i^2 \rightarrow \mathbb{R}$ are the distance functions defined for particular attributes $a_i \in A$.

Aggarwal *et al.* [40] have examined the meaningfulness of the concept of similarity in high-dimensional real-value spaces investigating the effectiveness of the l_p -norm based metric in dependence on the value of the parameter p . They proved the following result:

For the uniform distribution of 2 points x, y in the cube $(0, 1)^n$ with the norm l_p ($p \geq 1$):

$$\lim_{n \rightarrow \infty} E \left[\left(\frac{\max(\|x\|_p, \|y\|_p) - \min(\|x\|_p, \|y\|_p)}{\min(\|x\|_p, \|y\|_p)} \right) \sqrt{n} \right] = C \sqrt{\frac{1}{2p+1}},$$

where C is a positive constant and $\|\cdot\|_p$ denotes the standard norm in the space l_p .

It shows that the smaller p , the larger relative contrast is between the point closer to and the point farther from the beginning of the coordinate system. It indicates that the smaller p , the more effective metric is induced from the l_p -norm. In the context of this result, $p = 1$ is the optimal trade-off between the quality of the measure and its properties: $p = 1$ is the minimal index of the l_p -norm that preserves the triangular inequality. The fractional distance measures with $p < 1$ do not have this property. Hence in

in many applications the value $p = 1$ is assumed and the linear sum of metrics $\rho_i : V_i^2 \rightarrow \mathbb{R}$ for particular attributes $a_i \in A$ is used:

$$\rho(x, y) = \sum_{i=1}^n \rho_i(x_i, y_i). \quad (2)$$

In the problem of learning from a set of examples U the particular distance functions ρ_i are induced from a training set U . It means that the metric definition depends on the provided examples and it is different for different data sets.

The metric definition in equation (2) treats all attributes $a_i \in A$ as equally important. However, there are numerous factors that make attributes unequally significant for concept learning. For example,

- Some attributes can be strongly correlated with the concept, while other attributes can be independent of the concept.
- More than one attribute can correspond to the same information; hence, taking one attribute into consideration can make other attributes redundant.
- Some attributes can contain noise in values, which makes them less trustworthy than attributes with the exact information.

Therefore, in many applications attribute weighting has a significant impact on the metric quality. To take all these factors into account and to improve the quality of a metric one can use the weighted version of the metric definition:

$$\rho(x, y) = \sum_{i=1}^n w_i \cdot \rho_i(x_i, y_i). \quad (3)$$

In literature one can find many attribute weighting methods proposed to optimize a metric [21, 22, 41–43].

49.4 k Nearest Neighbors as Granule-Based Classifier

One of the most popular analogy-based classifiers is the k nearest neighbors (k -nn) [21, 23, 29, 31–33]. First, k -nn induces a metric ρ from a set of examples U (the examples of such metrics are provided in the next section) and stores all the examples from U in memory. Then for each object x to be classified k -nn finds the k nearest neighbors $NN(x, k)$ of the object x among the examples in U according to the metric ρ . The set of the nearest neighbors $NN(x, k)$ is then used for selection of the decision for x . The most common model of voting by the neighbors is the majority model. In this model the object x is assigned with the most frequent decision among the k nearest neighbors:

$$dec_{k-nn}(x) := \arg \max_{d_j \in V_{dec}} |\{y \in NN(x, k) : dec(y) = d_j\}|. \quad (4)$$

Ties are broken arbitrary in favor of the decision d_j with the smallest index j or in favor of a randomly selected decision among the ties.

In the majority voting model all the k neighbors in $NN(x, k)$ are equally important. In the literature there are a number of other voting models that take into consideration the distances from the neighbors to the test object x [33, 44]. Dudani [33] proposed the inverse distance weight where the weight of a neighbor vote is inversely proportional to the distance from this neighbor to the test object x :

$$dec_{weighted-knn}(x) := \arg \max_{d_j \in V_{dec}} \sum_{y \in NN(x, k) : dec(y) = d_j} \frac{1}{\rho(x, y)}. \quad (5)$$

In this way closer neighbors are more important for classification than farther neighbors. It has been argued that the significance of the information for an object to be tested provided by the neighbors

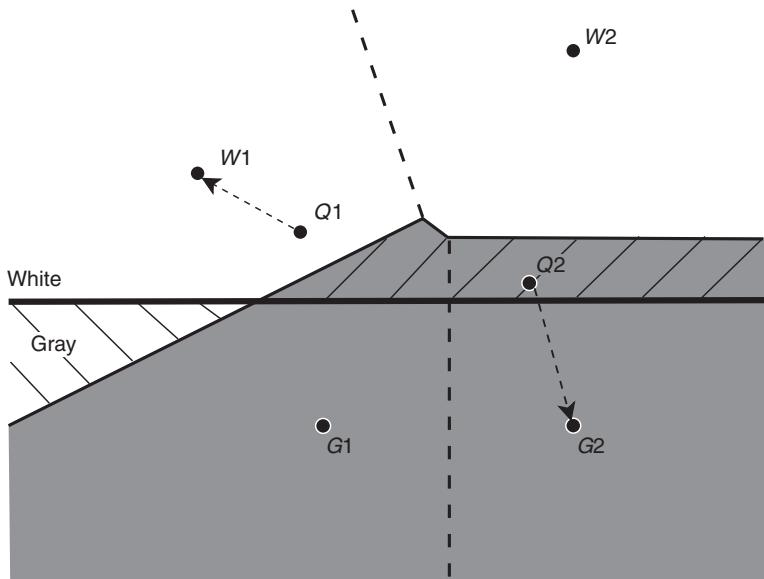


Figure 49.1 Learning a cut in the Euclidean plane by 1-nn from four examples: W_1 , W_2 , G_1 , and G_2 . The cut is defined by the thick horizontal line. The area above the cut is labeled as *white* and the area below as *gray*.

correlates with the distance of the neighbors to the tested object [21, 23, 29, 45–47] and therefore the distance weighted voting models can outperform the majority voting model.

The k nearest neighbors method is a basic example of analogy-based reasoning. In this approach a reasoning system assumes that there is a database providing the complete information about exemplary objects. When the system is asked about another object with an incomplete information, it retrieves similar (analogous) objects from the database and the missing information is completed on the basis of the information about the retrieved objects.

In the k -nn, the induced metric ρ plays the role of a similarity measure. The smaller the distance is between two objects, the more similar they are. It is important for the similarity measure to be defined in such a way that it uses only the information that is available both for the exemplary objects in the database and for the object in the query. In the problem of decision learning it means that the metric uses only the values of the non-decision attributes.

Let us consider a simple example of a concept to be learnt from Figure 49.1. In this example the plane \mathbb{R}^2 is the universe of objects U^∞ and the function $dec : \mathbb{R}^2 \rightarrow \{\text{White}, \text{Gray}\}$ is defined by

$$dec(x) = \begin{cases} \text{White} & \text{if } x \text{ lies above the thick horizontal line} \\ \text{Gray} & \text{otherwise.} \end{cases}$$

The task is to learn the concept dec from the set of 4 examples $U = \{W_1, W_2, G_1, G_2\}$. W_1 and W_2 are examples of the class *White* and G_1 and G_2 are examples of the class *Gray*.

We describe how this concept is learnt by k -nn in case of $k = 1$ (the nearest neighbor rule) and the Euclidean metric. In this case a point to be classified is assigned with the decision of the nearest example. For example on Figure 49.1 the example W_1 is the nearest one to the point Q_1 , so Q_1 is classified as *White*, and G_2 is the nearest example to the point Q_2 , so Q_2 is classified as *Gray*. The set of all the points that are classified by one example, i.e., all the points for which this example is the nearest one, constitute a polygon on the plane that can be either bounded or unbounded. The set of such polygons determined by particular examples constitute the Voronoi diagram.

In this example each decision class occupies a certain half plane: *White* occupies the upper half plane over the horizontal cut and *Gray* occupies the bottom half plane. The white and gray areas at Figure 49.1 show the function defined by the nearest neighbor rule dec_{1-nn} , not the original decision classes. The nearest neighbor rule approximates the area occupied by each decision class with the polygons of the Voronoi diagram determined by the examples that belong to this class. The half plane *White* is approximated by the two white unbounded polygons determined by the examples $W1$ and $W2$. The half plane *Gray* is approximated by the two gray unbounded polygons determined by the examples $G1$ and $G2$.

One could see the most of the area visible at Figure 49.1 is classified correctly but the two areas adjacent to the horizontal cut (shaded slantwise) are misclassified: the band over the cut stretching to the right belongs to the class *White* but it is classified as *Gray* and the unbounded triangle on the left belongs to the class *Gray* but it is classified as *White*. In particular, the point $Q2$ is an example of the misclassified point: it belongs to the class *White* but is classified as *Gray*. This example shows that like other classification models k nearest neighbors is not a perfect model of an original concept. It defines its certain approximation.

Sections 49.2 and 49.3 describe how the problem of decision learning and the notion of metric relate to the elementary granulation of the universe. Now we show that the k nearest neighbors classifier can be defined as a granule-based classifier. First, we introduce the definitions of the k -neighborhood radius $R(x, k)$ and the k -neighborhood $B(x, k)$:

$$R(x, k) = \min \{r \in \mathbb{R} : |\{y \in U : \rho(y, x) \leq r\}| \geq k\}.$$

The k -neighborhood $B(x, k)$ is defined as the ball in the universe U^∞ centered at x with the radius $R(x, k)$:

$$B(x, k) = \{y \in U^\infty : \rho(x, y) \leq R(x, k)\}.$$

The set of k -neighborhoods for all objects $x \in U^\infty$ defines the granulation $\pi(U, k)$ of the universe:

$$\pi(U, k) = \{B(x, k) : x \in U^\infty\}.$$

Figure 49.2 show 1-neighborhoods for the three points: $Q1$, $Q2$, and $Q3$. In case of $k = 1$, each granule $B(x, 1)$ is the ball centered at x in which the nearest example lies on the boundary of the ball.

The notion of k -neighborhood can be used to define formally the set of the k nearest neighbors of any object x from the universe U^∞ :

$$NN(x, k) = \{y \in U : \rho(y, x) \leq R(x, k)\} = B(x, k) \cap U. \quad (6)$$

The examples from the set U constitute all the available information about the concept to be learned. From equation (6) one can see that the set of the k nearest neighbors of x represents all the information on the granule $B(x, k)$ available for the classifier. K -nn selects the decision for x using this information.

Now we define the following function $dec^{\pi(U, k)} : \pi(U, k) \rightarrow V_{dec}$ for the granulation $\pi(U, k)$:

$$dec^{\pi(U, k)}(B(x, k)) = \arg \max_{d_j \in V_{dec}} |\{y \in B(x, k) \cap U : dec(y) = d_j\}|. \quad (7)$$

One can write this definition using the partition of the universe into the decision classes:

$$dec^{\pi(U, k)}(B(x, k)) = \arg \max_{d_j \in V_{dec}} |B(x, k) \cap Class(d_j) \cap U|.$$

From equations (4), (6), and (7) we obtain the following formula for the k nearest neighbors classifier:

$$dec_{k-nn}(x) = dec^{\pi(U, k)}(B(x, k)).$$

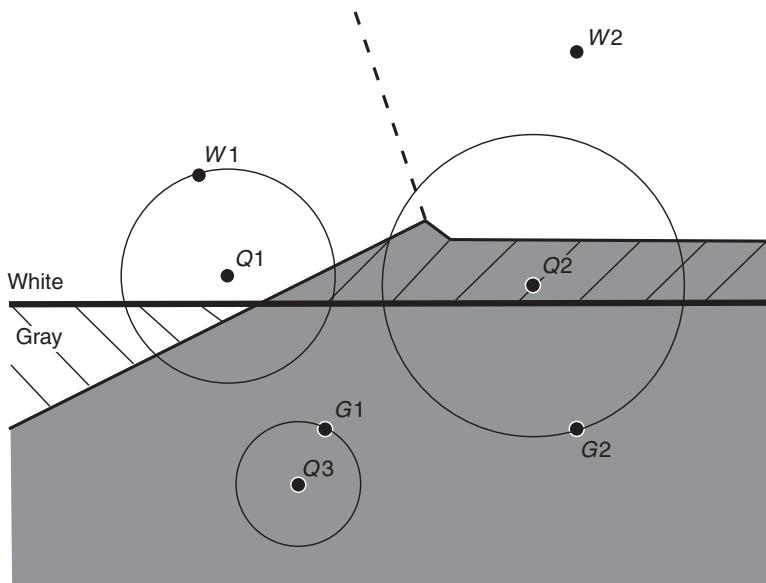


Figure 49.2 The examples of 1-neighborhoods: $B(Q_1, 1)$, $B(Q_2, 1)$, and $B(Q_3, 1)$ with regard to four examples: W_1 , W_2 , G_1 , and G_2 from the problem of horizontal cut learning

One can provide the analogous function on granules to define the weighted version of the k -nn classifier:

$$dec_{weighted}^{\pi(U,k)}(B(x, k)) = \arg \max_{d_j \in V_{dec}} \sum_{y \in B(x, k) \cap Class(d_j) \cap U} \frac{1}{\rho(x, y)}$$

$$dec_{weighted-knn}(x) = dec_{weighted}^{\pi(U,k)}(B(x, k)).$$

These formulas prove that k -nn can be defined as a function on certain granulation of the universe. This allows us to express this kind of classifiers in terms of the language of granules and builds a bridge between analogy-based reasoning and granular computing.

49.5 Metric Examples

In this section we present a review of the most popular metrics used in metric-based classifiers. To construct a metric one can use both general mathematical properties of the domains of attribute values and specific information provided by the examples.

The nature of numerical and nominal attributes is completely different. Domains with numerical values provide rich algebraic information: they have the structure of linear order and provide the absolute difference as a natural metric. In domains with nominal values the identity relation is the only relation available for a metric constructor. This difference caused the specialization of metrics for particular attribute types: metrics for numerical attributes are often based on the absolute difference whereas metrics for nominal attributes mainly use the information contained in data.

49.5.1 City-Block and Hamming Metric

This metric is widely used in the literature. It combines the city-block (Manhattan) distance for the values of numerical attributes and the Hamming distance for the values of nominal attributes.

The distance $\rho_i(x_i, y_i)$ between two values x_i and y_i of a numerical attribute a_i in the city-block distance is defined by

$$\rho_i(x_i, y_i) = |x_i - y_i|. \quad (8)$$

The scale of values for different domains of numerical attributes can be different. To make the distance measures for different numerical attributes equally significant it is better to use the normalized value difference. Two types of normalization are used. In the first one the difference is normalized with the range of the values of the attribute a_i

$$\rho_i(x_i, y_i) = \frac{|x_i - y_i|}{\max_i - \min_i}, \quad (9)$$

where $\max_i = \max_{x \in U} x_i$ and $\min_i = \min_{x \in U} x_i$ are the maximal and the minimal value of the attribute a_i in the training set U . In the second type of normalization the value difference is normalized with the standard deviation of the values of the attribute a_i in the training set U :

$$\rho_i(x_i, y_i) = \frac{|x_i - y_i|}{2\sigma_i}$$

where $\sigma_i = \sqrt{\frac{\sum_{x \in U} (x_i - \mu_i)^2}{|U|}}$ and $\mu_i = \frac{\sum_{x \in U} x_i}{|U|}$.

The distance $\rho_i(x_i, y_i)$ between two nominal values x_i, y_i in the Hamming distance is defined by the Kronecker delta:

$$\rho_i(x_i, y_i) = \begin{cases} 1 & \text{if } x_i \neq y_i \\ 0 & \text{if } x_i = y_i. \end{cases}$$

The combined city-block and Hamming metric sums the normalized value differences for numerical attributes and the values of Kronecker delta for nominal attributes. The normalization of numerical attributes with the range of values $\max_i - \min_i$ makes numerical and nominal attributes equally significant: the range of distances between values is [0; 1] for each attribute. The only possible distance values for nominal attributes are the limiting values 0 and 1, whereas the normalized distance definition for numerical attributes can give any value between 0 and 1. It results from the type of an attribute: the domain of a nominal attribute is only a set of values and the only relation in this domain is the equality relation. The domain of a numerical attribute are the real numbers and this domain is much more informative: it has the structure of linear order and the natural metric, i.e., the absolute difference.

Below we define an important property of metrics related to numerical attributes:

The metric ρ is *consistent with the natural linear order of numerical values* if and only if for each numerical attribute a_i and for each three real values $v_1 \leq v_2 \leq v_3$, the following conditions hold: $\rho_i(v_1, v_2) \leq \rho_i(v_1, v_3)$ and $\rho_i(v_2, v_3) \leq \rho_i(v_1, v_3)$.

The values of a numerical attribute reflect usually a measure of a certain natural property of analyzed objects, e.g., size, age, or measured quantities like temperature. Therefore, the natural linear order of numerical values helps often obtain useful information for measuring similarity between objects and the notion of metric consistency describes the metrics that preserve this linear order.

The city-block metric is consistent with the natural linear order.

The city-block metric depends linearly on the absolute difference as defined in equation (8) or (9). Since the absolute difference is consistent with the natural linear order, the city-block metric is consistent too.

49.5.2 Joint City-Block and Value Difference Metric

Section 49.5.1 provides the metric definition that combines the city-block metric for numerical attributes and the Hamming metric for nominal attributes. In this section we focus on nominal attributes.

The definition of the Hamming metric uses only the relation of equality in the domain of values of a nominal attribute. This is the only relation that can be assumed in general about nominal attributes. This relation carries often insufficient information; in particular, it is much less informative than the structure of the domains for numerical attributes where the values have the structure of linear order with a distance measure between the values.

Although in general one can assume nothing more than equality relation on nominal values, in the problem of learning from examples the goal is to induce a classification model from examples assuming that a problem and data are fixed. It means that in the process of classification model induction the information encoded in the database of examples should be used. In the k nearest neighbors method this database can be used to extract meaningful information about relation between values of each nominal attribute and to construct a metric.

This fact has been used first by Stanfill and Waltz who defined a measure to compare the values of a nominal attribute [25]. The definition of this measure, called the value difference metric, is valid only for the problem of learning from examples. It defines how much the values of a nominal attribute $a_i \in A$ differ in relation to the decision dec . More precisely, the VDM estimates the conditional decision probability $P(dec = d_j | a_i = v)$ given a nominal value v and uses the estimated decision probabilities to compare nominal values. The VDM distance between two nominal values x_i and y_i is defined by the difference between the estimated decision probabilities $P(dec = d_j | a_i = x_i)$, $P(dec = d_j | a_i = y_i)$ corresponding to the values x_i , y_i (see Figure 49.3):

$$\rho_i(x_i, y_i) = \sum_{d_j \in V_{dec}} |P(dec = d_j | a_i = x_i) - P(dec = d_j | a_i = y_i)|. \quad (10)$$

The estimation of the decision probability $P(dec = d_j | a_i = v)$ is done from the training set U . For each value v , it is defined by the decision distribution in the set of all the training objects that have the value of the nominal attribute a_i equal to v :

$$PVDM(dec = d_j | a_i = v) = \frac{|\{x \in U : dec(x) = d_j \wedge x_i = v\}|}{|\{x \in U : x_i = v\}|}.$$

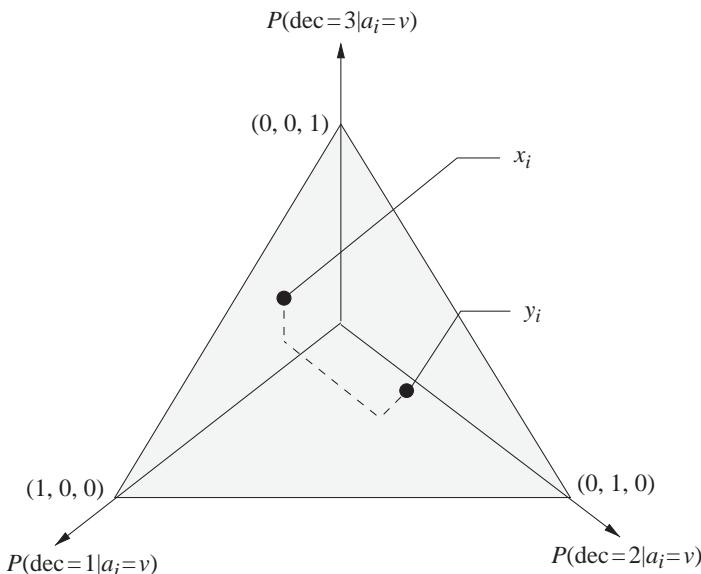


Figure 49.3 An example: the value difference metric for the three decision values $V_{dec} = \{1, 2, 3\}$. The distance between two nominal values x_i and y_i corresponds to the length of the dashed line

From equation (10) and the definition of $P_{VDM}(dec = d_j | a_i = v)$ one can see that the more similar the correlations between each of two nominal values $x_i, y_i \in V_i$ and the decisions $d_1, \dots, d_m \in V_{dec}$ in the training set of examples U are the smaller the distance in equation (10) is between x_i and y_i . Different variants of these metric were used in many applications [25–27].

To define a complete metric the VDM needs to be combined with another distance function for numerical attributes. For each pair of possible data objects $x, y \in U^\infty$, the following condition $\rho_i(x_i, y_i) \leq 2$ is satisfied for any nominal attribute $a_i \in A$. It means that the range of possible distances for the values of nominal attributes in the VDM is $[0; 2]$. It corresponds well to the city-block distance for a numerical attribute a_i normalized by the range of the values of this attribute in the training set U (see Section 49.5.1):

$$\rho_i(x_i, y_i) = \frac{|x_i - y_i|}{\max_i - \min_i}.$$

The range of this normalized city-block metric is $[0; 1]$ for the objects in the training set U . In the whole universe U^∞ this range can be exceeded but it happens very rarely in practice. The most important property is that the ranges of such a normalized numerical metric and the VDM are of the same order.

The above-described combination of the distance functions for nominal and numerical attributes was proposed by Domingos [28]. The experimental results [21] proved that this combination is more effective than the same normalized city-block metric combined with the Hamming metric.

49.5.3 Extensions of Value Difference Metric for Numerical Attributes

The normalized city-block metric used in the previous section to define the joint metric uses information from the training set: it normalizes the difference between two numerical values v_1 and v_2 by the range of the values of a numerical attribute $\max_i - \min_i$ in the training set. However, it defines the distance between values of the numerical attribute on the basis of the information about this attribute only, whereas the distance definition for nominal attributes makes use of the correlation between the nominal values of an attribute and the decision values. Since this approach improved the effectiveness of metrics for nominal attributes, analogous solutions have been provided for numerical attributes.

Wilson and Martinez proposed two distance definitions for numerical attributes analogous to VDM: windowed value difference metric (WVDM) [30] and interpolated value difference metric (IVDM) [30, 31]. WVDM estimates the decision probability of a given attribute value in an interval of attribute values centered at this value. IVDM is an effective approximation of the metric WVDM.

In WVDM and IVDM the intervals used for decision probability estimation have always the same length. They do not take into account varying density of numerical values. In many problems the density of numerical values is not constant and the relation of being similar between two numerical values depends on the range where these two numerical values occur. It means that the same difference between two numerical values has different meaning in different ranges of the attribute values. For example, the meaning of the temperature difference of the one Celsius degree for the concept of water freezing is different for the temperatures over 20° and for the temperatures close to zero. To solve this problem density-based value difference metric (DBVDM) [21] was proposed that is a modification of the WVDM metric.

The metrics IVDM, WVDM, and DBVDM use the information about the correlation between the numerical values and the decision from the training set. However, contrary to the city-block metric none of those three metrics is consistent with the natural linear order of numerical values (see Definition 49.5.1). All the three metrics are based more than the city-block metric on the information included in training data and less on the general properties of numerical attributes.

49.6 Combination of k -nn and Rules as Operation on Granules

In Section 49.4 it was shown that k -nn can be defined as a function on granules. In this section we describe how multistrategy approach with the k nearest neighbors as one of the strategies can be defined in terms of operations on granules. In particular, we consider the combination of the k nearest neighbors method

with rule-based approach. Rules are one of the most popular notion used to describe granulation of the universe of objects. They are also widely used in machine learning [48–52].

Multistrategy approach is expected to combine information from both models at the moment of classification. This idea in application to k -nn and rules was described in [34]. The method selects a set of examples that participate in decision voting using both the set of k nearest neighbors and the rules.

Each rule $\alpha \Rightarrow dec = d_j$ consists of a premise α and a consequent $dec = d_j$. The premise α is a formula describing the objects from the universe U^∞ that match the rule. The consequent $dec = d_j$ denotes the decision value that is assigned to an object if it matches the rule.

Let R be any set of rules. The definition of the combined classifier does not make any assumptions about this rule set. It uses the notion of k -support:

The k -support of a rule $\alpha \Rightarrow dec = d_j$ for an object x is the set:

$$k - support(x, \alpha \Rightarrow dec = d_j) = \{y \in NN(x, k) : y \text{ matches } \alpha \wedge dec(y) = d_j\}.$$

The k -support of the rule contains only those objects from the original support set that belong to the set of the k nearest neighbors.

Now, we define the classification model that combines the k -nn method with any set of rules R by using the k -supports of these rules:

$$dec_{knn-rules}(x, R) := \arg \max_{d_j \in V_{dec}} \left| \bigcup_{\alpha \Rightarrow dec = d_j \in R: x \text{ satisfies } \alpha} k - support(x, \alpha \Rightarrow dec = d_j) \right|. \quad (11)$$

The combined classifier can be defined by the equivalent formula:

$$dec_{knn-rules}(x, R) := \arg \max_{d_j \in V_{dec}} \sum_{y \in NN(x, k): dec(y) = d_j} \delta(y, R), \quad (12)$$

where the value of $\delta(y, R)$ is defined by

$$\delta(y, R) := \begin{cases} 1 & \text{if } \exists r \in R \text{ that covers } x \text{ and is supported by } y \\ 0 & \text{otherwise} \end{cases}.$$

The equivalent definition shows that the combined classifier can be considered as a special sort of the k nearest neighbors method: it can be viewed as the k -nn classifier with the specific rule-based zero-one voting model. Such a zero-one voting model is a sort of filtering: it excludes some of the k nearest neighbors from voting. This gives more certainty that the remaining neighbors are appropriate for decision making. Such a voting model can easily be combined with other voting models, e.g., with the inverse distance weights defined in equation (5):

$$dec_{weighted-knn-rules}(x, R) := \arg \max_{d_j \in V_{dec}} \sum_{y \in NN(x, k): dec(y) = d_j} \frac{\delta(y, R)}{\rho(x, y)}. \quad (13)$$

For any set of rules R each rule from R covers a certain set of objects from the universe. For a given rule $\alpha \Rightarrow dec = d_j$, we denote this set by $[\alpha]$. In other words, $[\alpha]$ is the set of all objects from the universe U^∞ matching the rule $\alpha \Rightarrow dec = d_j$. If a set of rules satisfies a certain condition, it defines a granulation:

If each object x in the universe U^∞ matches at least one rule $r \in R$:

$$U^\infty = \bigcup_{\alpha \Rightarrow dec = d_j} [\alpha]$$

then the set of rules R defines the granulation:

$$\pi(R) = \{[\alpha] : \alpha \Rightarrow dec = d_j \in R\}.$$

By $\pi_{d_j}(R)$, we denote the subset of granules represented by the rules with the decision $d_j \in V_{dec}$:

$$\pi_{d_j}(R) = \{[\alpha] : \alpha \Rightarrow dec = d_j \in R\}.$$

Then $\pi(R) = \pi_{d_1}(R) \cup \dots \cup \pi_{d_m}(R)$, where $V_{dec} = \{d_1, \dots, d_m\}$.

In Section 49.4 we defined the k -nn classifier as a function on the granulation $\pi(U, k)$ (see equation (7)). By analogy we can define the classifier combining k -nn with rules in terms of operations on granules. To define it we consider the union of granulations $\pi(U, k) \cup \pi(R)$. Since the granulations are closed with regard to the union operation the union $\pi(U, k) \cup \pi(R)$ is also a granulation of the universe U^∞ .

Since $NN(x, k) = B(x, k) \cap U$ (see equation (6)) k -support of a rule $\alpha \Rightarrow dec = d_j$ for an object x can be defined by the formula

$$k-support(x, \alpha \Rightarrow dec = d_j) = B(x, k) \cap [\alpha] \cap Class(d_j) \cap U.$$

This formula shows that k -support is a result of an operation on granules restricted to the information in the training set U .

Hence the classifier $dec_{knn-rules}$ can be defined by the equivalent formula:

$$dec_{knn-rules}(x) = \arg \max_{d_j \in V_{dec}} \left| B(x, k) \cap \bigcup_{[\alpha] \in \pi_{d_j}(R): x \in [\alpha]} [\alpha] \cap Class(d_j) \cap U \right|. \quad (14)$$

In this way we formulated the classifier combining k -nn with rules in terms of union and intersection operations on granules from the granulation $\pi(U, k) \cup \pi(R)$. The result of this operations is restricted to the available information about the concept to be learnt (the training set) and the decision assigned is a function defined on this result.

49.7 Granulation by Metric-Based Rules

In this section we consider granulation with the use of rules and show how this kind of granulation can be generalized to a metric-dependent form. A popular granulation used in classification methods is the granulation by the set of all minimal consistent rules [52]. In this granulation each minimal consistent rule describes a single granule. Minimal consistent rules are defined in the context of a set of examples. The set of all minimal consistent rules has good theoretical properties: there is one-to-one correspondence between minimal consistent rules and local reducts [53].

The original notion of a minimal consistent rule was based on the form of a rule with equality conditions:

A *rule* consists of a premise and a consequent:

$$a_{i_1} = v_1 \wedge \dots \wedge a_{i_p} = v_p \Rightarrow dec = d_j.$$

The premise is conjunction of attribute conditions and the consequent indicates a decision value. A rule is said to cover an object x with attribute values (x_1, \dots, x_n) , and vice versa; the example x is said to match the rule if all the attribute conditions in the premise of the rule are satisfied by the object values: $x_{i_1} = v_1, \dots, x_{i_p} = v_p$.

The notions of consistency and minimality relate to the set of available examples U . Consistency describes the rules that classify correctly all the covered objects in a given training set:

A rule $a_{i_1} = v_1 \wedge \dots \wedge a_{i_p} = v_p \Rightarrow dec = d_j$ is *consistent* with a training set U if for each object $x \in U$ matching the rule the decision of the rule is correct; i.e., $dec(x) = d_j$.

The notion of minimality selects the consistent rules of the minimum length in terms of the number of conditions in the premise of a rule:

A consistent rule $a_{i_1} = v_1 \wedge \dots \wedge a_{i_p} = v_p \Rightarrow dec = d_j$ is *minimal* in a training set U if for each proper subset of conditions occurring in the premise of this rule $C \subset \{a_{i_1} = v_1, \dots, a_{i_p} = v_p\}$ the rule built from these conditions; i.e., $\bigwedge C \Rightarrow dec = d_j$ is inconsistent with the training set U .

Minimal consistent rules have the important property: they maximize the set of covered objects in a training set U .

Let us denote the set of all minimal consistent rules for the training set U by $MCR(U)$. It is known that minimal consistent rules cover the whole universe U^∞ and in this way they define a granulation of the universe:

If a training set U is consistent, i.e., there is no pair of objects $x, y \in U$ such that $x_i = y_i$ for each $a_i \in A$ and $dec(x) \neq dec(y)$, then the set:

$$\pi(MCR(U)) = \{[\alpha] : \alpha \Rightarrow dec = d_j \text{ is a consistent minimal rule}\}$$

is a granulation of the universe U^∞ .

To prove that $\pi(MCR(U))$ is a granulation we need to show that for each object $x \in U^\infty$, there is at least one granule $[\alpha] \in \pi(MCR(U))$ such that $x \in [\alpha]$. Let us consider the rule $a_1 = x_1 \wedge \dots \wedge a_n = x_n \Rightarrow dec = d_j$. If there is an example $y \in U$ such that $x_i = y_i$ for all $a_i \in A$, then d_j is equal to the decision of this example $dec(y)$; otherwise d_j can be any decision from V_{dec} . Since U is consistent, all the objects $z \in U$ with $z_i = x_i$ for all $a_i \in A$ have the same decision as y : $dec(z) = dec(y) = d_j$. Hence, the rule $a_1 = x_1 \wedge \dots \wedge a_n = x_n \Rightarrow dec = d_j$ is consistent with U . Now, as long as in the rule there is a condition that can be removed and the rule remains consistent, we remove this condition. After removing each condition the rule covers all objects that it covered before removal, so at the end, when no condition can be removed, we obtain a minimal consistent rule that still covers x .

The original definition of minimal consistent rules was proposed for data with nominal attributes and it used equality as the only form of conditions on attributes in the premise of a rule (see Definition 7). In [34] this approach was generalized to data with both nominal and numerical attributes and with a metric ρ of the form from equation (1). Equality as the condition in the premise of the rule from Definition 7 represents selection of attribute values, in this case always a single value. Equality conditions can be replaced with a more general metric-based form of conditions. This form allows us to select more than one attribute value in a single attribute condition and, thus, to obtain more general rules:

A *generalized rule* consists of a premise and a consequent:

$$\rho_{i_1}(v_1, *) \leq r_1 \wedge \dots \wedge \rho_{i_p}(v_p, *) < r_p \Rightarrow dec = d_j.$$

Each condition $\rho_{i_q}(v_q) \leq r_q$ or $\rho_{i_q}(v_q) < r_q$ in the premise of the generalized rule represents the range of acceptable values of a given attribute a_{i_q} around a given value v_q . The range is specified by the distance function ρ_{i_q} that is the component of the total distance ρ and by the threshold r_q .

Consistency definition for a generalized rule is the same as for an equality-based rule (see Definition 7). The next step is the generalization of the notion of rule minimality:

A consistent generalized rule $\rho_{i_1}(v_1, *) < r_1 \wedge \dots \wedge \rho_{i_p}(v_p, *) < r_p \Rightarrow dec = d_j$ is *minimal* in a training set U if for each attribute $a_{i_q} \in \{a_{i_1}, \dots, a_{i_p}\}$ occurring in the premise of the generalized rule the rule $\rho_{i_1}(v_1, *) < r_1 \wedge \dots \wedge \rho_{i_q}(v_q, *) \leq r_q \wedge \dots \wedge \rho_{i_p}(v_p, *) < r_p \Rightarrow dec = d_j$ with the enlarged range of acceptable values on this attribute (obtained by replacing $<$ by \leq in the condition of the original rule) is inconsistent with the training set U .

Observe that each condition in the premise of a minimal consistent generalized rule is always a strict inequality. It results from the assumption that a training set U is finite.

Let us denote the set of all generalized minimal consistent rules by $GMCR(U)$. This set retains the granulation property:

If a training set U is metrically consistent, i.e., there is no pair of objects $x, y \in U$ such that $\rho(x, y) = 0$ and $dec(x) \neq dec(y)$, then the set

$$\pi(GMCR(U)) = \{[\alpha] : \alpha \Rightarrow dec = d_j \text{ is a generalized minimal consistent rule}\}$$

is a granulation of the universe U^∞ .

To prove that $\pi(GMCR(U))$ is a granulation we need to show that for each object $x \in U^\infty$, there is at least one granule $[\alpha] \in \pi(GMCR(U))$ such that $x \in [\alpha]$. In other words, there is at least one generalized minimal consistent rule that covers the object x .

We define the sequence of rules r_0, \dots, r_n in the following way. The first rule r_0 in the sequence is the rule $\rho_1(x_1, *) = 0 \wedge \dots \wedge \rho_n(x_n, *) = 0 \Rightarrow dec = d_q$. If there is an example $y \in U$ such that $\rho(x, y) = 0$, then d_q is equal to the decision of this example $dec(y)$; otherwise, d_q can be any decision from V_{dec} . Since U is metrically consistent all the objects $z \in U$ with $\rho(x, z) = 0$ have the same decision as y : $dec(z) = dec(y) = d_q$. Hence, the rule $\rho_1(x_1, *) = 0 \wedge \dots \wedge \rho_n(x_n, *) = 0 \Rightarrow dec = d_q$ is consistent with U . To define each next rule r_i we assume that the previous rule r_{i-1} :

$$\bigwedge_{1 \leq j < i} \rho_j(x_j, *) < max_j \wedge \bigwedge_{i \leq j \leq n} a_j = x_j \Rightarrow dec = d_q.$$

is consistent with the training set U and the first $i - 1$ conditions of the rule r_{i-1} are maximally general; i.e., replacing any strong inequality $\rho_j(a_j, x_j) < max_j$ for $j < i$ by the weak makes this rule inconsistent. Let S_i be the set of all the examples that satisfy the premise of the rule r_{i-1} with the condition on the attribute a_i removed:

$$S_i = \{z \in U : z \text{ satisfies } \bigwedge_{1 \leq j < i} \rho_j(x_j, *) < max_j \wedge \bigwedge_{i < j \leq n} a_j = x_j\}.$$

In the rule r_i the i th condition is maximally extended in such a way that the rule remains consistent. It means that the range of acceptable values for the attribute a_i in the rule r_i has to be not larger than the attribute distance from x to any object in S_i with a decision different from d_q . If S_i does not contain an object with a decision different from d_q , the range remains unlimited:

$$max_i = \begin{cases} \infty & \text{if } \forall z \in S_i dec(z) = d_q \\ \min\{\rho_i(x_i, z_i) : z \in S_i \wedge dec(z) \neq d_q\} & \text{otherwise.} \end{cases} \quad (15)$$

If we limit the range of values on the attribute a_i in the rule r_i by the max_i with the strong inequality in the condition:

$$\bigwedge_{1 \leq j < i} \rho_j(x_j, *) < max_j \wedge \rho_i(x_i, *) < max_i \wedge \bigwedge_{i < j \leq n} a_i = x_i \Rightarrow dec = d_q,$$

then it ensures that the rule r_i remains consistent. On the other hand, the value of max_i in equation (15) has been chosen in such a way that replacing the strong inequality by the weak inequality or replacing the range by a value larger than max_i causes the situation where a certain object with a decision different from d_q satisfies the condition on the attribute a_i and the whole premise of the rule r_i ; i.e., the rule r_i becomes inconsistent.

Since r_{i-1} was consistent and the condition $\rho_i(x_i, *) < max_i$ covers all the objects that satisfy the condition $a_i = x_i$ from the rule r_{i-1} the ranges for the previous attributes max_1, \dots, max_{i-1} remain maximal in the rule r_i : widening one of these ranges in the rule r_{i-1} makes an inconsistent object match r_{i-1} and the same happens for the rule r_i .

By induction the last rule $r_n : \bigwedge_{1 \leq j \leq n} \rho_j(x_j, *) < max_j \Rightarrow dec = d_q$ in the defined sequence is consistent too and all the conditions are maximally general. Then r_n is consistent and minimal. Since the first rule r_0 covers the object x and each rule r_i covers all the objects matching the previous rule r_{i-1} in the sequence, the last rule r_n which is a generalized minimal consistent rule covers the object x too.

W proved that the set of all generalized minimal consistent rules $GMCR(U)$ defines a granulation $\pi(GMCR(U))$ of the universe if the training set U is metrically consistent. It shows that the notion of metric can be used to define new kinds of granules and in this way the expressibility of the granulation models is significantly enlarged.

If one uses generalized minimal consistent rules in the classifier combining k -nn with rules the decision assigned to an object is the result of operation on granules defined by these generalized rules (see equation (14)).

The number of all minimal consistent rules can be exponential in relation both to the number of attributes $|A|$ and to the training set size $|U|$ [21] and this fact remains true for generalized version of minimal consistent rules. Therefore computing all minimal consistent rules is often infeasible and many

rule induction algorithms are based on a smaller set of rules. However, there is the effective classification method [54] that classifies objects on the basis of the set of all minimal consistent rules without computing them explicitly. This method also works for generalized rules and has been successfully incorporated into the classifier combining k -nn and all generalized minimal consistent rules [21, 34]. This makes the method possible to be used in practice.

49.8 Conclusion

This chapter established the bridge between analogy-based classification and granular computing. It shows that analogy-based approach uses information granules to define a data model. The chapter also describes the types of information granules and the operations on granules used in this approach.

First the relation between the elementary granulation and the notion of similarity was considered. Similarity is the fundamental notion used in analogy-based methods and classification models. It is often represented by a metric. We showed that each metric induced from data defines the corresponding distance measure between elementary granules. It allows us to reason not only about particular objects but about granules too.

K -neighborhoods are another type of granules used in analogy-based approach. A granule of this type is defined as the ball-shaped neighborhood of an object in the metric space. This granule type reflects the natural idea of selecting the group of objects that are similar to a given single object. The k nearest neighbors, the basic model of analogy-based approach, is a certain function defined on such granules.

In the chapter we also showed that the operations of intersections and unions on granules can be used to describe hybrid models. The classification method combining the k nearest neighbors with rules is an example in which k -neighborhoods are combined by these operations with more classical kind of granules, described with the use of rule language. Such rule-based granules are often used in granular computations.

The next step was to extend the language of rules by the use of metric. Such metric-based rules describe metric-based granules. This extension widens the expressibility of rule language and allows us to describe new kinds of granules and new granulation models. In particular, the generalized rules can also be used in the classification models combining analogy-based approach with rules. Such a combined model has been implemented as one of the classification methods in the rough set exploration system [24, 55] and has been verified on real-life data.

The bridge between analogy-based reasoning and granular computing presented in this chapter is based on natural observation that all objects similar to a given object constitute a granule. In this way granulation represents a similarity relation. For many real-life problems the problem of defining an accurate similarity measure is a challenging problem and still not solved. The granulation-based approach has important features. Analogies found in real-life problems are often non-symmetrical and intransitive and in this sense granulation is universal: similarity described by granulation does not have to be symmetrical or transitive. Moreover, in classical analogy-based methods the similarity measure is usually defined globally for the whole universe of objects. This is not perfect in case of specific cases and anomalies and limits accuracy of the model. Granulation-based similarity does not make any global assumption about similarity relation that is invariable in the whole universe. Those features make granular computations an interesting approach to define similarities and give the chance to overcome the limitations of global approach and to build more accurate similarity models.

The similarity measure can be even more accurate if it is learned from data. This issues another challenge for granular computing: how to search for the granulation that is the most appropriate for a given problem. Mellanie Mitchell investigated this for a certain specific domain [38, 39] and the results are promising. However, the question how to approach this problem in general remains still open.

Acknowledgments

The research has been supported by the grant N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland and by the grant Innovative Economy Operational Programme 2007–2013 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland.

References

- [1] S.J. Russell. *Use of Knowledge in Analogy and Induction*. Morgan Kaufmann, San Francisco, CA, 1989.
- [2] D.W. Aha. The omnipresence of case-based reasoning in science and applications. *Knowl.-Based Syst.* 11(5–6) (1998) 261–273.
- [3] D.B. Leake (ed.) *Case-Based Reasoning: Experiences, Lessons and Future Directions*. AAAI Press/MIT Press, MA, 1996.
- [4] N. Beckmann, H.P. Kriegel, R. Schneider, and B. Seeger. The R^* -tree: An efficient and robust access method for points and rectangles. In: *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, Atlantic City, NJ, 1990, pp. 322–331.
- [5] J.L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM* 18(9) (1975) 509–517.
- [6] S. Berchtold, D. Keim, and H.P. Kriegel. The X-tree: An index structure for high dimensional data. In: *Proceedings of the Twenty Second International Conference on Very Large Databases*, Mumbai, India, September 3–6, 1996, pp. 28–39.
- [7] S. Brin. Near neighbor search in large metric spaces. In: *Proceedings of the Twenty First International Conference on Very Large Databases*, Zurich, Switzerland, September 11–15, 1995, pp. 574–584.
- [8] E. Chavez, G. Navarro, R. Baeza-Yates, and J.L. Marroquin. *Searching in Metric Spaces*. Technical Report TR/DCC-99-3. Department of Computer Science, University of Chile, Chile, 1999.
- [9] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In: *Proceedings of the Twenty Third International Conference on Very Large Databases*, Athens, Greece, August 25–29, 1997, pp. 426–435.
- [10] R. Finkel and J. Bentley. Quad-trees: A data structure for retrieval and composite keys. *ACTA Inf.* 4(1) (1974) 1–9.
- [11] K. Fukunaga and P.M. Narendra. A branch and bound algorithm for computing k-nearest neighbors. *IEEE Trans. Comput.* 24(7) (1975) 750–753.
- [12] V. Gaede and O. Gunther. Multidimensional access methods. *ACM Comput. Surv.* 30(2) (1998) 170–231.
- [13] A. Guttman. R-trees: A dynamic index structure for spatial searching. In: *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, Boston, MA, 1984, pp. 47–57.
- [14] I. Kalantari and G. McDonald. A data structure and an algorithm for the nearest point problem. *IEEE Trans. Softw. Eng.* 9(5) (1983) 631–634.
- [15] J. Nievergelt, H. Hinterberger, and K. Sevcik. The grid file: An adaptable symmetric multikey file structure. *ACM Trans. Database Syst.* 9(1) (1984) 38–71.
- [16] J. Robinson. The K-D-B-tree: A search structure for large multi-dimensional dynamic indexes. In: *Proceedings of the 1981 ACM SIGMOD International Conference on Management of Data*, New York, 1981, pp. 10–18.
- [17] J. Uhlmann. Satisfying general proximity/similarity queries with metric trees. *Inf. Process. Lett.* 40(4) (1991) 175–179.
- [18] J. Ward, Jr. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* 58 (1963) 236–244.
- [19] A.G. Wojna. Center-based indexing for nearest neighbors search. In: *Proceedings of the Third IEEE International Conference on Data Mining*, Melbourne, FL, 2003. IEEE Computer Society Press, pp. 681–684.
- [20] A.G. Wojna. Center-based indexing in vector and metric spaces. *Fundam. Inf.* 56(3) (2003) 285–310.
- [21] A.G. Wojna. Analogy-based reasoning in classifier construction. In: *Transactions on Rough Sets IV*, Vol. 3700 of *Lectures Notes in Artificial Intelligence*. Springer-Verlag, 2005, pp. 277–374.
- [22] D.W. Aha. Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *Int. J. Man-Mach. Stud.* 36 (1992) 267–287.
- [23] D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Mach. Learn.* 6 (1991) 37–66.
- [24] J.G. Bazan, M. Szczuka, A.G. Wojna, and M. Wojnarski. On the evolution of Rough Set Exploration System. In: *Proceedings of the Fourth International Conference on Rough Sets and Current Trends in Computing*, volume 3066 of *Lectures Notes in Artificial Intelligence*, Uppsala, Sweden, 2004. Springer-Verlag, pp. 592–601.
- [25] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Commun. ACM* 29(12) (1986) 1213–1228.
- [26] Y. Biberman. A context similarity measure. In: *Proceedings of the Ninth European Conference on Machine Learning*, Catania, Italy, 1994, pp. 49–63.
- [27] S. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Mach. Learn.* 10 (1993) 57–78.
- [28] P. Domingos. Unifying instance-based and rule-based induction. *Mach. Learn.* 24(2) (1996) 141–168.
- [29] D. Wetschereck. *A Study of Distance-Based Machine Learning Algorithms*. Ph.D. Thesis. Oregon State University, 1994.
- [30] D.R. Wilson and T.R. Martinez. Improved heterogeneous distance functions. *J. Artif. Intell. Res.* 6 (1997) 1–34.

- [31] D.R. Wilson and T.R. Martinez. An integrated instance-based learning algorithm. *Comput. Intell.* 16(1) (2000) 1–28.
- [32] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.
- [33] S. Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Trans. Syst. Man Cybern.* 6 (1976) 325–327.
- [34] A.G. Wojna. Combination of metric-based and rule-based classification. In: *Proceedings of the Tenth International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, Vol. 3641 of *Lectures Notes in Artificial Intelligence*, Regina, Canada, 2005. Springer-Verlag, pp. 501–511.
- [35] A.R. Golding and P.S. Rosenbloom. Improving rule-based systems through case-based reasoning. In: *Proceedings of the Ninth National Conference on Artificial Intelligence*, Anaheim, CA, 1991, pp. 22–27.
- [36] A.R. Golding and P.S. Rosenbloom. Improving accuracy by combining rule-based and case-based reasoning. *Artif. Intell.* 87(1–2) (1996) 215–254.
- [37] J. Li, K. Ramamohanarao, and G. Dong. Combining the strength of pattern frequency and distance for classification. In: *Proceedings of the Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Hong Kong, 2001, pp. 455–466.
- [38] M. Mitchell. Analogy-making as complex adaptive system. In: L. Segel and I. Cohen (eds), *Design Principles for the Immune System and Other Distributed Autonomous Systems*. Oxford University Press, New York, 2001.
- [39] M. Mitchell. *Analogy-Making as Perception: A Computer Model*. MIT Press, Cambridge, MA, 1993.
- [40] Ch. C. Aggarwal, A. Hinneburg, and D.A. Keim. On the surprising behaviour of distance metrics in high dimensional space. In: *Proceedings of the Eighth International Conference on Database Theory*, London, UK, 2001, pp. 420–434.
- [41] I. Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In: *Proceedings of the Seventh European Conference on Machine Learning*, Vol. 784 of *Lectures Notes in Artificial Intelligence*, Catania, Italy, 1994. Springer-Verlag, pp. 171–182.
- [42] D. Lowe. Similarity metric learning for a variable kernel classifier. *Neural Comput.* 7 (1995) 72–85.
- [43] D. Wettschereck, D.W. Aha, and T. Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artif. Intell. Rev.* 11 (1997) 273–314.
- [44] R.N. Shepard. Toward a universal law of generalization for psychological science. *Science* 237 (1987) 1317–1323.
- [45] J.E.S. Macleod, A. Luk, and D.M. Titterington. A re-examination of the distance-weighted k-nearest-neighbor classification rule. *IEEE Trans. Syst. Man Cybern.* 17(4) (1987) 689–696.
- [46] D. Wolpert. Constructing a generalizer superior to NETtalk via mathematical theory of generalization. *Neural Netw.* 3 (1989) 445–452.
- [47] J. Zavrel. An empirical re-examination of weighted voting for k-nn. In: *Proceedings of the Seventh Belgian-Dutch Conference on Machine Learning*, Tilburg, The Netherlands, 1997, pp. 139–148.
- [48] J.G. Bazan and M. Szczuka. RSES and RSESlib – a collection of tools for rough set computations. In: *Proceedings of the Second International Conference on Rough Sets and Current Trends in Computing*, Vol. 2005 of *Lectures Notes in Artificial Intelligence*, Banff, Canada, 2000. Springer-Verlag, pp. 106–113.
- [49] P. Clark and T. Niblett. The CN2 induction algorithm. *Mach. Learn.* 3 (1989) 261–284.
- [50] J.W. Grzymala-Busse. LERS – a system for learning from examples based on rough sets. In: R. Slowinski (ed.), *Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer Academic Publishers, Dordrecht, Boston, London, 1992, pp. 3–18.
- [51] R.S. Michalski, I. Mozetic, J. Hong, and H. Lavrac. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In: *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, CA, August 11–15, 1986, pp. 1041–1045.
- [52] A. Skowron and C. Rauszer. The discernibility matrices and functions in information systems. In: R. Slowinski (ed.) *Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer Academic Publishers, Dordrecht, 1992, pp. 331–362.
- [53] J. Wróblewski. Covering with reducts – a fast algorithm for rule generation. In: *Proceedings of the First International Conference on Rough Sets and Current Trends in Computing*, Vol. 1424 of *Lectures Notes in Artificial Intelligence*, Warsaw, Poland, 1998. Springer-Verlag, pp. 402–407.
- [54] J.G. Bazan. Discovery of decision rules by matching new objects against data tables. In: *Proceedings of the First International Conference on Rough Sets and Current Trends in Computing*, Vol. 1424 of *Lectures Notes in Artificial Intelligence*, Warsaw, Poland, 1998. Springer-Verlag, pp. 521–528.
- [55] A. Skowron, J. Bazan, R. Latkowski, et al. *Rough Set Exploration System*. Institute of Mathematics, Warsaw University, Poland. <http://logic.mimuw.edu.pl/~rses>, accessed 2006.

50

Approximation Spaces in Conflict Analysis: A Rough Set Framework

Sheela Ramanna

50.1 Introduction

Rough set theory and the introduction of approximation spaces establish the foundation for granular computing and provide frameworks for perception and knowledge discovery in many areas [1]. One of the areas where information granulation is necessary is in conflict analysis and negotiation. Disputes and negotiations about various issues are commonplace in many organizations such as government and industry. To this end, many mathematical formal models of conflict situations have been proposed and studied, e.g., [2–10]. The approach used in this work is based on a different kind of relationship in the data. This relationship is not a dependency, but a conflict [11–16]. Formally, a conflict relation can be viewed as a special kind of discernibility, i.e., negation (not necessarily classical) of indiscernibility relation, which is the basis of rough set theory [17]. Thus, indiscernibility and conflict are closely related from a logical point of view. It is also interesting to note that almost all mathematical models of conflict situations are strongly domain dependent. Previous work on the application of rough sets to conflict resolution and negotiations between agents made it possible to introduce approximate reasoning about vague concepts [14]. Also, a granular approach to classifier construction for behavior pattern identification of complex objects can be found in [18].

Requirements interaction management (RIM) has become a critical area of requirements engineering. One of the key problems in RIM is a lack of systematic techniques for detecting and resolving conflicts [19]. There are two sources of conflicts: *social conflicts* where stakeholders have different viewpoints and *technical conflicts* where specification of requirements is inconsistent. Inconsistent requirements (technical conflicts) using classifiers based on rough sets can be found in [20]. Rough-set-based social conflict model involving high-level requirements negotiation was presented in [21]. This model was based on the win–win model¹ discussed in [22]. The problem of conflicts in software engineering has been studied extensively (see, e.g., [23–26]). Recent work in the application of rough sets to handling uncertainty in software engineering can be found in [24, 27, 28]. This chapter extends our earlier work

¹ See <http://sunset.usc.edu/research/WINWIN>.

on generalized conflict model with approximation spaces [29, 30] and analysis of conflict dynamics with risk patterns [31]. In this chapter, the focus is on a granular approach to representing requirements interaction and performing analysis of conflict dynamics to facilitate scope negotiation.

The contribution of this chapter is an *enhanced* conflict model based on rough sets for *requirements interaction* which encapsulates the two main sources of conflict. The enhancements include (i) a new technical model and (ii) the addition of a new decision attribute TC to the complex conflict model that captures trace-dependency information and the type of requirements. This is necessary because trace dependency is an indication of potential overlap between two requirements. The degree of overlap in turns aids in assessing the degree of technical conflict. Granular assessment of conflict dynamics is made possible by considering approximation spaces where the rough coverage function is used to measure the degree of conformity of sets of similar requirements to negotiation standards. Conflict graphs are used to analyze conflict situations, reason about the degree of conflict, and explore coalitions and discernibility degree. We illustrate our approach using a detailed set of requirements for an automated home lighting system. We have also suggested an advanced form of information granulation and adaptive learning based on near sets for modeling and reasoning about conflicts.

This chapter is organized as follows. A detailed discussion of the basic concepts of conflict and models of conflicts is given in Section 50.2. This discussion includes a formal model of rough-set-based basic and complex conflicts models. In Section 50.3, a discussion of conflicts in the context of requirements interaction is introduced. In addition, social and technical models for requirements interaction are presented. Section 50.3.4 presents a combined social and technical model in a rough set framework. An example that demonstrates the granular approach to assessing conflict dynamics with approximation spaces is discussed in Section 50.4.

50.2 Conflict Theory

The approach to conflicts used in this work is based not on functional-dependency between data, but on conflicts [14]. Formally, a conflict relation can be viewed as a special kind of discernibility, i.e., negation (not necessarily classical) of indiscernibility relation which is the basis of rough set theory [17, 32]. Thus, indiscernibility and conflict are closely related from a logical point of view. It is also interesting to note that almost all mathematical models of conflict situations are strongly domain dependent.

50.2.1 Concepts

The basic concepts of conflict theory that we use in this chapter are due to [14]. Let us assume that we are given a finite, non-empty set Ag called the *universe*. Elements of Ag will be referred to as *agents*. Let a *voting function* $v : Ag \rightarrow \{-1, 0, 1\}$, which is a number representing a voting result about some issue under negotiation. Numbers in $\{-1, 0, 1\}$ are interpreted as *against*, *neutral*, and *favorable*, respectively. The pair $CS = (Ag, V)$, where V is a set of voting functions, will be called a *conflict situation*.

To express relations between agents, we define three basic binary relations on the universe: *agreement*, *neutrality*, and *disagreement*. To this end, for a given voting function v , we first define the auxiliary function in (1).

$$\phi_v(ag, ag') = \begin{cases} 1, & \text{if } v(ag)v(ag') = 1 \text{ or } v(ag) = v(ag') = 0, \\ 0, & \text{if } v(ag)v(ag') = 0 \text{ and } \text{non}(v(ag)) = v(ag') = 0, \\ -1, & \text{if } v(ag)v(ag') = -1, \end{cases} \quad (1)$$

where $\phi_v(ag, ag') = 1$ means agents ag and ag' have the same opinion about an issue v (*agree* on issue v), $\phi_v(ag, ag') = 0$ means that at least one agent ag or ag' has no opinion about an issue v (is *neutral* on v), and $\phi_v(ag, ag') = -1$ means that both agents have different opinions about an issue v (are in *conflict* on issue v). Three basic relations R_v^+, R_v^0 , and R_v^- on Ag^2 called *agreement*, *neutrality*, and *disagreement*

relations, respectively, are defined in (2).

$$\begin{aligned} R_v^+(ag, ag') &\text{ iff } \phi_v(ag, ag') = 1, \\ R_v^0(ag, ag') &\text{ iff } \phi_v(ag, ag') = 0, \\ R_v^-(ag, ag') &\text{ iff } \phi_v(ag, ag') = -1. \end{aligned} \quad (2)$$

It is easily seen that the *agreement* relation has the properties listed in (3):

$$\begin{aligned} R_v^+(ag, ag). \\ R_v^+(ag, ag') &\text{ implies } R_v^+(ag', ag). \\ R_v^+(ag, ag') \text{ and } R_v^+(ag', ag'') &\text{ imply } R_v^+(ag, ag''). \end{aligned} \quad (3)$$

Hence, R_v^+ is an *equivalence* relation. Each equivalence class of the agreement relation will be called a *coalition* with respect to v . The conflict or disagreement relation has properties given in (4):

$$\begin{aligned} \text{Not } R_v^-(ag, ag). \\ R_v^-(ag, ag') &\text{ implies } R_v^-(ag', ag). \\ R_v^-(ag, ag') \text{ and } R_v^-(ag', ag'') &\text{ imply } R_v^+(ag, ag''). \\ R_v^-(ag, ag') \text{ and } R_v^+(ag', ag'') &\text{ imply } R_v^-(ag, ag''). \end{aligned} \quad (4)$$

For the neutrality relation, we have (5):

$$\begin{aligned} \text{not } R_v^0(ag, ag), \\ R_v^0(ag, ag') = R_v^0(ag', ag). \end{aligned} \quad (5)$$

Let us observe that in the conflict and neutrality relations there are no coalitions and they are not equivalence relations. Moreover, any distinct two of the three relations R_v^+ , R_v^0 , and R_v^- are pairwise disjoint, i.e., every pair of objects (ag, ag') belongs to exactly one of the above-defined relations (is in conflict/disagreement, agreement, or is neutral).

With every conflict situation $CS = (Ag, v)$, we will associate a *conflict graph*. In Figure 50.1, solid lines denote conflicts, dotted lines denote agreements, and for simplicity, neutrality is not shown explicitly in the graph. As one can see, agents B , C , and D form a coalition with respect to an issue v . A conflict degree $Con(CS)$ of the conflict situation $CS = (Ag, v)$ is defined by

$$Con(CS) = \frac{\sum_{\{(ag, ag'): \phi_v(ag, ag') = -1\}} |\phi_v(ag, ag')|}{2\lceil \frac{n}{2} \rceil \times (n - \lceil \frac{n}{2} \rceil)}, \quad (6)$$

where $n = Card(Ag)$. Observe that $Con(CS)$ is a measure of discernibility between agents from Ag relative to the voting function v .

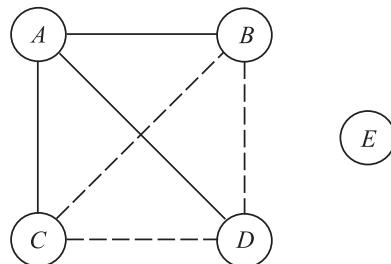


Figure 50.1 Exemplary conflict graph

For a more general conflict situation, $CS = (Ag, V)$, where $V = \{v_1, \dots, v_k\}$, is a finite set of voting functions each for a different issue. The *conflict degree* in CS (*tension generated by V*) can be defined by (7).

$$Con(CS) = \frac{\sum_{i=1}^k Con(CS_i)}{k}. \quad (7)$$

50.2.2 Rough-Set-Based Basic Conflict Model

Information systems provide a practical basis for expressing certain basic concepts of rough set theory. This is the starting point for our basic conflict model. In a rough set approach to conflict analysis, an information system is represented by a table containing rows labeled by *objects (agents)* and columns by *attributes (issues)*. The entries of the table are *values of attributes (votes)*, which are uniquely assigned to each agent and attribute; i.e., each entry corresponds to a row x and column a , representing opinion of an agent x about issue a . Formally, an *information system* can be defined as a pair $S = (U, A)$, where U is a non-empty, finite set called the *universe* (elements of U are called *objects*) and A is a non-empty, finite set of *attributes* [17].

Every attribute $a \in A$ is a total function $a : U \rightarrow V_a$, where V_a is the set of *values* of a , called the *domain* of a . Elements of V_a will be referred to as *opinions*, and $a(x)$ is the opinion of an agent x about issue a . Although the above given definition is general, for conflict analysis we will need its simplified version, where the domain of each attribute is restricted to three values only, i.e., $V_a = \{-1, 0, 1\}$, for every a , meaning *disagreement*, *neutral*, and *agreement* respectively. For the sake of simplicity, we will assume $V_a = \{-, 0, +\}$. Every information system with the above-mentioned restriction will be referred to as a *situation*.

We now observe that any conflict situation $CS = (Ag, V)$ can be treated as an information system where $Ag = \{ag_1, \dots, ag_n\}$ and $V = \{v_1, \dots, v_k\}$ with the set of objects Ag (*agents*) and the set V of negotiation *issues*. In other words, a conflict situation CS is the *basic conflict model* which permits simple, yet powerful, assessment of degree of conflict among agents (see Section 50.4). A discussion of exemplary conflict situations can be found in [21, 29, 30].

50.2.3 Rough-Set-Based Complex Conflict Model

Decision systems provide yet another practical and powerful framework for representing knowledge, particularly for discovering patterns in objects and in classification of objects into decision classes. In addition, decision systems also make it possible to represent and reason about objects at a higher level of granularity. This is necessary for modeling complex conflict situations. The extension of the basic model of conflict to a decision system with complex decisions was introduced in [29]. We recall some basic assumptions that agents in the complex conflict model are represented by conflict situations $CS = (Ag, v)$, where Ag is the set of lower level agents and v is a voting function defined on Ag for $v \in V$. Hence, agents in the complex conflict model are related to groups of lower level agents linked by a voting function. The voting functions in the complex conflict models are defined on such conflict situations. The set of the voting functions for the complex conflict model is denoted by A . In this, way we obtain an information system (U, A) , where U is the set of situations. Observe that any situation $CS = (Ag, v)$ can be represented by a matrix

$$[v(ag)]_{ag \in Ag}, \quad (8)$$

where $v(ag)$ is the result of voting by the agent $ag \in Ag$. We can extend the information system (U, A) to the decision system (U, A, d) , assuming that $d(s) = Con(CS_v)$ for any $CS = (Ag, v)$. For the constructed decision system (U, A, d) , one can measure discernibility between compound decision values which correspond to conflict situations in the constructed decision table [32]. The reducts of this decision table

relative to decision have a natural interpretation with respect to conflicts. An illustration of conflict analysis with similarity relation can be found in [31].

50.3 Conflicts and Requirements Engineering

Requirements engineering provides the appropriate mechanism for understanding what the customer wants, analyzing the need, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification, and managing the requirements as they are transformed into an operational system [33]. Consequently, requirements engineering research spans a wide range of topics, but a topic of increasing importance is the analysis and management of dependencies (relationships) among requirements also known as RIM [19].

50.3.1 Requirements Interaction

Understanding requirements conflict is one of the objectives of RIM. There are two sources of requirements conflicts. Social conflicts are due to the fact that development of complex software systems involves a collaborative process of requirements identification through negotiation by stakeholders. Technical conflicts arise due to inconsistent or contradictory specification of requirements. These two sources of conflicts are intertwined. That is, inconsistent requirements often reflect the inconsistent need of stakeholders. One of the key problems in requirements engineering is a lack of systematic techniques for detecting and resolving conflicts.

50.3.2 Model for Social Conflicts

The complex conflict model (U, A, d) described in Section 50.2.3 serves as a framework of representing social conflicts where subjective views of stakeholders can be represented in the form of voting. We also assume that $d(s) = \text{Con}(CS_v)$ for any $CS = (Ag, v)$. As a result, the universe Ag will now consist of SH (set of stakeholders) and the *voting function* $v : SH \rightarrow \{-1, 0, 1\}$, which is a number representing the voting result about some issue about requirements under negotiation. So a conflict (social) situation in the context of requirements determination is formally represented as $CS = (SH, V)$, where $SH = \{sh_1, \dots, sh_n\}$ and $V = \{v_1, \dots, v_k\}$. Let V denote the set of scope negotiation parameters. The conflict situation can now be interpreted as opinions held by two or more stakeholders about requirements that cause an inconsistency. The model for social conflicts has been used to achieve consensus on the high-level requirements (for details, see [21]).

50.3.3 Model for Technical Conflicts

Since technical conflicts arise due to contradictory specifications of requirements, we now introduce a new model to handle the representation of requirements conflicts. Requirements *conflict* with each other when they make contradictory statements about common software attributes, and they *cooperate* when they mutually enforce such attributes [34]. These software attributes are generally referred to as persistent or non-functional attributes and include quality attributes such as efficiency, reliability, scalability, usability, and security, to name a few.

Formally, a model for technical conflicts can be represented as a *decision system* defined as $TCS = (U, B, ri)$, where U is a non-empty, finite set called the *universe* (elements of U are called *requirements*), and B is a non-empty, finite set of *conflict attributes* and ri represents requirement interaction degree with the restriction $B = \{\text{Type}, \text{Degree of Overlap}, \text{Artifact}\}$. Every attribute $a \in B$ is a total function $a : U \rightarrow V_a$, where V_a is the set of *values* of a , called the *domain* of a . Although the above given

definition is general, for technical conflicts, we also need to restrict the domain of each attribute as follows:

- $V_{Type} = \{FR, ER, UR, RR, SR, RCR, AR, MR\}$ means functionality, efficiency, usability, reliability, security, recoverability, accuracy, and maintainability, respectively indicating the type of requirement.
- $V_{DegreeofOverlap} = [0, 1]$.
- $V_{Artifact} = \{R_1, \dots, R_k\}$.
- $V_{ri} = \{SC, WC, VWC, NC\}$ represents the degree of conflict: strong conflict, weak conflict, very weak conflict, and no conflict, respectively.

The model is designed to capture information from requirements traceability² [35]. Requirements traceability involves defining and maintaining relationships with artifacts created as a part of systems development, such as architectural designs, requirements, and source code, to name a few. In this chapter, we restrict the *artifact information* to other requirements as an aid to identifying conflicting (or cooperating) requirements [36]. The exemplary domains for degree of overlap and conflict degrees are due to [34]. In this chapter, we assume that an automated requirements traceability tool makes it possible to automatically extract (i) conflicts and cooperation information among requirements and (ii) trace dependencies. The degree of overlap between requirements and the conflict degrees is to a large extent manually assessed.

50.3.4 Enhanced Model – Combining Social and Technical Conflicts

We now introduced an enhanced model based on the complex conflict model described in Section 50.2.3, which captures both social and technical conflict information. Recall (i) that the decision system (U, A, d) models social conflicts in d which represents the conflict degree as a result of voting by agents (stakeholders) on issue v , where $d(s) = Con(CS_v)$ for any $CS = (Ag, v)$, and (ii) the decision system (U, B, ri) , models technical conflicts in ri , where $ri(R) \in V_{ri}$ is the degree of conflict for any requirement $R \in U$. The enhanced model is another decision system (U, A, d, ri) , where elements of U are requirements and C is a non-empty, finite set of requirements scope negotiation parameters, decision d represents social conflicts, and decision ri represents requirements interaction (technical conflicts). Note that whereas d is computed, ri is assessed subjectively based on overlap and the type of requirement. What follows is an illustration of the conflict models.

50.4 Granular Conflict Dynamics Assessment

Information granulation is useful for solving complex problems such as conflicts. Information granules can be treated as linked collections (clumps) of objects drawn together by the criteria of indiscernibility, similarity, or functionality [37]. A granule can be a feature, decision rules or sets of decision rules, classifiers, or approximation spaces [38, 39]. In this section, we illustrate construction and reasoning with compound conflict granules.

50.4.1 Example

Achieving consensus on a detailed set of requirements for each high-level requirement that was agreed by all stakeholders is a complex collaborative process. This is a crucial step as it determines the scope of the project. In this section, we demonstrate how the enhanced conflict model can be used to assess conflict dynamics at different levels of granularity which would aid in scope negotiation. We will focus on the detailed set of requirements for a single high-level requirement (R_1 – custom lighting scenes). A complete example of the problem of achieving agreement on high-level system requirements for a home

² IEEE Std. 830-1998.

Table 50.1 Social conflict model

Negotiation parameters						
R_1	<i>Effort</i>	<i>Importance</i>	<i>Stability</i>	<i>Risk</i>	<i>Testability</i>	<i>Conflict Degree</i>
$r_{1.1}$	M	H	N	L	Y	0.22
$r_{1.2}$	M	H	N	L	Y	0.44
$r_{1.3}$	H	M	N	M	Y	0.2
$r_{1.4}$	L	H	Y	L	Y	0.0
$r_{1.5}$	M	H	P	H	Y	0.67
$r_{1.6}$	M	L	P	H	N	0.89

lighting automation system (HLAS) described in [40] can be found in [21]. Assume that R_1 includes the following specifications (objects):

- $r_{1.1}$ – ability to control up to a maximum of 20 custom lighting scenes,
- $r_{1.2}$ – each scene provides a preset level of illumination (within 5 s) for each lighting bank,
- $r_{1.3}$ – maximum range of a scene is 20 m,
- $r_{1.4}$ – activated using control switch only,
- $r_{1.5}$ – activated within 3 s using central control unit,
- $r_{1.6}$ – ability to control an additional two lighting scenes in the yard.

R_1 – Social Conflict Model

The social conflict model (U, A, d) consists of a set A of negotiation parameters and a decision attribute d denoting a complex decision. The condition attributes are scope negotiation parameters assessed by the development team and the decision attribute is a compound decision denoting the conflict degree based on subjective opinions of stakeholders. We consider the following negotiation parameters:

- *Effort* which is a rough estimate of development effort (high, medium, or low);
- *Importance* which determines whether a requirement is essential to the project (high, medium, or low);
- *Stability* of a requirement which indicates its volatility (yes, perhaps, or no);
- *Risk* which indicates whether the requirement is technically achievable (high, medium, or low);
- *Testability* which indicates whether a requirement is testable (yes or no).

Table 50.1 is an illustration of the complex conflict model for the HLAS requirement R_1 described in Section 50.2.3, which forms the social conflict model in the context of requirements interaction. Table 50.2 is a partial illustration of the basic conflict model for the HLAS requirement R_1 and is the source for computation of the conflict degree defined in (6). We can now compute the conflict degree for $CS_{r1.1} = (SH, V)$, using equation 7, where $V = \{Priority, Effort, Risk\}$, $Con(CS_{r1.1}) = 0.22$

Table 50.2 Requirement $r_{1.1}$ – Basic conflict model

Voting results			
<i>Stakeholder</i>	<i>Priority</i>	<i>Effort</i>	<i>Risk</i>
sh_1	+	+	0
sh_2	0	-	0
sh_3	+	0	-
sh_4	+	+	+
sh_5	+	0	-

Table 50.3 Technical conflict model

Requirement interaction parameters				
R_1	Type	Degree of Overlap	Artifact	ri
$r_{1.1}$	FR	0.7	$r_{1.6}$	WC
$r_{1.2}$	ER	0.6	$r_{1.5}$	VWC
$r_{1.3}$	FR	0.0	—	NC
$r_{1.4}$	FR	0.8	$r_{1.5}$	SC
$r_{1.5}$	ER	1.0	$r_{1.4}$	WC
$r_{1.6}$	FR	0.5	$r_{1.1}$	VWC

with $Con((SH, Priority)) = 0$, $Con((SH, Effort)) = 0.33$, and $Con((SH, Risk)) = 0.33$. Note that + indicates the highest level of support, — indicates the lowest level of support, and 0 indicates the intermediate level of support. For instance, for the issue *Priority*, + means critical, — means useful, and 0 means important. Voting for the remaining requirements $r_{1.2} \dots r_{1.6}$ is performed in a similar manner.

R₁ – Technical Conflict Model

Recall that the model for technical conflicts is defined as $TCS = (U, B, ri)$, where U is a non-empty, finite set of requirements and B is a non-empty, finite set of *conflict attributes*, and ri represents requirement interaction degree. The assessment of interaction degree follows the approach specified in [34]. Briefly, the approach is based on a generic model of potential conflict and cooperation which highlights the nature of added requirements on other attributes of the system. For example, if a requirement adds *new functionality* to the system, it may have (i) no effect (0) on the overall *functionality*, (ii) negative effect (−) on *efficiency*, (iii) positive effect (+) on *usability*, (iv) negative effect (−) on *reliability*, (v) negative effect (−) on *security*, (vi) no effect (0) on *recoverability*, (vii) no effect (0) on *accuracy*, and (viii) no effect (0) on *maintainability*. This model is very general and is a worst best-case scenario. In practice, one must take into account the degree of overlap between requirements and the type of requirement since it has a direct bearing on the degree of conflict or cooperation. Trace dependencies based on scenarios and observations are used to arrive at the degree of overlap [36].

Table 50.3 captures requirements interaction based on its impact on software attributes. For example, $r_{1.4}$ adds functionality to the system where activation is through control switch only and it conflicts with $r_{1.5}$ which is an efficiency requirement with a 3-s. activation specification. In addition, $r_{1.4}$ is a subset of $r_{1.5}$ in terms of functionality. Hence it is a strong conflict. In general, it is possible to define a look-up table for project teams to determine the requirements interaction [34]. Alternatively, ri can be defined in terms of a requirements interaction function:

$$ri_R(R, R') = SC \text{ if } o(R, R') = 1 \text{ and } R = FR \text{ and } R' = ER, \quad (9)$$

where $ri_R(R, R') = SC$ means that there is strong conflict in the case where there is 100% overlap between functionality and efficiency requirements. Note that for a subset overlap, it is not always the case that $ri_R(R, R') = ri_{R'}(R', R)$.

R₁ – Enhanced Model

Recall that the enhanced model is another decision system (U, A, d, ri) , where elements of U are requirements and C is a non-empty, finite set of requirements scope negotiation parameters, decision d represents social conflicts, and decision ri represents requirements interaction (technical conflicts). Table 50.4 represents the enhanced model with two decision attributes.

Observation 1. Nature of Conflicts in Large-Scale Projects

The underlying philosophy in the enhanced sociotechnical model is that requirements for harmony in social interaction among project stakeholders and in the expression of technical specifications inevitably lead to conflicts that must be resolved during negotiation. Although it is fairly obvious that social

Table 50.4 Sample enhanced model

Social and technical conflicts							
<i>R</i> ₁	<i>Effort</i>	<i>Importance</i>	<i>Stability</i>	<i>Risk</i>	<i>Testability</i>	<i>SC</i>	<i>TC</i>
<i>r</i> _{1.1}	M	H	N	L	Y	0.22	WC
<i>r</i> _{1.2}	M	H	N	L	Y	0.44	VWC
<i>r</i> _{1.3}	H	M	N	M	Y	0.2	NC
<i>r</i> _{1.4}	L	H	Y	L	Y	0.0	SC
<i>r</i> _{1.5}	M	H	P	H	Y	0.67	WC
<i>r</i> _{1.6}	M	L	P	H	N	0.89	VWC

interaction of stakeholders and the need for consistency in technical specifications is always present (inherent) during large-scale projects, government programs, and military offense or defense planning, the social and technical components are seldom considered together in the analysis of conflicts. For simplicity, we refer to only large-scale projects in this chapter. The kernel of the enhanced model is a synthesis of the sociotechnical negotiation parameters to facilitate study of the incipient degree of conflict. The basic approach in the enhanced model is to suggest a framework that makes sense of the Weltanschauung (worldview) implicit in social and technical requirements for a project. It should also be observed that resolution of sociotechnical conflicts can be aided by a fusion of the approximation spaces that takes advantage of the near-set approach to perceptual synthesis [41–45]. The use of near sets in conflict resolution is part of ongoing research but is outside the scope of this chapter.

50.4.2 Assessment with Approximation Spaces

In reasoning about conflicts, it is necessary to view objects (requirements) in terms of more compound granules such as approximation spaces. Assessment of conflict dynamics with approximations spaces for social conflicts were first introduced in [29] and later elaborated in [30]. In this chapter, the focus is on the impact of both social and technical conflicts. Generalized approximation spaces were introduced in [46]. Let $DS = (U_{req}, A, d, ri)$. For any Boolean combination of descriptors over DS and α , the semantics of α in DS is denoted by $\|\alpha\|_{DS}$, i.e., the set of all objects from U satisfying α [17]. Briefly, a generalized approximation space $GAS = (U_{req}, N_B, v_B)$, where for any objects $r \in U$ the neighborhood $N_B(r)$ is defined by

$$N_B(r) = \|\bigwedge_{a \in B} (a = a(r))\|_{DS}, \quad (10)$$

and the coverage function v_B is defined by

$$v_B(X, Y) = \begin{cases} \frac{|X \cap Y|}{|Y|}, & \text{if } Y \neq \emptyset, \\ 1, & \text{if } Y = \emptyset, \end{cases} \quad (11)$$

where $X, Y \subseteq U$. This form of specialization of a GAS is called a *lower approximation space* [47]. Assuming that the lower approximation $B_* D_i$ represents an acceptable level and we are interested in the values

$$v_B(N_B(r), B_* D_L) \quad (12)$$

of the coverage function specialized in the context of a decision system DS for the neighborhoods $N_B(r)$ and the acceptable level $B_* D_L$ for conflict negotiation. Note that D_i denotes the i th decision either for d or for ri ; i.e., $D_i = \{u \in U_{req} : d(u) = i \text{ or } ri(u) = i\}$, which is a set of requirements from U_{req} with conflict level i . Let us assume that values for SC in Table 50.4 are classified as follows: *L* (*low conflict degree* ≤ 0.3), *M* ($0.3 \leq \text{medium conflict degree} \leq 0.7$), and *H* (*conflict degree* > 0.7).

Let ξ_B denote the partition of a set of objects relative a set of functions B representing object features. We can construct a compound granule (lower approximation space) relative to SC with elementary granules (scope negotiation parameter B set to importance) as follows:

$$\begin{aligned} B &= \{\text{Importance}\}, \\ \xi_B &= [r_{1.1}] \cup [r_{1.3}] \cup [r_{1.6}], = \{r_{1.1}, r_{1.2}, r_{1.4}, r_{1.5}\} \cup \{r_{1.3}\} \cup \{r_{1.6}\}, \\ D_L &= \{r \in U : d(r) = L\} = \{r_{1.1}, r_{1.3}, r_{1.4}\}, \\ N_B(r_{1.1}) &= \{r_{1.1}, r_{1.2}, r_{1.4}, r_{1.5}\}, \\ N_B(r_{1.3}) &= \{r_{1.3}\}, \\ N_B(r_{1.6}) &= \{r_{1.6}\}, \\ B_*D_L &= \{r_{1.3}\}, \\ v_B(N_B(r_{1.1}), B_*D_L) &= 0.0, \\ v_B(N_B(r_{1.3}), B_*D_L) &= 1.0, \\ v_B(N_B(r_{1.6}), B_*D_L) &= 0.0. \end{aligned}$$

Note that the chosen level of social conflict is *low*. Based on the experimental rough coverage values, if we set a threshold tr for acceptance as 0.5 such that $v_B(N_B(r), B_*D_L) > tr$, then requirement $r_{1.3}$ would have the lowest level of social conflict and also be classified with certainty. Based on the rough coverage value, we now have to look at the technical conflict level for $r_{1.3}$. In this particular example, it so happens that this requirement is not inconsistent with any other requirement. So $r_{1.3}$ can be included with certainty as a part of the overall system requirements. In addition, $r_{1.1}$ and $r_{1.4}$ are of *high importance* and *low social conflict*. In this case, $r_{1.1}$ *weakly conflicts* with requirement $r_{1.6}$ and $r_{1.4}$ *strongly conflicts* with requirement $r_{1.5}$. This means that $r_{1.1}$ may have to be modified slightly. However, requirements $r_{1.4}$ and $r_{1.5}$ will have to be respecified and their impact assessed before they are included in the overall requirements.

Observation 2. Information Granulation in Conflict Assessment

Observe that information granulation starts with the B -partition ξ_B of a set of sample objects X gathered during negotiation. Each partition consists of equivalence classes called elementary sets that constitute fine-grained information granulation, where each elementary set contains objects with matching descriptions. For example, from Table 50.4, we obtain three elementary sets, namely,

$$[r_{1.1}], [r_{1.3}], [r_{1.6}].$$

The elementary sets in a partition are used to construct complex information granules such as the lower approximation and upper approximation of a decision class that reflects a perception of the the sample objects relative to a concept important in conflict resolution. For example, if we consider information granulation relative to low conflict degree L for the decision class D_L , we obtain the lower approximation

$$B_*D_L = \{[r_{1.3}]_B\} = \{r_{1.3}\}.$$

The information granule represented by B_*D_L can be used as a sort of benchmark in evaluating the degree of conformity of each elementary granule to the benchmark (see, e.g., [48]).

Observation 3. Adaptive Learning as a Means of Conflict Resolution

It has been shown that it is possible to combine near sets and approximate adaptive learning [41, 49]. This is a part of the basic approach to an advanced form of information granulation suggested in Observation 1. Implicit in this approach is a hierarchy of approximation spaces, namely, GAS_{SC} , GAS_{TC} synthesized in a nearness approximation space $NAS_{SC,TC}$ as shown in Figure 50.2a. This leads to the

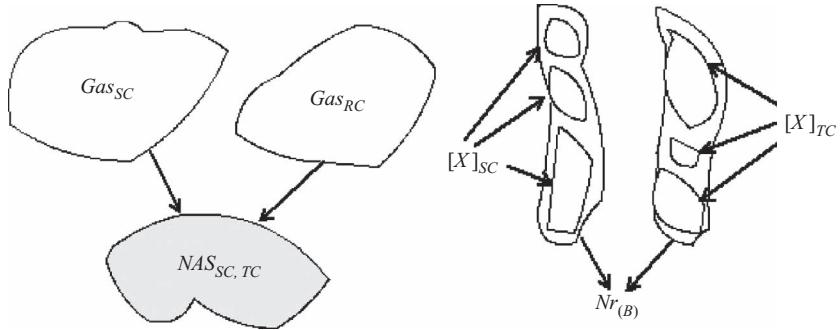


Figure 50.2 Near-set approach to information granulation: (a) perceptual synthesis and (b) family of neighborhoods.

construction of a family of neighborhoods $N_r(B)$ like the one shown in Figure 50.2b, where

$$N_r(A) = \bigcup_{B \subseteq P_r(A)} [x]_B,$$

where $P_r(A) = \{B \subseteq A \mid |B| = r\}$ for any r , such that $1 \leq r \leq |A|$. That is, r denotes the number of features used to construct families of neighborhoods. For the sake of clarity, we sometimes write $[x]_{B_r}$ to specify that the equivalence class represents a neighborhood formed using r features from B . Families of neighborhoods are constructed for each combination of probe functions in B using $\binom{|B|}{r}$, i.e., $|B|$ probe functions taken r at a time. Information about a sample $X \subseteq U$ can be approximated from information contained in B by constructing a $N_r(B)$ -lower approximation

$$N_r(B)_* X = \bigcup_{x: [x]_{B_r} \subseteq X} [x]_{B_r},$$

and a $N_r(B)$ -upper approximation

$$N_r(B)^* X = \bigcup_{x: [x]_{B_r} \cap X \neq \emptyset} [x]_{B_r}.$$

Then, $N_r(B)_* X \subseteq N_r(B)^* X$ and the boundary region $BND_{N_r(B)}(X)$ between upper and lower approximations of a set X is defined to be the complement of $N_r(B)_* X$; i.e.,

$$BND_{N_r(B)}(X) = N_r(B)^* X \setminus N_r(B)_* X = \{x \in N_r(B)^* X \mid x \notin N_r(B)_* X\}.$$

A set X is termed a ‘near set’ relative to a chosen family of neighborhoods $N_r(B)$ iff $|BND_{N_r(B)}(X)| \geq 0$. This approach to information granulation has led to the introduction of a family of approximate adaptive learning algorithms [49] that have recently been implemented in two different biologically inspired systems representing swarms of organisms that learn to cooperate and survive using the delayed reward (reinforcement) approach to learning. The basic idea, here, is to define states, actions, and rewards in the context of conflict negotiation. Briefly, a system state s results from a chosen action. A wide spectrum of possible actions can be identified in the context of negotiation. For example, let

$$A = \{a \mid a = \text{negotiation action}\},$$

where a negotiation action might be compromise, bid, withdraw, substitute, and so on. In general, a reward r is a mapping from a set of objects X to \mathfrak{M} (set of reals). The definition of r in the context of

conflict resolution flows naturally from what the computation of coverage values using average coverage

$$\overline{v_B([x]_B, N_r(B)_*D)}.$$

Let th denote a threshold for acceptable average coverage values. Then $r(x)$ is computed in the following way:

$$r(x) = \begin{cases} 1, & \text{if } \overline{v([x]_B, N_r(B)_*D)} \geq th, \\ 0, & \text{otherwise.} \end{cases}$$

The intent here is to suggest the possibility of using near-set-based adaptive learning as a direct means of conflict resolution.

50.5 Conclusion

This chapter introduces an *enhanced* conflict model based on rough sets for *requirements interaction*, which encapsulates both social conflicts (stakeholder viewpoints) and technical conflicts (requirements interaction). The approximation-space-based framework inherent in the proposed research makes it possible to represent requirements (both functional and nonfunctional) and their attributes, identifying technical conflicts and coalitions (cooperation) and incorporating trace-dependency information. The granular-based approach is important because it offers the ability to reason about conflict dynamics based on various criteria. The proposed research points to a means of simplifying the representation of technical and social conflicts. Also, more in-depth analysis of complex conflicts situations can be achieved using risk patterns where one can measure deviations of the conflict degree among social viewpoints as well as technical viewpoints. This can be done with distance reducts extracted from conflict data using Boolean reasoning [50–52]. This model also makes it possible for assessing technical conflict dynamics with a discernibility degree measure. It has been suggested that it is possible to combine near sets and approximate adaptive learning in conflict resolution.

Acknowledgments

The author gratefully acknowledges Andrzej Skowron and James F. Peters for their insightful comments. The author's research is supported by NSERC Canada grant 194376 .

References

- [1] J.F. Peters and A. Skowron. *Zdzislaw Pawlak life and work (1926–2006)*. *Inf. Sci.* 177 (2007) 1–2.
- [2] J.L. Casti. *Alternative Realities – Mathematical Models of Nature and Man*. John Wiley and Sons, New York, 1989.
- [3] C.H. Coombs and G.S. Avrunin. *The Structure of Conflict*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1988.
- [4] R. Deja and A. Skowron. On some conflict models and conflict resolutions. *Rom. J. Inf. Sci. Technol.* 5(1–2) (2002) 69–82.
- [5] R. Kowalski. *A Logic-Based Approach to Conflict Resolution*, 2003, pp. 1–28 [manuscript].
- [6] S. Kraus. *Strategic Negotiations in Multiagent Environments*. The MIT Press, Cambridge, MA, 2001.
- [7] G. Lai, C. Li, K. Sycara, and J. Giampapa. *Literature Review on Multi-attribute Negotiations*. Technical Report CMU-RI-TR-04-66. Carnegie Mellon University, Pittsburg, 2004, pp. 1–35.
- [8] Y. Maeda, K. Senoo, and H. Tanaka. *Interval Density Function in Conflict Analysis*, LNAI 1711. Springer-Verlag, Berlin, 1999, pp. 382–389.
- [9] A. Nakamura. Conflict logic with degrees. In: S.K. Pal and A. Skowron (eds), *Rough Fuzzy Hybridization: A New Trend in Decision-Making*. Springer-Verlag, Berlin, 1999, pp. 136–150.
- [10] Z. Pawlak. An inquiry into anatomy of conflicts. *J. Inf. Sci.* 109 (1998) 65–78.
- [11] Z. Pawlak. On conflicts. *Int. J. Man-Mach. Stud.* 21 (1984) 127–134.

- [12] Z. Pawlak. *On conflicts* (in Polish). Polish Scientific Publishers, Warsaw, 1987.
- [13] Z. Pawlak. Anatomy of conflict. *Bull. Eur. Assoc. Theor. Comput. Sci.* 50 (1993) 234–247.
- [14] Z. Pawlak and A. Skowron. Rough sets and conflict analysis. In: *E-Service Intelligence – Methodologies, Technologies and Applications, Book series on Computational Intelligence*. Springer-Verlag, Berlin, 2006 [to appear].
- [15] Z. Pawlak and A. Skowron. Rough sets: Some extensions. *Inf. Sci.* 177 (2007) 28–40.
- [16] K. Sycara. Multiagent systems. *AI Mag.* 19 (Summer 1998) 79–2.
- [17] Z. Pawlak. *Rough Sets – Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.
- [18] J. Bazan and A. Skowron. Classifiers based on approximate reasoning schemes. In: B. Dumit-Keplicz, A. Jankowski, A. Skowron and M. Szczuka (eds), *Monitoring, Security, and Rescue Tasks in Multiagent Systems MSRAS, Advances in Soft Computing*. Springer, Heidelberg, 2005, pp. 191–202.
- [19] W.N. Robinson, D.S. Pawłowski, and V. Volkov. Requirements interaction management. *ACM Comput. Surv.* 35(2) (2003) 132–190.
- [20] Z. Li and G. Ruhe. Uncertainty handling in tabular-based requirements using rough sets. In: D. Ślęzak, J.T. Yao, J.F. Peters, W. Ziarko, and X. Hu (eds), *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, LNAI 3642. Springer-Verlag, Berlin, 2005, pp. 678–687.
- [21] A. Skowron, S. Ramanna, and J.F. Peters. Conflict analysis and information systems: A rough set approach. In: G. Wang, J.F. Peters, A. Skowron, and Y.Y. Yao (eds), *Proceedings of the International Conference on Rough Sets and Knowledge Technology (RSKT 2006)*, LNAI, Chongqing, China, July 24–26, 2006, LNCS 4062. Springer-Verlag, Heidelberg, 2006, pp. 233–240.
- [22] B. Boehm, P. Grünbacher, and J. Kepler. *Developing Groupware for Requirements Negotiation: Lessons Learned*. IEEE Software, NJ, 2001, pp. 46–55.
- [23] T. Cohene and S. Easterbrook. Contextual risk analysis for interview design. In: *Proceedings of 13th IEEE International Requirements Engineering Conference (RE'05)*, Paris, France, 2005, pp. 1–10.
- [24] B. Curtis, H. Krasner, and N. Iscoe. A field study of the software design process for large systems. *Commun. ACM* 31(11) (1988) 1268–1287.
- [25] S. Easterbrook. Handling conflict between domain descriptions with computer-supported negotiation. *Knowl. Acquis. Int. J.* 3(1991) 255–289.
- [26] A. Finkelstein, M. Goedicke, J. Kramer, and C. Niskier. Viewpoint oriented software development: Methods and viewpoints in requirements engineering. In: J. Bergstra and L. Feijen (eds), *Methods for Formal Specification*, LNCS 490. Springer-Verlag, Heidelberg, 1989, pp. 29–54.
- [27] J.F. Peters and S. Ramanna. Approximation space for software models. In: J.F. Peters and A. Skowron (eds), *Transactions on Rough Sets I*, LNCS 3100. Springer-Verlag, Heidelberg, 2004, pp. 338–354.
- [28] J.F. Peters and S. Ramanna. Towards a software change classification system: A rough set approach. *Softw. Qual. J.* 11 (2003) 121–147.
- [29] S. Ramanna, J.F. Peters, and A. Skowron. Generalized conflict and resolution model with approximation spaces. In: *Proceedings of RSCTC 2006*, Kobe, Japan, November 2006, Lecture Notes in Artificial Intelligence, 4259. Springer-Verlag, Heidelberg, 2006, pp. 274–283.
- [30] S. Ramanna, J.F. Peters, and A. Skowron. Approaches to conflict dynamics based on rough sets. *Fundam. Inf.* 75 (2006) 1–16.
- [31] S. Ramanna, J.F. Peters, and A. Skowron. Analysis of conflict dynamics by risk patterns. In: H.-D. Burkhard, L. Czaja, W. Penczek, A. Salwicki, A. Skowron, and Z. Suraj (eds), *Proceedings of the 15th International Workshop on Concurrency, Specification, and Programming (CSP 2006)*, Wenditz, Germany, September, 27–29, 2006, Informatik-Berichte 206. Humboldt Universität, Berlin, 2006, pp. 469–479.
- [32] Z. Pawlak and A. Skowron. Rough sets and boolean reasoning. *Inf. Sci.* 177 (2007) 41–73.
- [33] R.H. Thayer and M. Dorfman. *Software Requirements Engineering*. IEEE Computer Society Press, CA, 1997.
- [34] A. Egyed and P. Grünbacher. *Identifying Requirements Conflicts and Cooperation: How Quality Attributes and Automated Traceability Can Help*. IEEE Software, NJ, 2004, pp. 50–58.
- [35] J.F. Peters and W. Pedrycz. *Software Engineering: An Engineering Approach*. John Wiley and Sons, New York, 2000.
- [36] A. Egyed and P. Grünbacher. Automating requirements traceability, beyond the record and replay paradigm. In *Proceedings of the 17th International Conference on Automated Software Engineering*, EEE CS Press, CA, 2002, pp. 163–171.
- [37] L.A. Zadeh. Toward a theory of fuzzy information granulation and its certainty in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* 90 (1997) 111–127.
- [38] J.F. Peters, A. Skowron, P. Synak, and S. Ramanna. Rough sets and information granulation. In: T. Bilgic, D. Baets, and O. Kaynak (eds), *Proceedings of IFSA LNAI 2715*. Springer-Verlag, Berlin, 2003, pp. 370–377.

- [39] A. Skowron and J. Stepaniuk. Information granules. *Int. J. Intell. Syst.* 16(1) (2001) 57–86.
- [40] D. Lefringwell and D. Widrig. *Managing Software Requirements*. Addison-Wesley, MA, 2003.
- [41] J.F. Peters, C. Henry, and D.S. Gunderson. Biologically-inspired approximate adaptive learning control strategies: A rough set approach. *Int. J. Hybrid Intell. Syst.* 4(4) (2007) 203–267.
- [42] J.F. Peters, A. Skowron, and J. Stepaniuk. Nearness of objects: Extension of approximation space model. *Fundam. Inf.* 76 (2007) 1–24.
- [43] J.F. Peters, A. Skowron, and J. Stepaniuk. Nearness in approximation spaces. In: G. Lindemann, H. Schliengloff, H.-D. Burkhard, L. Czaja, W. Penczek, A. Salwicki, A. Skowron, Z. Suraj (eds), *Proceedings of Concurrency, Specification & Programming (CS&P'2006)*. Informatik-Berichte Nr. 206. Humboldt-Universität zu Berlin, 2006, pp. 434–445.
- [44] J.F. Peters. Near sets. Special theory about nearness of objects. *Fundam. Inf.* 75(1–4) (2007) 407–433.
- [45] J.F. Peters. Near sets. Toward approximation space-based object recognition. In: J.T. Yao, P. Lingras, W.-Z. Wu, M. Szczuka, N. Cercone, and D. Ślezak (eds), *Proceedings of the Second International Conference on Rough Sets and Knowledge Technology (RSKT07), Joint Rough Set Symposium (JRS07), Lecture Notes in Artificial Intelligence 4481*. Springer-Verlag, Berlin, 2007, pp. 22–33.
- [46] A. Skowron and J. Stepaniuk. Generalized approximation spaces. In: T.Y. Lin and A.M. Wildberger (eds), *Soft Computing*. Simulation Councils, San Diego, 1995, pp. 18–21; see also Tolerance approximation spaces. *Fundam. Inf.* 27(2–3) (1996) 245–253.
- [47] J.F. Peters and C. Henry. Reinforcement learning with approximation spaces. *Fundam. Inf.* 71 (2006) 323–349.
- [48] S. Ramanna, J.F. Peters, and A. Skowron. Approximation space-based socio-technical conflict model. In: J.T. Yao, P. Lingras, W.-Z. Wu, M. Szczuka, N. Cercone, and D. Ślezak (eds), *Proceedings of the Second International Conference on Rough Sets and Knowledge Technology (RSKT07), Joint Rough Set Symposium (JRS07), Lecture Notes in Artificial Intelligence 4481*. Springer-Verlag, Berlin, 2007, pp. 476–483.
- [49] J.F. Peters. Toward approximate adaptive learning. In: *International Conference on Rough Sets and Emerging Intelligent Systems Paradigms in Memoriam Zdzisław Pawlak, Lecture Notes in Artificial Intelligence 4585*. Springer-Verlag, Berlin, Heidelberg, 2007, pp. 57–68.
- [50] A. Skowron. Extracting laws from decision tables. *Comput. Intell. Int. J.* 11(2) (1995) 371–388.
- [51] A. Skowron and C. Rauszer. The discernibility matrices and functions in information systems. In: R. Slowiński (ed), *Intelligent Decision Support – Handbook of Applications and Advances of the Rough Sets Theory, System Theory, Knowledge Engineering and Problem Solving*. Vol. 11. Kluwer, Dordrecht, 1992, pp. 331–362.
- [52] D. Ślezak. Approximate entropy reducts. *Fundam. Inf.* 53 (2002) 365–387.

51

Intervals in Finance and Economics: Bridge between Words and Numbers, Language of Strategy

Manuel Tarrazo

So when the computer came along – and more particularly, when I understood that a computer is not a number cruncher, but a general system for dealing with patterns of any type – I realized that you could formulate theories about human and social phenomena in language and pictures, and whatever you wanted on the computer, and you didn't have to go through this straitjacket of adding a lot of numbers.

–Herbert Simon [1]

This study examines interval analysis in finance and economics. Our goal is twofold: attract economic and finance colleagues into interval methods and, given the interdisciplinary nature of interval analysis, attract those colleagues familiar with interval methods into economic and finance problems.

We can understand granular computing as modeling with intervals, which can be linguistic (words, concepts-based), numerical, or both numerical and linguistic (hybrid). Granular computing describes several problem-solving methodologies, and examples can be found in a wide range of contexts. It is advantageous to approach granular computing using the most recent references. Klir describes granulation as ‘a fuzzy counterpart to classical quantization,’ [2, p. 295], arising from a ‘fundamental inconsistency between the infinite precision required to distinguish real numbers and the finite precision of any measuring instrument,’ [2, p. 297]. Pedrycz and Gomide [3] study granulation in alternative, but related, contexts: First, in the context of frames of cognition, by exploring how linguistic labels carry fuzzy information [3, p. 66]; later on in the context of linguistic variables: ‘The number of linguistic values defines the granulation, and therefore the fuzzy partition, of the corresponding universe,’ [3, p. 169], and also as ‘fuzzy points’ or information ‘patches.’ These patches can be used to approximate fuzzy functions, Kosko [4]. These patches, in turn, make fuzzy sets work as universal approximators – Kreinovich, Nguyen, and Yam [5].

The main message of our contribution can be clearly stated: intervals bridge the traditional computing void between using words and numbers and, as we will show, processing intervals can be regarded as the language or algebra of strategy. Zadeh [6, p. ix] observes that it is a tradition in science to regard natural languages (i.e., words) with suspicion because natural languages lack precision, and clinging to that tradition is costly. In effect, in the case of finance and economics, the case for advancing our decision making by incorporating natural language is easy to make by reflecting on the following: (1) the insufficiency of our theories and number-based models to represent the world, (2) the many problems for which there are neither theories nor number-based models, (3) the fact that economic and financial problem solving is made on the uncertain basis of an unknown future, (4) the different types of information (qualitative, quantitative, news that is more or less reliable, predictions, expectations, mere hunches, etc.), and (5) the fact that humans possess the ability to make decisions in even very imprecise settings.

Strategizing is what we do when our knowledge is insufficient. One of the contributions of modern finance is to provide ways to protect ourselves from a lack of knowledge, which is often represented as an interval for potential future values of the relevant variables or events. In the number-based case, we may do so by trading derivative securities (e.g., a common stock option) to take care of a potential range of future closing values of a given common stock (hedging). We can also hold positions in a variety of stocks to diminish the possibility that some of them will turn out to be bad investments (diversification). In addition, we may buy specific fixed-income instruments to cover for a range of future interest rates (immunization). The risk management technique of covering a range of outcomes with (apparently contradictory) positions extends to non-numerical situations. Take life or car insurance, for example. We all take care not to have any problems but, in any case, we cover for the range of outcomes (interval: [accident, no accident]) by purchasing insurance. Intervals are, therefore, instrumental in our strategic (risk management) efforts, whether the problem requires numerical or conceptual treatment. That is why exploring interval methodologies is critical to advance our fields. Moreover, in finance and economics it is very hard to find any problem when purely numerical information suffices to reach a permanent solution; instead, we find ourselves using hybrid information (words, numbers) to find adaptive or progressive solutions that we hope will bring us closer to our goals. Think, for example, of your salaries, mortgage contract, or retirement plans.

These pages are not ‘A Survey of Granular Computing in Economics and Finance.’ Such a task, if feasible at all (contributions to the area of fuzzy sets were already estimated to exceed 15,000 in the mid-1990s), would probably need a different medium than a single study. Our strategy is to examine modeling in light of the type of decisions in our field and show the potential of interval-related methodologies in financial decision making. Our more modest approach, however, makes for a coherent narrative, where we can offer a greater amount of detail to the readers. The study has four parts: (1) a brief overview of decision making in economics and finance; (2) a presentation of selected interval methodologies, such as symbolic algebra, interval analysis, and other fuzzy-set-based methods; (3) a review of some applications; and (4) the critical role of intervals in the management of uncertainty through strategy making.

During the last half of a century we let ourselves be seduced by the comforting thought that our problems could be routinely solved with numbers – and perhaps that computers would do that for us. In general, quantitative analysis always enhances our understanding of the problem. Setbacks appear when problems are exclusively analyzed with exact methods, usually calculus plus probability, because these methods waste a great deal of information and disregard other potentially helpful methodologies. To the extent that interval-based methods use natural language, they provide new ways to invigorate qualitative analysis in finance and economics.

51.1 Decision Making in Economics and Finance: An Overview

In this section we briefly introduce decision making in finance and economics to show the potential of granular computing (hybrid: words plus numbers), and especially interval-based methods, in these disciplines.

Our societies are based on the exchange of goods and services, and every exchange represents a decision by economic agents – individuals or households, firms, and public institutions. Economics studies

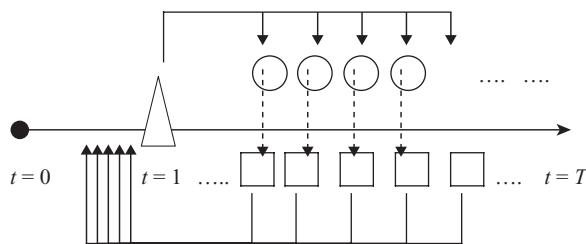


Figure 51.1 Schematic representation of a financial decision

exchanges at the household and at the aggregate levels (e.g., regional, nationwide, or multicountry groupings). Although modern economics has many branches and overlaps with many other disciplines (ecology, computer science, mathematics, psychology), its theoretical backbone is organized into macro- and microtheory. Macroeconomics studies the behavior of economic aggregates as represented, for example, in the national income product accounts (consumption, investment, government expenses, money demand and supply, gross national product, import and exports, employment, and so on). Microeconomics, on the other hand, studies the behavior of individual and less aggregated units and markets. A key characteristic of microeconomics is that it makes use of stylized and frequently homogeneous agents and markets. This is done for two main reasons: to uncover the main factors in decision making, and to conform to the aggregation requirements of macroeconomics. For example, the ‘perfect competition’ model is a famous microeconomic construction that shows how markets can function in a setting where neither single producers nor single consumers have the power to influence prices. In contrast to microeconomics, business and finance concentrate on specific firms and individuals/households in specific situations and markets.

Finance focuses on how individuals and firms administer their resources over time, which, in our complex economies, is done using financial instruments (stocks, bonds, loans, bank accounts, etc.). A schematic, but rather effective, depiction of a financial decision is given in Figure 51.1.

The decisions we study in finance are important because they affect several periods in the future: for example, college financing, house purchase, and retirement planning for individuals; for firms, new product introduction, mergers and acquisitions, and long-term financing. The recommended process is first to start thinking about the decision ($t = 0$), indicated by a triangle, and to start preparing for it well before doing anything irreversible. The preparations may consist of saving for the down payment of a house, doing market research, or saving for retirement. The next step is to actually do what we are planning for – buy the house, go to college, open a factory overseas, start the development of a new pharmaceutical drug, and so on. The action is represented by the triangle in Figure 51.1.

The squares represent financial instruments held by investors, who provide funds (the line arrows linking squares to the triangle) to initiate the project. Later on, as the project generates its own cash flows, represented by circles, part of these are returned to the investors as capital gains and/or dividends – the dashed arrows going from the circles to the squares. Note the sophistication of our system. Not only do we not need to ‘have’ the funds to start a project (buy a house, go to college, start a company), but we obtain financing for these projects against a future that does not exist yet. Herein lies the mechanism responsible for the impressive growth capacity of our economic/financial system: the ability to make things happen based on the optimism that the project will turn out all right. This factor also explains the importance of the financial sphere to manage risk: through diversification, hedging, and insurance, investors and lenders will try to take positions that diminish their exposure to certain risks.

Early contributions to economics – Smith’s 1779 *Wealth of Nations*, Marx, and Stuart Mill for example – invariably took text form. Malthus and Ricardo employed mostly qualitative analysis in their writings as well. But the need to quantify in their specific problems soon became evident (population-food; incomes-prices for farmers, landlords, and industrialist). By the end of the nineteenth century, a group of

distinguished researchers (Walras, Jevons) used the calculus concepts of increments and derivatives to model marginal gains and marginal costs. Alfred Marshall integrated marginal concepts into the coherent supply-and-demand model that can still be found in textbooks. His book, *Principles of Economics* [7], published in 1890, is still good reading and reveals calculus to be the main tool (other than sharp perception and logical reasoning) used by Marshall. A particularly fertile period followed with respect to economic theories. By the mid-1940s, Samuelson's [8] text *Foundations of Economic Analysis* (1948, early edition dated 1941) seemed to justify the impression that everything in economics could be handled with exact methods (calculus, difference, and differential equations) and probability. After all, powerful mathematical tools had already been used successfully to model the world in physics and medicine. The advent of machine computing was imminent and, with it, the possibility of furthering quantitative analysis. Computers arrived, and additional mathematical methods, such as mathematical programming, were developed. Despite very rare exceptions (e.g., Herbert Simon's contributions), these developments pushed economics more and more into exact, quantitative methods.

However, non-numerical efforts continued expanding our knowledge (Schumpeter, Veblen, Boulding, Galbraith, Hayek). Moreover, major contributors to conventional quantitative methods often warned about the limitations of purely quantitative methods. One of the best examples is Keynes, whose path-breaking macroeconomic analyses were developed on mostly qualitative terms, and who was wary about the newly born econometrics. Early in the 1920s, Knight distinguished between risk (measurable) and uncertainty (not measurable).

In microeconomics, consumer theory is a very good example of the need to use hybrid (words-plus-numbers) methods. During the 1930s, Von Mises suggested that consumers optimize in ordinal rather than cardinal terms. In a 1948 commentary on a linear expenditure system, Samuelson pointed out that the entire analysis could have been carried out using an ordinal preference system. In Arrow's [9, p. 109] words, "In the field of consumers' demand theory, the ordinalist position turned out to create no problems; cardinal utility has no explanatory power above and beyond ordinal." Further, some properties thought to be essential for optimizing were not 'To summarise: continuity and order are invariant properties of ordinal variables, but differentiability and convexity are not. Fortunately, the later properties are not essential to the optimization process' (McManus [10, p. 102]; see also Uzawa [11]).

Luce [12] and Armstrong (see references to Armstrong's research in Luce's study) anticipated some of the perception-based rationale to use intervals in preference analysis. Luce notes that 'the intransitivity of some indifference relations ... reflects the inability of an instrument to discriminate relatively to an imposed task' [12, p. 179]. Luce's work followed on Armstrong who already had pointed out earlier, in 1950, 'the nontransitiveness of indifference must be recognized and explained on [the bases of] a theory of choice, and the only explanation that seems to work is based on the imperfect powers of discrimination of the human mind whereby inequalities become recognizable only when of sufficient magnitude' (Luce [12, p. 179]).

The appeal for hybrid methodologies even employs the wording of modern fuzzy sets:

- 'There has always been a temptation to classify economic goods in clearly defined groups, about which a number of sharp propositions could be made, and the popular liking for dogmas that have the air of being profound and are yet easily handled. But great mischief seems to have been done by yielding to this temptation, and drawing broad artificial lines of division where Nature has none. . . . There is not in real life a clear line of division between things that are and are not capital, or that are or are not necessities, or again between labor that is and not is productive' (Marshall [7, p. xv]).
- 'Most of the chief distinctions marked by economic terms are differences not of kind but of degree' (Marshall [7, p. 52]).
- 'Liquidity and carrying costs are both a matter of degree,' Keynes [13, p. 239].

Still, for the most part, the literature in finance and economics has inclined towards quantitative, numbers-only, exact methods.

The current state of the art presents difficult challenges specific to each discipline. With respect to macroeconomics, its situation can be summarized in a sentence: 'The reports of the death of large scale macroeconomic forecasting models are not exaggerated' (Diebold [14, p. 175]). One of the main

problems is establishing measurable causality relations and developing theory-based models. The 1930s and 1940s were rich in theory development, especially in the wake of Keynesian formulations. Computing provided further impetus. However, halfway through the 1980s macroeconomic modeling mutated from causal, or theory-based, models to non-causal models (e.g., time series). Unfortunately, without theory, macroeconomic efforts are cryptic, mystifying, and politically not saleable.

The major problem in microeconomics is that standardizing does not represent well either individual agents or markets. Further, it often leads to unconvincing, toylike representations of reality (e.g., general equilibrium formulations, the perfect competition paradigm).

In finance the challenge is immense. First, we need to learn how individuals handle time in their decision making. For example, it is not possible to calculate numbers (e.g., present values) without ‘bracketing’ the planning horizon. The horizon of many projects is not well defined (e.g., technological obsolescence of equipment, life expectancy, and retirement planning). Further, as the representation of the financial decision shows, financial problem solving is fully forward looking. One implication is that information is incomplete since we never know the future, and it comes in different forms (words, numbers, more or less reliable, etc.). Another implication is that every financial decision must involve risk management and strategy-making efforts. Finally, some sciences are able to rely on biological or physical constants and systems endowed with a degree of repetition. Economics, business and finance, however, can rely only minimally on repetition and, instead, must anticipate what is non-recurrent. (They are idiographic, as opposed to nomothetic, in Windelband’s terminology.)

The advancement of knowledge depends in part on studying each problem with the appropriate tools, and rarely on disfiguring the problem so that a given method can be applied to it. Otherwise, our findings may amount only to, as Hayek put it, ‘pretence of knowledge.’ As we have shown, the problems themselves seem to call for hybrid (words-plus-numbers) methodologies, some of which we will examine next.

51.2 Selected Interval Methodologies

In this section we review the methodologies supplying techniques, tools, and methods to modeling with intervals. As noted frequently during this study, the intervals can be of a numerical, linguistic, or of a hybrid (words and numbers) nature, thereby representing a bridge between words and numbers. Granular computing also includes other approaches such as handling concepts in a graphical manner to represent knowledge and to make decisions.

We distinguish four fountainheads supplying interval and granular computing methods of interest for finance and economics: fuzzy sets, symbolic algebra, interval mathematics, and concept mapping. Before we examine them, it is advantageous to briefly review the key elements in modeling, since they play a fundamental role in motivating the new methodologies.

Models are simplified versions of the reality we use to help us deal with particular problems. They can be numerical, linguistic, conversational, graphical, comprehensive, partial, and so on. In some cases, models mimic real processes, in some other cases – e.g., business cycle of prosperity and recessions, evolution of interest rates – models resemble shortcuts we make because we cannot ‘capture’ their corresponding reality. There are six key elements in modeling efforts:

1. Every model has a significant structure.
2. The structure is always relational – something related to something else.
3. All our intellective productions are only approximations, representations of whatever is real, including ourselves: ‘We know our own subject only as appearance, not as it is itself’ (Kant [15, p. 168]). The concepts we work with – categories and the resulting schemas – are limited by our human physical and thinking limitations.
4. Models often include notions of connectedness and causality.
5. Modeling also includes a particular way (a syntax or algebra) to process logical elements. In medieval times, an ‘algebraist’ was the person who could compose and fix broken bones, cuts, and other events we would now describe as medical. Nowadays, an algebraist arranges matrices and vectors (linear algebra), or sets and classes (algebra of sets).

6. The ‘human’ element: (a) a tendency to confuse causality with sequencing (Hume); (b) laying meaning when we have no idea of what is going on (Leibniz’s principle of sufficient reason – ‘There must be a reason for it’); and (c) a risk to be distracted with epiphenomena: ‘Appearances are only representations of things which are unknown as regards to what they may be in themselves. As mere representations, they are subject to no law of connection save that which the connecting faculty prescribes’ (Kant [15, p.173]). Interestingly, Schopenhauer’s contributions and his extensions to Kant’s representation model seem to anticipate fuzzy-set-related concepts (Tarrazo [16]).

A. Words, numbers, and fuzzy sets

An efficient way to appreciate the importance and relevance of fuzzy sets for finance and economics is to examine the evolution of the research program of Lotfi A. Zadeh. Fuzzy sets – sets whose membership is a matter of degree – were first introduced in Zadeh [17]. The years that followed produced an avalanche of contributions, excellent overviews can be found in Pedrycz and Gomide [3] and Klir and Yuan [18]. Klir and Yuan [19] and Yager et alia [20] contain selections of Zadeh’s own studies (42 and 18, respectively) in the areas related to fuzzy sets. In addition to more than 100 fuzzy-set-related contributions, Zadeh has also made over a hundred contributions to non-fuzzy, engineering, computing, and information-related areas. The studies we mention in this section can be found in the two previously mentioned selections.

Zadeh [17] also studied fuzzy sets intersections and convexity. It is very significant that the first contribution in this area focuses on sets and the algebra of sets. It means that the origin of the idea of ‘degrees’ was, if not incubated in, at least cradled in the algebra of sets and, more tellingly, in words. As we know, every word (cars, houses, 4-year colleges . . . students) can represent sets. Fuzzy sets, therefore, like Plato’s forms, have a language-based genealogy. Zadeh [21] studied the concept of ‘shadows of fuzzy sets,’ which pierced the limitations of probability to represent sets: ‘In fact, there are many situations in which the source of imprecision is not a random variable but a class or classes which do not possess sharply defined boundaries’ [21, p. 37]. Zadeh [22] focused on the grouping of sets into classes and similarity relations. This line of research would become the foundation of one of the major areas within fuzzy sets: relational equations probabilistic modeling. Note that up to this point fuzzy sets have been all about words. In short time, however, the author would also explore fuzzy algorithms, fuzzy languages and their machine representation, fuzzy semantics, fuzzy systems, mappings, and fuzzy control.

Because we are all part of complex systems, our decisions must take into account those systems and their properties: ‘Why is fuzziness so relevant to complexity? Because no matter what the nature of the system is, when complexity exceeds a certain threshold it becomes impractical or computationally infeasible to make precise assertions about it’ (Zadeh [22, p. 470]). In Zadeh [23], this idea evolved into what is referred to as Zadeh’s incompatibility principle. Zadeh’s research in ‘approximate reasoning’ takes advantage of the human ability to reason in qualitative, imprecise terms when facing overwhelming system complexity (see Zadeh [24]).

Bellman and Zadeh [25] focused on decision making in a fuzzy environment: ‘By decision making in a fuzzy environment is meant a decision process in which the goals and/or the constraints, but not necessarily the system under control, are fuzzy in nature. This means that the goals and/or the constraints are classes of alternatives whose boundaries are not sharply defined’ [25, p. 141]. This study questioned the practice of equating imprecision with randomness, and it was also one of the first contributions integrating words and numbers (i.e., hybrid modeling).

Zadeh [26] expanded on the ‘linguistic approach’ to decision making, which he applied to the study of preferences as a basis of choice in social contexts [27]:

In a sharp break with deeply entrenched traditions in science, the linguistic approach abandons the use of numbers and precise models of reasoning, and adopts instead flexible systems of verbal characterizations which apply to the values of variables, the relations between variables and the truth-values as well as the probabilities of assertion about them. The rationale for this seemingly retrograde step of employing words in place of numbers is that verbal characterizations are intrinsically approximate in nature and hence are better suited for the description of systems and processes which are as complex and as ill-defined as those which relate to human judgment and decision making. . . . It should be stressed, however, that the linguistic approach is into the traditional non-mathematical way of dealing with humanistic systems. Rather, it represents a

blend between the quantitative and the qualitative, relying on the use of words when numerical characterizations are not appropriate and using numbers to make the meaning of words more precise (Zadeh [26, pp. 340–341]).

In 1973, Zadeh initiated publications on social systems and large-scale systems – see also Zadeh's [28] and [29]. Given the complexity of these systems, we can only approximate them; therefore, the only way to handle them is with fuzzy sets – words. Zadeh [23] introduced the concept of a linguistic variable, 'a variable whose values are sentences in a natural or artificial language.' The concept of linguistic variable benefited from Zadeh's further research on fuzzy constraints – 'elastic' constraints on the values assigned to a variable. Zadeh's research on linguistic variables became part of investigations on fuzzy logic and approximate reasoning [23, 30], and also on possibility theory as an alternative to probability [31].

Zadeh [32] introduced the concept of information granularity, a concept closely associated with fuzzy sets and related areas. Granularity handles information that comes in lumps, which appear often in a noncontinuous but still gradual form, and whose semantic content varies. For traditional computing and modeling, information granularity creates problems of continuity, convexity, linearity, static-dynamic classifications, probabilistic nature, calculability, and precision. The good news is that granularity can convey the necessary information in the most direct way. For example, when we are told that 'the parking lot is full,' we turn around and look for parking elsewhere. There is no need to count the cars, to learn about the parking lot's capacity, to check that the first derivative of the car-numbers function is zero, or that the accumulation point in the limit computation has been reached, and so on. Zadeh [28] remarks that 'much of the uncertainty associated with soft data is nonstatistical in nature.'

After studies on prototype theory (how an instance can become a representative element for a given fuzzy set), precisation of meaning, fuzzy probability, and management of uncertainty in expert systems, Zadeh [33] focused on commonsense knowledge, which he described as a 'collection of dispositions.' Dispositions are statements expressing informal knowledge, but some knowledge nonetheless: birds can fly, icy roads are slippery, what is scarce is expensive, the price is low don't expect much, etc. The knowledge content of such statements can be assessed with test-score semantics. An outline of a theory of usuality, one way to come up with dispositions, was set forth in Zadeh [34, 35], and a theory of dispositions was offered in Zadeh [35]. As could be expected, linguistic variables are instrumental in commonsense knowledge and reasoning.

The title of one of Zadeh's recent studies summarizes where his research has led him to: 'From Computing with Numbers to Computing with Words –From the Manipulation of Measurements to the Manipulation of Perceptions' (Zadeh [6]). Note that his entire research program reflects each of the previously noted key elements in modeling – structure, relational representation of knowledge, approximate, and produced and used by humans. Kant had already stressed the role played by perception: 'Anticipation of perception: In all appearances, the real that is an object of sensation has intensive magnitude, that is a degree' (Kant [15, p. 201]). Zadeh [6] notes,

'[What previous researchers did not appreciate was] the fundamental importance of the remarkable human capability to perform a wide variety of physical and mental tasks without any measurement. . . . Underlying this remarkable ability is the brain's crucial ability to manipulate perceptions. . . . Measurements are crisp numbers whereas perceptions are fuzzy numbers or, more generally, fuzzy granules, that is, clumps of objects in which the transition from membership to nonmembership is gradual rather than abrupt. . . . A concomitant of fuzziness of perceptions is the preponderant partiality of human concepts in the sense that the validity of most human concepts is a matter of degree. . . . Furthermore, most human concepts have a granular structure and are context-dependent. . . . In essence, a granule is a clump of physical or mental objects (points) drawn together by indistinguishability, similarity, proximity or functionality. . . . The methodology of computing with words may be viewed as an attempt to harness the highly expressive power of natural languages by developing ways of computing with words or propositions drawn from a natural language. To a limited extent, this has been done in fuzzy logic through the use of concepts of linguistic variables and linguistic if-then rules. Computing with words moves much further in this direction' (Zadeh [6, p. 37, and volume foreword]).

In sum, focusing on perceptions means that although things and events in the world may be quantitative, what they mean to us and the way we primarily relate to them are qualitative. The rationale for computing with words boils down to this: we need to compute with words when (a) concepts are too complex to manipulate with numbers, (b) our knowledge is insufficient, (c) the information is too imprecise, and (d) ‘when there is a tolerance for imprecision which can be exploited to achieve tractability, robustness, low solution costs and better rapport with reality’ [6, p. 48].

B. Symbolic Algebra and Relational Equations

Set theory is a relatively recent product of modern mathematics (Cantor, 1870s). It deals with groupings of logical objects called ‘elements.’ At some point it was thought that set theory could work as a foundation for the entire mathematical edifice, but that was a tall order. Axiomatic set theory is a formalization of naïve set theory, where sets are taken as a primitive concept, that is, a concept explained by its usage. Langer [36] illustrates how the concepts from set theory – elements, sets, and classes – can be used as models themselves since they all represent logical structures. These logical structures are likely to include analogies, abstractions, and generalizations that permit logical, although symbolic, problem solving.

Classical or crisp sets, and groupings of sets (classes), have ‘crisp’ membership functions because they incorporate the principle of excluded middle. Membership, therefore, is a binary deal. Under these conditions, the classical algebra of sets cannot be very ‘symbolic,’ because symbols derive their power of representation from having a range of meaning. Zadeh’s fuzzy sets are particularly promising to enable set theory to be used in practical problems, especially in those problems when we must work with grouping of fuzzy sets – take, for example, the retirement planning problem.

At different points in our lives we must take care of different financial responsibilities. We can define three classes of concepts:

$$\text{Age} = \{25, 35, 45, 55, 65\}$$

$$\text{Properties} = \{\text{Liquidity, Income, Appreciation, Safety}\}$$

$$\text{Assets} = \{\text{Savings Accounts, Bonds, Stocks, Real Estate}\}$$

We have some knowledge indicating the relative importance of each of the properties for each of the representative ages, and also concerning the properties of each of these financial instruments. We can represent this relational knowledge as two relational matrices: $R1 = \text{Assets-Properties}$, and $R2 = \text{Properties-Age}$. The entries in these relational matrices are (fuzzy) numbers indicating possibility rather than probability. For example, safety is more important for a 65-year-old than for a 25-year-old. The problem for a financial planner is to compose or integrate the knowledge contained in each of the relationships, that is, to find another relation $R3 = \text{Assets-Ages}$, which would suggest the most advisable portfolios at different ages. That is, $R3 = \text{Assets-Ages} = R1 \bullet R2$, where ‘•’ represents a given rule of composition of fuzzy relational equations. And the (fuzzy) numbers obtained can be interpreted as indicators of adequacy, or suitability, and used as simple percentages: ‘People around 45 years old should have approximately 10% of their wealth in savings accounts, about 10% in bonds, about 40% in stocks for growth, and the rest in real estate.’ The financial advisor and the investor can speak the same language, among other advantages. Contrary to appearances, finding these numerical indicators is easy to do, see Tarrazo [37–39]. A summary of the model discussed in this section is given in Exhibit 51.1. Tarrazo [39] applies relational equations to business planning and strategy.

Economics and finance colleagues may have noted that the approach taken in the financial planning problem can easily be generalized to a rather flexible, relatively context-independent model of choice. The objects can be, for instance, medications with known characteristics and the subjects can be patients exhibiting specific sets of symptoms:

$$R1 = \text{Objects-Characteristics}, R2 = \text{Characteristics-Subject situation}$$

$$R3 = \text{Objects-Subject} = R1 \bullet R2$$

Exhibit 51.1 Relational equations and individual financial planning

Classes, sets, and relational matrices:

$$\text{Assets} = \{\text{Sav, Cb, Cs, Re}\} = \{\text{Savings accounts, Bonds, Stocks, Real Estate}\}$$

$$\text{Properties} = \{\text{L, I, A, S}\} = \{\text{Liquidity, Income, Appreciation, Safety}\}$$

$$\text{Age} = \{25, 35, 45, 55, 65\}$$

	L	I	A	S		25	35	45	55	65
Sav.	0.6	0.3	0.0	0.4	L	0.3	0.15	0.1	0.1	0.0
Cb	0.2	0.5	0.1	0.3	I	0.1	0.0	0.1	0.2	0.6
Cs	0.2	0.1	0.4	0.2	A	0.0	0.6	0.7	0.5	0.0
Re	0.0	0.1	0.5	0.1	S	0.6	0.25	0.1	0.2	0.4

$$\begin{aligned} R3 &= R1 \bullet R2 = R1 (\text{Assets, A-properties}) \bullet R2 (\text{A-properties, Age}) \\ &= R3 (\text{Assets, Age}) \end{aligned}$$

Max-min rule:

Assets\Age:	25	35	45	55	65
Savings	0.40	0.25	0.10	0.20	0.40
Cbonds	0.30	0.25	0.10	0.20	0.50
Cstocks	0.20	0.40	0.40	0.40	0.20
Restate	0.10	0.50	0.50	0.50	0.10
Totals	1.00	1.40	1.10	1.30	1.20

$$\begin{aligned} a11 \text{ in } R3 &= \max (\min (0.6, 0.3), \min(0.3, 0.1), \min(0.0, 0.0), \min(0.4, 0.6)) \\ &= \max (0.3, 0.1, 0.0, 0.4) = 0.40 \end{aligned}$$

Max-min rule (normalized):

Assets\Age:	25	35	45	55	65
Savings	0.40	0.18	0.09	0.15	0.33
Cbonds	0.30	0.18	0.09	0.15	0.42
Cstocks	0.20	0.29	0.36	0.31	0.17
Restate	0.10	0.36	0.45	0.38	0.08
Totals	1.00	1.00	1.00	1.00	1.00

Arithmetic product rule:

Assets\Age:	25	35	45	55	65
Savings	0.45	0.19	0.13	0.20	0.34
Cbonds	0.29	0.165	0.17	0.23	0.42
Cstocks	0.19	0.32	0.33	0.28	0.14
Restate	0.07	0.325	0.37	0.29	0.10
Totals	1.00	1.00	1.00	1.00	1.00

Source: Tarrazo [38, p. 105].

What matters is that we may be able to make headway despite imprecision, lack of hard numbers, the imposing complexity of the problem, the lack of comprehensive planning models, and the insufficient knowledge of the user.

Relational equations are one of the major contributions of the methodology of fuzzy sets. The interested reader should consult first the textbooks on fuzzy sets by Klir and Yuan [18], Pedrycz and Gomide [3], and the introductory work by Tanaka [40]. Peeva and Kyosev [41] condense the state-of-the-art of the field and include very useful MATLAB programs.

Relational equations admit multilayering and different composition rules. They also encapsulate distinctions between intension (quality), and quantity (extension), as studied in the very specialized monograph by Belohlavek [42]. The relational equations probabilistic manipulations are the modeling language of fuzzy sets. Further, they come very close to being the ‘universal algebra’ (or ‘ars combinatorica’) envisioned by Leibniz: a language to express and process knowledge.

C. Interval Analysis

We learned about interval analysis while consulting Klir and Yuan’s chapter on fuzzy numbers [18, p. 117]. The concept of a fuzzy number and the need for such a construct can be hard to understand. A simple way to introduce them is to see that sometimes we use ranges of numbers to express word concepts. For example, a portfolio manager may say, ‘in this portfolio we need some assets with returns in the 20–30 percent range.’ That would be an alternative to expressing the thought that growth stocks should be purchased. The emphasis is not on the type of instruments because that information is already contextually understood, but on the (fuzzy) number, or numerical range, which is the important information transmitted.

We then went to our university library and checked out the available holdings on interval analysis: Deif’s *Sensitivity Analysis in Linear Systems* [43], Kuperman’s *Approximate Linear Algebraic Equations* [44], Moore’s [45, 46] monographs on interval analysis, Neumaier’s *Interval Methods for Systems of Equations* [47], and Hansen’s *Global Optimization Using Interval Analysis* [48]. During our first reading, there were issues of rounding and floating-point arithmetic that seemed unnecessary to understand issues such as approximate equations and global optimizations searches. Therefore, we focused on anything related to approximate simultaneous equations systems, simply because these systems appear in every problem of interest in economics and finance:

- Microeconomics: consumer choice, production, market equilibrium and price determination, multi-market equilibrium, general equilibrium.
- Macroeconomic models.
- Finance: corporate financial planning, portfolio optimization, break-even analysis, inventory, capital structure, etc.

Simultaneous equations are central in economics and finance for two major reasons:

- (a) Decision-making models consist of objectives or goals, usually expressed in an objective function, variables upon which we have control and others upon which we do not have control (i.e., given), and restrictions with respect to how our choice variables relate to the objectives and also concerning how much we can do with our control variables (income and purchase of a given good, for example). The solution to decision problems takes the form of a compromise between what we want and what we can have, or between what we would like to do and what we can actually do. The ‘equilibrium’ adjective in the headings of the topics listed above cue us to find a solution for the set of simultaneous equations representing the problem. Note that simultaneous equations systems such as $Ax = b$ subsist whether the optimization has more variables than equations (usual mathematical programming case), or more equations than variables (usual least squares situation).
- (b) A simultaneous equation system represents a point in the hyperspace expressed in the field of real numbers, R^k . As long as the model handles ‘ k ’ variables to find a given point, a system of the form $Ax = b$ appears. (A represents a k -by- k matrix, and b and x k -by-1 vectors, respectively).

Interval systems of equations can be expressed as

$$\begin{vmatrix} [a_{11}^L, & a_{11}^U] & [a_{12}^L, & a_{12}^U] \\ [a_{21}^L, & a_{21}^U] & [a_{22}^L, & a_{22}^U] \end{vmatrix} = \begin{vmatrix} x_1 \\ x_2 \end{vmatrix} \begin{vmatrix} [b_1^L, & b_1^U] \\ [b_2^L, & b_2^U] \end{vmatrix}, \quad (1)$$

where the superscripts ‘L’ and ‘U’ stand for lower and upper, respectively, in (1). The terms ‘inf’ (infimum) and ‘sup’ (supremum) are also used to denote lower and upper bounds, respectively. Sometimes, the intervals are expressed with reference to an error term, a real valued number ε , which modifies one or more parameters as in $[\delta \pm \varepsilon]$. The error term does not have to be the same for each parameter. In general, we can express approximate or interval systems of equations as: $A^I x^I = b^I$.

Two fundamental problems in (1) are (1) finding the solution set $x^* = \{x_1^*, x_2^*\}$, which is the shaded area in the charts; and (2) establishing the range for the admissible solutions, or intervals of uncertainty, for each of the variables involved, $x_i^* = [x_{i1}^L, x_{i1}^U]$. The boxes formed by the interval of uncertainty include the solution set and other non-feasible points.

An important area within interval analysis focuses on rounding errors and floating-point accuracy, and the intervals are very, very small – infinitesimal. This area is known as validated, reliable computing, and verified computing, and it has yielded impressive results in assuring the significance and accuracy of computing results, in providing verified proofs of mathematical theorems, and also in providing the only reliable methods for global searching in some multivariate optimization problems. Such small intervals, however, do not bring any relief to the problems we are dealing with in finance and economics, where the measurements, variable definitions, and even the model characterizations are suspect. We wanted a formal way to relax the exact solutions in traditional models to make them more practical. What we are looking for is known as linear algebraic approximate equations systems (AES).

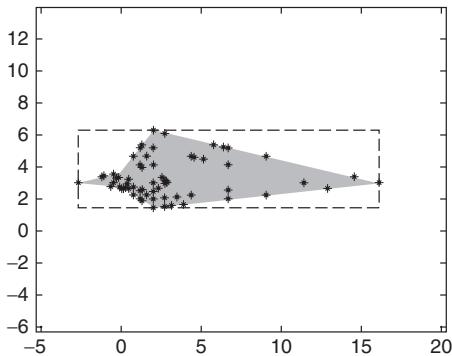
Exhibit 51.2 shows some AES. The graphics were created with ‘INTLAB –INTerval LABoratory’, Version 5.3. This software package has been developed by Siegfried M. Rump at Hamburg University of Technology, Germany, and is available at no cost for private and academic research purposes: <http://www.ti3.tu-harburg.de/rump/intlab/>.

Instead of point solutions, AES solutions are boxes, areas, spheres, and hyperspheres. Example 1 can be used to learn what happens when the interval parameters increase by adding and subtracting a common error ε . The solution set remains in the positive orthant for $\varepsilon \leq 0.5$ (e.g., $\varepsilon = 0.1$), but spills over to the negative orthants when $\varepsilon > 0.5$ (e.g., $\varepsilon = 0.7$). For $\varepsilon = 1$, the system becomes critically ill conditioned and the solution set is unbounded. In some cases, as in the Example 6, there is not much room for errors or imprecision because the solution set is relatively small. Note also that the solution set need not be convex, and in some cases may not form an ‘insular’ (i.e., non-disjoint) set. We know that exact models can break down (singular matrices), or be very sensitive to changes in the parameters, but AES indicates how easy it is to have false confidence in exact models.

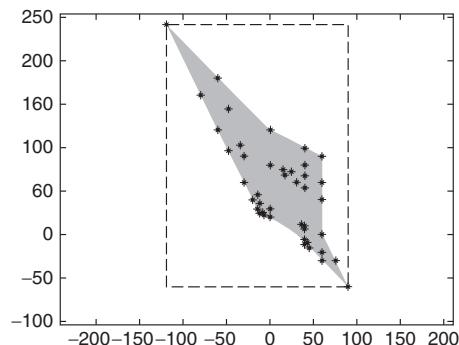
AES may have immediate benefits for research on specific practical problems. A portfolio manager, for example, may want to observe how close the optimal holding solutions are to the ‘sell, short’ (negative) side, instead of to the ‘buy, long’ (positive) side – a world of difference in terms of actual investing strategy and trading. In the case of portfolio optimization, wide ranges – including both the positive and the negative regions, for example – may indicate that the mathematical model is simply not reliable. AES can provide an indication of the structural reliability of the model and also complement the information provided by probability indicators. Applications of AES to portfolio optimization (Markowitz and ‘beta’ models), proforma financial statements generation, and macroeconomic modeling can be found in Tarrazo [49].

In general, we believe that AES offers great potential in finance and economics for the following, reasons, among others:

- (a) Recognizes the approximate nature of our observations, especially in the social sciences.
- (b) Accounts for potential numerical errors.
- (c) Creates room alternative model specifications (linear–non-linear, static–dynamic, stochastic–deterministic, etc.). For example, a box is a space defined by two interval equations and allows for linear and non-linear behavior in its inside.
- (d) Permits better modeling of expectations and uncertainty.

Exhibit 51.2 Linear algebraic approximate equations systems (AES)1. Kupperman [44, p. 145], $\varepsilon = 0.7$ 

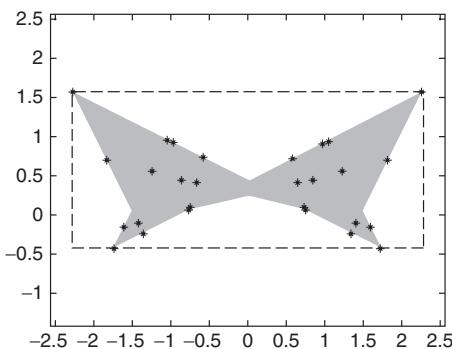
2. Hansen [48, p. 36]



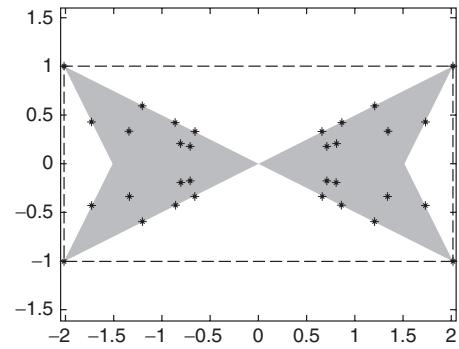
1 $A = [\text{infsup}(0.3, 1.7) \text{ infsup}(-2.7, -1.3); \text{infsup}(0.3, 1.7) \text{ infsup}(1.3, 2.7)]; b = [\text{infsup}(-4.7, -3.3); \text{infsup}(7.3, 8.7)]; \text{plotlinsol}(A, b, [], 1)$

2 $A = [\text{infsup}(2, 3) \text{ infsup}(0, 1); \text{infsup}(1, 2) \text{ infsup}(2, 3)]; b = [\text{infsup}(0, 120); \text{infsup}(60, 240)]; \text{plotlinsol}(A, b, [], 1)$

3. INTLAB example



4. Neumaier [47, p. 92]



3 $A = [\text{infsup}(2, 4) \text{ infsup}(-1, 1); \text{infsup}(-1, 1) \text{ infsup}(2, 4)]; b = [\text{infsup}(-3, 3); 8]; \text{plotlinsol}(A, b, [], 1)$

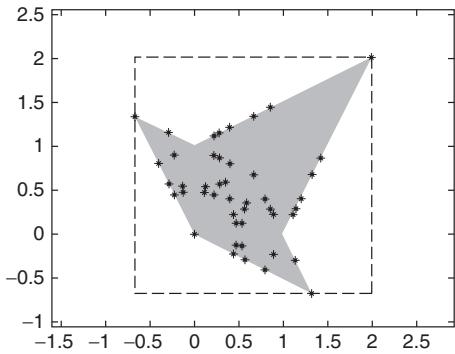
4 $A = [\text{infsup}(2, 4) \text{ infsup}(-1, 1); \text{infsup}(-1, 1) \text{ infsup}(2, 4)]; b = [\text{infsup}(-3, 3); \text{infsup}(0, 0)]; \text{plotlinsol}(A, b, [], 1)$

AES, as other interval methods, forms a bridge between words and numbers. This can be shown in different ways. Using the portfolio optimization case, the positive side of solution enclosures for the optimal holdings represent the words ‘long’ and ‘buy’; the negative sides, the words ‘short’ and ‘sell.’ Another way of showing how AES bridges words and numbers is to reflect on the numerical enclosures represented by relational equations in the previous subsection, which also uses portfolio optimization as an example.

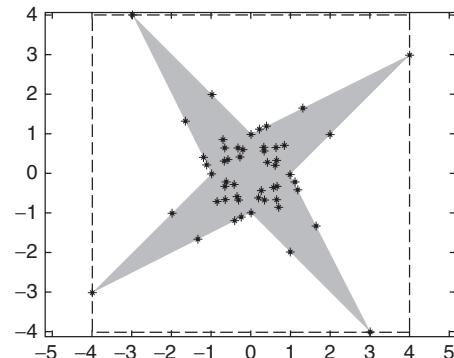
The relationship between fuzzy numbers and interval analysis must be studied with care. In a way, one could say that fuzzy numbers are to interval equations what AES are to reliable computing: both

Exhibit 51.2 (continued)

5. Deif [43, p. 118]



6. Deif [43, p. 60]



5 $A = [\text{infsup}(2,4) \text{ infsup}(-1,1); \text{infsup}(-1,1) \text{ infsup}(2,4)]; b = [\text{infsup}(0,2); \text{infsup}(0,2)];$
 $\text{plotlinsol}(A,b,[],1)$

6 $A = [\text{infsup}(2,4) \text{ infsup}(-2,1); \text{infsup}(-1,2) \text{ infsup}(2,4)]; b = [\text{infsup}(-2,2); \text{infsup}(-2,2)];$
 $\text{plotlinsol}(A,b,[],1)$

Note: The graphics were created with ‘INTLAB –INTerval LABoratory’, Version 5.3. Developed by Professor Siegfried M. Rump at Hamburg University of Technology, Germany, <http://www.ti3.tu-harburg.de/rump/intlab/>

fuzzy numbers and AES employ much larger intervals than their counterparts. This, however, is a rather crude characterization because the size of intervals may or may not say much without making reference to specific modeling cases. Furthermore, what matters is making models more helpful by using the best interval width for the problem at hand. The applications of AES to portfolio optimization add to the interpretation of numerical solutions and provide additional indicators of modeling reliability, but the standard approach to the problem is retained in its basic form. The applications of fuzzy sets to portfolio theory, as shown in the relational equations case, respond to different concerns (too imprecise definitions, lack of knowledge, lack of an acceptable numerical asset allocation model, etc.). Despite its inherent difficulty, optimizing a portfolio of common stocks is still a much less difficult problem than comprehensive retirement planning. The difficulties involved in quantitative asset allocation, such as that used in retirement planning, are very considerable as noted in Tarrazo and Murray (2004). This study shows some serious misunderstandings found in most textbooks, and even made by some well-known researchers in well-respected sources.

Readers interested in applying interval methods to their research problems may opt to focus on simultaneous equations systems, as we did, and leave aside infinitesimal accuracy and computing issues such as programming language and hardware which, nonetheless, may need to be considered in global optimization endeavors. With respect to interval analysis software, currently INTLAB only accepts 2-by-2 systems. Tarrazo [37, 49] provides a way to solve AES and obtain uncertainty ranges using binary programming and Excel Solver using Microsoft Excel spreadsheets. The premium version of Microsoft Excel Solver includes some interval methods to solve global optimization problems. Intel Corporation has very recently launched its Intel® Math Kernel Library, which ‘contains highly optimized, thread-safe, mathematical functions for engineering, scientific, and financial applications. The Cluster Edition contains all the Intel® Math Kernel Library functions plus ScaLAPACK (Scalable LAPACK)’ (<http://www.intel.com/cd/software/products/asmo-na/eng/perflib/mkl/index.htm>)

D. Graphically Oriented Concept Mapping

What do we do when neither words nor numbers seems to help solve our problems? Let us see:

Professor Challenger has devised means for getting us on to this plateau when it appeared to be inaccessible; I think that we should now call upon him to use the same ingenuity in getting us back to the world from which we came. . . . The problem of the descent is at first sight a formidable one, said he, and yet I cannot doubt that the intellect can solve it. I am prepared to agree with our colleague that a protracted stay in Maple White Land is at present inadvisable, and that the question of our return will soon have to be faced. I absolutely refuse to leave, however, until we have made at least a superficial examination of this country, and are able to take back with us something in the nature of a chart.

— Arthur Conan Doyle, *The Lost World*, Chapter 11.

Just as the people in the situation above, we may try to clarify our thoughts by writing down the concepts cropping in our mind, linking them, and studying the ramifications of those connections; we may simply want to have ‘something in the nature of a chart.’

Charts and mappings of concepts and variables routinely appear in business and economic works (research articles, business presentations, and so on). And their importance can be ascertained from the fact that Eden’s [50] ‘Cognitive Mapping’ has become one of the 30 most ‘influential’ articles published in *The European Journal of Operational Research* since its foundation (1975).

Objects like mapping, concept analysis, category formation, diagrammatic reasoning, conceptual spaces and mappings, and mappings in knowledge representation can be found in computer science, especially artificial intelligence and database management (particularly in the design of network, hierarchical, and relational logical models), and in related mathematical areas. Furthermore, modern cognitive psychology studies conceptual and idea mapping in a very comprehensive manner that includes – for example, in Eysenck and Kane [51] – mental representations, objects, concepts, categories, relations, events, schemas, and scripts (short summaries in natural language of a problem to be solved). Mappings of concepts are also important when we solve problems with analogical reasoning. Mayer [52] notes that ‘analogical reasoning occurs when we abstract a solution strategy from a previous problem and relate that information to a new problem we are to solve,’ [52, p. 415]. Analogical problem solving exploits structural similarities between problems and situations and may use a variety of procedures such as models, worked-out examples and cases, protocols, and graphical representations. Successful analogical transfer requires that the problem solver (a) recognize the potential of comparing problems, (b) be able to abstract the similarities and commonality in the structures, and (c) carry out the appropriate mapping from the components of the base (solved) problem into the (new) target problem. Analogs are problems that have the same underlying structure as the new problem but not the same ‘surface characteristics.’

Conceptual charts and mappings are already omnipresent in our daily affairs. For example, diagramming is common in critical thinking analysis and argumentative writing, software design and menu-navigation techniques, presentation tools, database management, and also internet searches, where they show not only their promise but also their efficiency – one only has to search for ‘granular computing’ or ‘asset allocation,’ on www.kartoo.com.

Eden [50] shares that he became interested in cognitive mapping when he noticed that his ‘main tools,’ simulation and dynamic programming, ‘never had any substantial impact on organizational life.’ However, his most successful and useful projects for his clients were ‘those where the modeling technique was simple and consequently transparent to the client, and seemed to organize his/their thinking rather than suggesting a course of actions.’ The following reflection of Eden reminds us of those expressed by Zadeh:

Out of the first few years of pondering grew cognitive mapping as the first strands of a reflective OR [Operational Research] practice. . . . The basis of this development was, to me then, a profound (sic!) discovery that managers think and work for most of their lives with language and ideas

not with numbers and mathematical symbols. Therefore, OR should be making a model building contribution to the way managers work with ideas. If OR models of any sort were going to have any real impact on the decisions they made they must gently shift the way in which the manager sees his decision making world (Eden [50, pp. 1–2]).

We would like to highlight the potential of a particular type of charts, fuzzy cognitive maps (FCM), in finance and economics. FCM have been studied by Kosko [53 and references therein]. They include features and ideas from fuzzy sets, neural network representations, graph theory, lattice theory, and dynamic equations. The distinctive feature is that an FCM is a concept chart where the concepts are numerically, in addition to logically, linked. They can be set up as a set of equations representing the relations and links in the problem, into which we can input initial data and iterate (as in Jacobi's method, for example) until we gather clues about how solutions may form. Fuzzy cognitive maps are well suited to represent certain types of neural networks (connectionist and causal). In our opinion, FCM may be helpful to represent mostly qualitative problems that, nonetheless, also include clearly defined numerical indicators. Fuzzy cognitive maps are, therefore, part of the hybrid, words-plus-numbers characteristic of granular computing. Many strategic analysis and financial counseling problems seem to require a hybrid approach.

E. The Readily Available Treasure Chest of Interval Tools, Techniques, and Methods

This subsection lists tools, techniques, procedures, and methods contained in the four methodologies presented. One purpose is to show that there are many available interval techniques. We also want to show how these specific interval techniques bridge numbers-only methods (rightmost column) and words-only methods (leftmost column). The center columns list interval methods which, depending on the specific problem, may, or may not need, be 'fuzzy.' Granular computing would incorporate several headings across columns, given its interdisciplinary, hybrid nature – see the plurality of methodologies represented in Wang [54] and in Bargiela and Pedrycz [55].

It is now appropriate to provide references to some of the techniques in Exhibit 51.3. In the next section, we will provide additional references specifically focused on finance and economics not mentioned here.

Exhibit 51.3 Interval analysis: bridge between words and numbers

Words	→	Interval	Interval (fuzzy)	←	Numbers
<hr/>					
Propositional calculus					
Algebra of sets					
Symbolic logic		Relational equations			Simultaneous equations
		Approximate equations			
		Interval mathematics			Math. programming & Calculus + linear algebra
			Fuzzy methods		
			Fuzzy numbers		Math. programming
			Fuzzy math programming		
			Fuzzy multiattribute		Math. programming
			Fuzzy multicriteria		Math. programming
Combinatorial optimization					
Integers					
Binary variables					
			Possibility theory		Probability theory
			Possibilistic programming		Math. programming
			Linguistic variables		
Words					
Content analysis					
Venn diagrams		Cognitive maps	Fuzzy cognitive maps		Lattice and graph theory

Fuzzy sets and related methods:

- Comprehensive references: Pedrycz and Gomide [3] Klir and Yuan [18], Zimmerman [56], and Wang [54], especially the contributions by Zadeh on the transition from computing with numbers to computing with words, Meystel and Wang on computing with words, and Dubois, Hadj-Ali, and Prade on fuzzy qualitative reasoning with words.
- Business decision making and fuzzy mathematical programming: Zimmerman [56], Lai and Hwang [57].
- Multiattribute and multicriteria decision problems: Zhang and Pu [58], Gurnani et alia [59], See and Lewis [60], Ma et alia [61], Greco et alia [62], Pawlak [63], Fuller and Carlson [64], Zimmerman [56 and references therein], Saaty [65], Jacquet-Lagreze and Siskos [66], Roy and Vincke [67], Freed and Glover [68].
- Rule-based fuzzy approach to business and industry: Von Altrock [69], Cox [70].
- Relational equations and fuzzy decision making: Tanaka [40], Peeva and Kyosev [41], and Baldwin and Pillsworth [71], which complements Bellman and Zadeh [25].
- Other useful references on fuzzy sets: Kosko and Isaka [72], Bonissone [73].

Interval analysis:

- Non-technical introduction: Hayes [74].
- Interval textbooks and monographs: Kuperman [44], Deif [43], Neumaier [47], Hansen [48].
- Application of approximate equations to selected finance and economics problems: Tarrazo [49].
- Interval software: Nenov and Fylstra [75]. Intel Corporation's [76] Reference Manual for the Intel® Math Kernel Library. A 2485-page compendium of interval analysis and computation in portable document format (16MB).

51.3 A Few Interval Contributions in Finance and Economics

Until the 1980s, one could find only a handful of research programs paving the way for interval analysis and granular computing in traditional finance and economics: (We are excluding the fuzzy-set-based contributions on mathematical programming referred to in the previous section.)

1. Lerner's Harvard seminars. One of them was entitled *Quantity and Quality* [77], and included interesting articles by Leontieff ('The Problem of Quality and Quantity in economics'), Spengler ('Quantification in Economics – its history'), and by Kemeny ('Mathematics Without Numbers'), where he reviews graph theory and balanced graphs, group transformations, networks, and logical choice.
2. Katzner's research programme, see *Analysis Without Measurement* [78] and *Unmeasured Information and the Methodology of Social Scientific Inquiry* [79].
3. Greenberg and Maybee's *Computer-Assisted Analysis and Model Simplification* [80], where some of the tools mentioned by Kemeny are explored.
4. Herbert Simon's research program.

The last entry is, of course, a very notable exception to the habit of using exact methods. Part of Simon's approach to research is succinctly described in his influential study 'A Behavioral Model of Rational Choice' [81, p. 96]: 'Broadly stated, the task is to replace the global rationality of economic man with a kind of rational behavior that is compatible with the access to information and the computational capacities that are actually possessed by organisms, including man, in the kinds of environments in which such organism exist.' The use of heuristics (simple ways to make decisions), and the models employed by people and organizations made for solutions that were approximate but good enough to get the job done. In other words, these decisions are still rational but are bounded by knowledge, information, and computing limitations. Simon can be regarded as the first granular computing researcher in economics/business.

Zadeh's reflection on modeling humanistic systems parallels Simon's observations on organizational behavior.

The following are examples of other contributions that stress the potential of intervals in economic and financial decision making. The contributions referenced below have been selected because (a) they study particularly challenging problems touched upon in the first part of the study, or (b) they represent direct applications of the methodologies presented in the second part.

1. Conventional methods, employing interval methods:

- Studies on quasiordering, semiorders, similarity relations, and individual preferences: Tversky [82], Rubinstein [83], Barbera et alia [84], and Kreps [85].
- Symbolic data analysis, Billard and Diday [86]. These authors integrate qualitative and quantitative data by first preprocessing information using the algebra of sets and other qualitative tools (e.g., clustering). The resulting information is more suitable to statistical processing (descriptive symbolic statistics, regression, and principal components). The methods studied by Billard and Diday would be necessary to apply statistical analysis to the individual investors of the financial planning example given when presenting relational equations (Section 2, Part B). Symbolic data analysis overlaps with concept analysis as studied in computer science.
- Graph theory exhibits a great deal of analytical versatility, which makes it ideal for integrating qualitative and quantitative analysis – see references to Kemeny and Greenberg and Maybee [80] above. Conyon and Muldoon [87] apply graph theory to the study of corporate boards. Tarrazo [88] uses graph theory to separate those securities to be held long (purchased) from those to be sold short.
- Simplified choice, subjective expected utility, and heuristics that work: Astebro and Elhedli [89], Hogarth and Karelaiia [90], Smith and Keeney [91], and Smith and Winterfeldt [92]. Schwarz [93] focuses on the psychological toll and stress that overexact decision-making protocols have on most people.

2. Interval mathematics techniques:

- Applications of interval techniques to economic problems (input–output analysis, hypothesis testing, global optimization of econometric functions, and automatic differentiation of economic functions: Jerrell [94–97] and Jerrell and Campione [98].
- Application of data envelopment analysis with interval data, Jablonsky et alia [99].
- Applications of approximate equations to financial planning, macroeconomic modeling, and portfolio optimization (Tarrazo [49, 100]); application of switching binary variables to short selling, Tarrazo and Alves [100].

3. Fuzzy-set-related methods:

- Buckley et alia [101] and Buckley [102] study on fuzzy equations in economics and finance: Leontief's open input–output problem, the internal rate of return, and a supply–demand model governed by a system of first-order ordinary differential equations.
- Ponsard's [103] review of fuzzy mathematical models in economics.
- Modeling individual decisions: time and decision making, Raha and Ray [104]; fuzzy treatments of preferences, Xu [105] and also Basu [106]; evaluation of cyclic transitive relations with fuzzy methods, De Baets et alia [107].
- Application of fuzzy sets to knowledge representation in business, Tarrazo [39]; investing, Lee et alia [108], Allen et alia [109], Shing and Nagasawa [110], and Tarrazo [111]; fuzzy relational equations models for individual financial planning, Tarrazo [38]; qualitative asset allocation, Tarrazo and Murray [112]; qualitative corporate financial planning [39].

- Corporate financial planning with fuzzy mathematical programming, Tarrazo and Gutierrez [113, 114]; a financial portfolio approach to inventory management, Corbett and Louri [115]; an inventory control model using fuzzy logic, Samanta and Al Araimi [116].
- Klir's [117] analysis of the concepts of information, uncertainty, and expectations in the work of G.L.S. Shackle.
- Efficiency analysis using fuzzy efficiency rankings, Kao and Liu [118].
- Application of fuzzy cognitive mapping to model the evaluation of information technology/systems infrastructure, Beskese et alia [119].

The previous list is not exhaustive: it reflects first the intention of this study to highlight the potential of interval methods in financial decision making, given our research interests and areas of expertise. Many other contributions can be found in the monographs mentioned in the previous section, as well as those journals issuing the largest amount of interdisciplinary publications on interval/granular computing methods, which include *Fuzzy Sets and Systems*, *The European Journal of Operational Research*, *Reliable Computing*, *Fuzzy Optimization and Decision Making*, and *The International Journal of Production Economics*, among many others. The area of decision sciences (e.g., *Management Science*, *Decision Sciences*) seems more receptive to new methods than are traditional finance journals.

As we indicated in the first part of this study, some of the major problems in economic and financial decision making require a hybrid analysis: that is, a methodology and computing techniques able to incorporate words and numbers. This objective may require the difficult task of contrasting alternative approaches, and/or integrating several methodologies. For example, Erol and Ferrell [120] study a procedure for integrating qualitative and quantitative data. The qualitative information is first handled via linguistic variables which, after a process of quantification, join numerical variables in a common goal programming framework. Righut and Kooths [121] apply fuzzy sets, neural networks, and genetic algorithms, within an artificial intelligence framework, to modeling exchange rate expectations. In particularly complex problems, the burden on the researcher is truly exceptional. Explaining the functioning of the stock market, for instance, requires formulating both realistically sophisticated agents and system properties while keeping the analysis computationally tractable. In order to do so, Tay et alia [122–124] develop artificial adaptive agents, whose expectations are formed via fuzzy inductive rules, and who are capable of exhibiting complex learning patterns, even in ill-defined market environments.

Fortunately, granular methodologies and interval methods offer the possibility of gaining considerable insight in important problems simply because the problem is approached in a more flexible way. Schjaer-Jacobsen [125], for example, compares gains using interval equations, fuzzy numbers (triangular and trapezoidal), and probability intervals in project evaluation. After a clear and concise presentation of each method, Schjaer-Jacobsen suggests that in some cases simple intervals (also known as ranges of uncertainty) may suffice to represent the unknown and make decisions; in some other cases, more complex depictions of ranges (e.g., trapezoidal fuzzy numbers) may be called for.

Echoing the difficulties we noted in the first section, Schjaer-Jacobsen motivates his study by asking: 'can it be made possible to establish by means of numerical calculations, a fairly useful and realistic picture of the economic consequences of strategic decisions despite the fact that little is known about the future?' [125, p. 91]. The answer is 'yes.' In our next section, we will show the role of intervals in strategy making – especially when little is known about the future.

51.4 Intervals, Risk Management, and Strategy

Modeling the subject and the object and dealing with the remaining void of unknown are the major modeling changes in each fundamental problem in finance, namely, consumer choice, production, corporate financial planning, portfolio optimization, and financial planning for individuals. In each case, there is always something we are aware of that, nonetheless, still escapes us and creates risky potential outcomes we must cover for. Consequently, a great deal of strategy implementation consists of applying risk management techniques. We mentioned diversification and insurance in the introduction to this study; we will focus on hedging in this section.

Hedging consists of taking contrary positions – the stock will go up, the stock will go down. If we did that by telling our broker to buy and sell the security in the same sentence, the broker would be happy to make some commissions but would also doubt our sanity. Of course, giving that order to our broker involves no risk but a certain loss in commissions. One major development in finance was the creation of derivative securities such as options, which are created by investors and give the buyer the right (the seller will have the obligation) to exercise his option to buy (calls) or sell (puts) the stock.

Assume a share of common stock sells at \$20. A call option may give the owner the right to buy at \$20 (option strike value) and the option contract itself may sell at \$2 (option premium). Obviously, if the stock does not go over \$22, it is not worthwhile to exercise the option. However, if the stock goes up to \$25, the owner of the call can buy the stock at \$20, sell it immediately, and enjoy a benefit of \$3 per share (profit = \$25 – \$2 – \$20). Now consider the case of a put option, which lets its owner sell stock at \$22 and costs \$2. If the stock stays at \$20, there is not much incentive to exercise the option because there are transaction costs. However, if the stock plummets to \$15, we could buy the stock right away and sell it to an unfortunate investor that bought it from us at \$22. This means we would make a profit of \$5 per share (profit = \$22 – 15\$ – 2 = \$5).

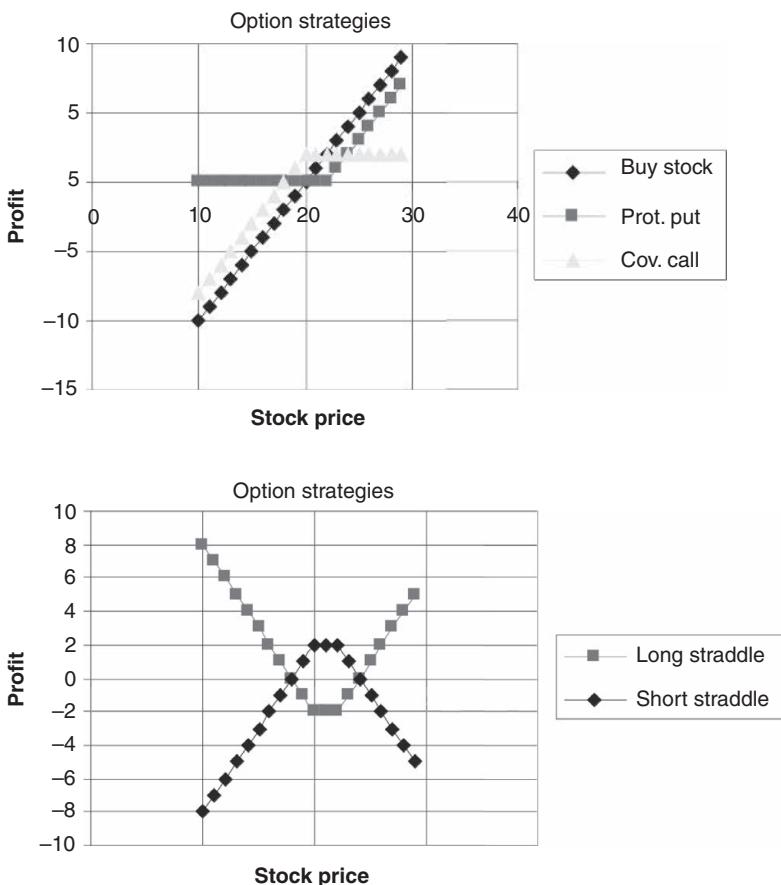
Each of the securities, if bought in isolation, carries risk: I buy stock because I think the stock can go up; the risk is the stock going down – the same with buying only a call option. Buying a put bets on the stock, which we do not own yet, going down; the risk is that the stock could go up. Options on the stock – also called derivative products of the stock – have been created to be the closest security to the stock itself, without being the stock itself. Hedging consists of taking contrary positions in closely related securities:

- (a) Protective put: buy stock and buy put.
- (b) Covered call writing: buy stock and write (sell) call.
- (c) Long straddle: buy calls and buy puts.
- (d) Short straddle: write (sell) a calls and sell (write) a puts.

Exhibit 51.4 shows two charts. The top one shows the profit profile for buying a stock only, for the protective put and the covered call. The ‘buy stock only’ is an upward sloping straight line. The profit profile for the hedged positions is calculated by adding up the profits from the stock position and from the option position.

The protective put shows a horizontal line if the stock goes down, which means the investor has eliminated downside risk completely. The covered call brings some money if the stock does not move (the call premium), leaves the investor exposed to downside risk, and also leaves room for making money if the stock goes up – a partial hedge. The bottom chart shows the profit profile of the long straddle, which means an investor, by buying options, bets something big is going to happen with the stock – best if it goes either way up or way down, to recover the money invested in premiums. The short straddle investor sells options. By moving away from options the investor is actually betting on ‘dead calm,’ markets where nothing happens. If this situation materializes, the investor simply keeps the premiums received. However, if the stock moves way up or way down, big losses may accrue.

Suppose you own a small hotel or restaurant in a coastal town. The fact that you actually own either one of these assets means you are actually invested in an asset with an implicit expectation: the weather will be good and many people will come to the coast. It also means that you are automatically exposed to a risk: the weather will be bad and nobody will come to your place. This type of risk affects many seasonal businesses (e.g., ski resorts) and very directly affects power companies, who may have contracted for the purchase of energy at a price and face a demand for energy that depends on the weather. Market participants in these products implement risk management strategies using intervals: if they stand to lose from their initial investment position, then they take an option position that would make money for them. Weather derivatives are very recent (late 1990s), but options and futures for farm-related goods, minerals, energy, foreign currencies, and financial products have been around for a long time. There are three major types of weather risk management products: (1) weather insurance contracts; (2) weather derivatives, which are privately negotiated, individualized agreements between two parties; and (3) weather futures and weather options on futures, both of which are traded publicly in an electronic auction setting at the

Exhibit 51.4 Derivatives hedging strategies and intervals

Chicago Mercantile Exchange. More information on these products can be found at the Web sites of the Chicago Mercantile Exchange and the Weather Risk Management Association.

Finally, note that these participants have been able to manage risk with little or no knowledge of the underlying process – the weather! The implications are far reaching: modern financial tools allow economic agents to protect themselves even if the underlying, risk-generating process is totally unknown.

51.5 Concluding Remarks

Intervals, by serving as a bridge between words and numbers, enable us to integrate and take advantage of information that numbers-only methods waste. By using exact models we are presuming a level of accuracy and knowledge (about the subject, the object, and about the problem itself) we rarely have. Interval methods, therefore, are steps in the right direction toward, at least, ridding our decision-making models from the false confidence exact methods imply. Furthermore, although many processes and events in the world may have quantitative representation (e.g., room temperature, stock quotes, musical sounds), what they mean to us, and the way we relate to them, may be primarily qualitative.

We have also shown how interval methods offer tools to protect ourselves from what we do not know. We will always be in need of strategic analysis because we will never know enough about the problems that worry us, and also because we never know what the future may bring. Our lack of knowledge causes

risk and contingencies for which we must compensate. Risk management strategies are implemented with intervals. With characteristic prescience, Keynes wrote [13, p. 297], ‘The object of analysis is not to provide a machine, or method of blind manipulation, which will furnish an infallible answer, but to provide ourselves with a method of thinking out particular problems.’ In sum, what matters is that by using hybrid methods we may be able to make headway despite imprecision, a lack of hard numbers, the imposing complexity of the problem, the lack of comprehensive planning models, and the insufficient knowledge of the user.

References

- [1] H. Simon. CMU’s Simon reflects on how computers will continue to shape the world. Interview by Byron Spice, Post-Gazette Science Editor, Monday, October 16, 2000. <http://www.post-gazette.com/regionstate/20001016simon2.asp>.
- [2] G. Klir. *Uncertainty and Information*. John Wiley and Sons, Hoboken, NJ, 2006.
- [3] W. Pedrycz and F. Gomide. *An Introduction to Fuzzy Sets*. The MIT Press, Cambridge, MA, 1998.
- [4] B. Kosko. *Fuzzy Engineering*. Prentice Hall, Upper Saddle River, NJ, 1997.
- [5] V. Kreinovich, H. Nguyen, and Y. Yam. Fuzzy systems are universal approximators for a smooth function and its derivative. *Int. J. Intell. Syst.* 15 (2000) 565–574.
- [6] L. Zadeh. From computing with numbers to computing with words – from manipulation of measurements to manipulation of perceptions. In: P. Wang (ed.), *Computing With Words*. John Wiley and Sons, New York, 2001, Chapter 2, pp. 13–34.
- [7] A. Marshall. *Principles of Economics*. Prometheus Books, Amherst, New York, 1997. (First published, 1890).
- [8] P. Samuelson. *Foundations of Economic Analysis*, Enlarged edition. Harvard University Press, Cambridge, 1983. (First published 1941, 2nd ed., 1948).
- [9] K. Arrow. *Social Choice and Individual Values*. Yale University Press, New Haven, London, 1963.
- [10] M. McManus. Transformations in economic theories. *Rev. Econ. Stud.* 25(2) (February 1958) 97–108.
- [11] H. Uzawa. Preference and rational choice in the theory of consumption. In: K. Arrow, S. Karlin, and P. Suppes (eds), *Mathematical Methods in the Social Sciences*, Stanford University Press, Stanford, 1960, pp. 129–148.
- [12] D. Luce. Semiorders and a theory of utility discrimination. *Econometrica* 24(2) (April 1956) 178–191.
- [13] J. Keynes. *The General Theory of Employment, Interest, and Money*. Prometheus Books, Amherst, New York, 1997. (First published in 1936).
- [14] F. Diebold. The past, present, and future of macroeconomic forecasting. *J. Econ. Perspect.* 12(2) (Spring 1998) 175–192.
- [15] I. Kant. *Critique of Pure Reason*. Translated by Norman Kemp Smith. St. Martin’s Press, New York, 1965. (First published in 1781).
- [16] M. Tarrazo. Schopenhauer’s prolegomenon to fuzzy sets. *Fuzzy Optim Decis. Mak.* 3(3) (September 2004) 227–254.
- [17] L. Zadeh. Fuzzy sets. *Inf. Control* 8, 3 (1965) 338–353. Yager et alia (eds) (1997). Klir and Yuan (1996).
- [18] G. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, PTR, Upper Saddle River, NJ, 1995.
- [19] G. Klir and B. Yuan (eds) Fuzzy sets, fuzzy logic, and fuzzy systems: the collected papers of Lotfi Zadeh. In: *Advances in Fuzzy Systems – Applications and Theory* 6. World Scientific Publishing, Singapore, 1996.
- [20] R. Yager, S. Ovchinnikov, R. Tong, and H. Nguyen (eds). *Fuzzy Sets and Applications: Selected Papers by L. A. Zadeh*. John Wiley and Sons, New York, 1987.
- [21] L. Zadeh. Shadows of fuzzy sets. In: *Problems in Transmission of Information* 2. Moscow, 1966, pp. 37–44. Klir and Yuan (1996).
- [22] L. Zadeh. Similarity relations and fuzzy orderings. *Infor. Sci.* 3, 2 (1971) 177–200. Yager et alia (eds). (1997).
- [23] L. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Sys. Man Cybern.* 3 (1973) 28–44. Yager et alia (eds) (1997).
- [24] L. Zadeh. A theory of approximate reasoning. In: J. Hayes, D. Michie, and L. Mikulich (eds.), *Mach. Intell.* 9, Halstead Press, New York, 1979, pp. 149–194. Yager et alia eds. (1997).
- [25] R. Bellman and L. Zadeh. Decision making in a fuzzy environment. *Manage. Sci.* 17(4) (1970) B141–B164. Also in R. Yager, S. Ovchinnikov, R. Tong, and H. Nguyen (eds). *Fuzzy Sets and Applications: Selected Papers by L. A. Zadeh*. John Wiley and Sons, Hoboken, NJ, 1987.
- [26] L. Zadeh. The linguistic approach and its application to decision analysis. In: Y. Ho and S. Mitter (eds.) *Directions in Large Scale Systems*. Plenum Press, New York, 1976, pp. 339–370. Klir and Yuan (1996).

- [27] L. Zadeh. Toward a theory of information granulation and its centrality in human reasoning. *Fuzzy Sets Syst.* 90(2) (1997) 111–127.
- [28] L. Zadeh. Fuzzy systems theory: A framework for the analysis of humanistic systems. In R. Cavallo (ed), *Recent Developments in Systems Methodology in Social Science Research*. Kluwer, Boston, 1981, pp. 25–41. Klir and Yuan (1996).
- [29] L. Zadeh. Linguistic characterizations of preference relations as a basis for choice in social systems. *Erkenntnis* 11(3) (1977) 383–410. Klir and Yuan (1996).
- [30] L. Zadeh. Fuzzy logic and approximate reasoning. *Synthese* 30 (1975) 407–428. Klir and Yuan (1996).
- [31] L. Zadeh. Fuzzy sets as the basis for a theory of possibility. *Fuzzy Sets Syst.* 1 (1978) 3–28. Reprinted in *Fuzzy Sets Syst.* 100 (Supplement): 9–34, 1999. Yager et alia (eds) (1997).
- [32] L. Zadeh. Fuzzy sets and information granularity. In: M. Gupta, R. Ragade, and R. Yager (eds.), *Advances in Fuzzy Set Theory and Applications*, North Holland, Amsterdam, 1979b, pp. 3–18. Klir and Yuan (1996).
- [33] L. Zadeh. A theory of commonsense knowledge. In: H.J. Skala, S. Termini, and E. Trillas (eds), *Aspects of Vagueness*, Reidel, Dordrecht, 1984, pp. 257–296. Yager et al (eds.) (1997).
- [34] L. Zadeh. Outline of a theory of usuality based on fuzzy logic. In: A. Jones, A. Kaufmann, and H. Zimmermann (eds), *Fuzzy Sets Theory and Applications*. Reidel, pp. 79–97, 1986. Klir and Yuan (1996).
- [35] L. Zadeh. A computational theory of dispositions. *Int. J. Intell. Syst.* 2 (1987) 39–63. Klir and Yuan (1996).
- [36] S. Langer. *An Introduction to Symbolic Logic*. Dover, New York, 1967.
- [37] M. Tarrazo. Calculating uncertainty intervals in approximate equations systems. *Appl. Numer. Math.* 26 (1998) 1–9.
- [38] M. Tarrazo. An application of fuzzy set theory to the individual investor problem. *Finan. Serv. Rev.* 6(2) (1997a) 97–107.
- [39] M. Tarrazo. A methodology and model for qualitative business planning. *Int. J. Bus. Res.* 3(1) (Fall 1997b) 41–62.
- [40] K. Tanaka. *An Introduction to Fuzzy Logic for Practical Applications*. Springer, New York, 1997.
- [41] K. Peeva and Y. Kyosev. *Fuzzy Relational Calculus: Theory, Applications, Software. Advances in Fuzzy Systems – Applications and Theory*, Vol. 22. World Scientific, Singapore, 2004.
- [42] R. Belohlavek. *Fuzzy Relational Systems*. Kluwer Academic, Dordrecht, 2002.
- [43] A. Deif. *Sensitivity Analysis in Linear Systems*. Springer-Verlag, New York, 1986.
- [44] I. Kuperman. *Approximate Linear Algebraic Equations*. Van Nostrand Reinhold Company, London, 1971.
- [45] R. Moore. *Interval Analysis*. Prentice Hall Series in Automatic Computation, Prentice Hall, Englewood Cliffs, NJ, 1966.
- [46] R. Moore. Methods and applications of interval analysis. *SIAM*, Philadelphia, 1979.
- [47] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, 1990.
- [48] E. Hansen. *Global Optimization Using Interval Analysis*. Marcel Dekker, New York, 1992.
- [49] M. Tarrazo. *Practical Applications of Approximate Equations in Finance and Economics*. Quorum Publishers, Greenwood Publishing Group, Westport, CT, 2001.
- [50] C. Eden. Cognitive mapping. *Eur. J. Oper. Res.* 36 (1) (1988) 1–13.
- [51] M. Eysenck and M. Keane. *Cognitive Psychology: A Student's Handbook*. Psychology Press, Hove, East Sussex, 1997.
- [52] R. Mayer. *Thinking, Problem-solving, Cognition*. W.H. Freeman and Company, New York, 1983.
- [53] B. Kosko. *Neural Networks and Fuzzy Systems: A Dynamic Approach to Machine Intelligence*. Prentice Hall, Englewood Cliffs, NJ, 1992.
- [54] P. Wang (ed.) *Computing With Words*. John Wiley and Sons, New York, 2001.
- [55] A. Bargiela and W. Pedrycz. *Granular Computing: An introduction*. Kluwer Academic Publishers, Dordrecht, 2003.
- [56] H. Zimmermann. *Fuzzy Set Theory and Its Applications*, 2nd. ed. Kluwer Academic Press, Dordrecht. 1991.
- [57] Y. Lai and C. Hwang. *Fuzzy Mathematical Programming and Applications*. Springer-Verlag, New York, 1993.
- [58] J. Zhang and P. Pu. *Survey of Multi-Attribute Decision Problems*. EPFL Technical Report No: IC/2004/54. Swiss Federal Institute of Technology, Lausanne, Switzerland, June 2004, pp. 1–14.
- [59] A. Gurnani, T. See, and K. Lewis. An approach to robust multi-attribute concept selection. In: *Proceedings of DETC 2003/DAC-48707*, Chicago, IL, September 2003, ASME 2003.
- [60] T. See and K. Lewis. Multi-attribute decision making using hypothetical equivalents. In: *Proceedings of DETC 2002/DAC-34079*, Montreal, Quebec Canada, September 2002, ASME 2002.
- [61] J. Ma, Z. Quan, Z. Fan, J. Liang, and D. Zhou. An approach to decision making based on preference information on alternatives. In: *Proceedings of the 34th Hawaii International Conference on System Sciences*, Maui, Hawaii, January 3–6, 2001.

- [62] S. Greco, B. Matarazzo, and R. Slowinski. Rough sets theory for multi-criteria decision analysis. *Eur. J. Oper. Res.* 129(1) (2001) 1–47.
- [63] Z. Pawlak. Rough sets and decision analysis. *INFOR* 38(3) (August 2000) 132–144.
- [64] R. Fuller and C. Carlson. Fuzzy multi-criteria decision making: Recent developments. *Fuzzy Sets Syst.* 78 (1996) 139–153.
- [65] T. Saaty. How to make a decision, the analytic hierarchy process. *Eur. J. Oper. Res.* 48(1) (1990) 9–26.
- [66] E. Jacquet-Lagreze, and J. Siskos. Assessing the set of additive utility functions for multi-criteria decision making, the UTA method. *Eur. J. Oper. Res.* 10(2) (1986) 151–164.
- [67] B. Roy and P. Vincke. Multi-criteria analysis: Survey and new directions. *Eur. J. Oper. Res.* 8(3) (1981) 207–218.
- [68] N. Freed and F. Glover. Simple but powerful goal programming models for discriminating problems. *Eur. J. Oper. Res.* 7(1) (1981) 44–60.
- [69] C. Von Altrock. *Fuzzy Logic and Neurofuzzy Applications in Business and Finance*. Prentice Hall PTR, Upper Saddle River, NJ, 1997.
- [70] E. Cox. *Fuzzy Logic for Business and Industry*. Charles River Media, Rockland, MA, 1995.
- [71] J. Baldwin and B. Pilsworth. Dynamic programming for fuzzy systems with fuzzy environment. *J. Math. Anal. Appl.* 85 (1982) 1–23.
- [72] B. Kosko and S. Isaka. Fuzzy logic. *Sci. Am.* (July 1993) 76–81.
- [73] P. Bonissone. A fuzzy sets based linguistic approach: theory and applications. In: M.M. Gupta and E. Sanchez (eds). *Approximate Reasoning in Decision Analysis*. North Holland, New York, 1982.
- [74] B. Hayes. A lucid interval. *Am. Sci.* 1(6) (November–December 2003) 484–488.
- [75] I. Nenov and D. Fylstra. Interval methods for accelerated global search in the microsoft excel solver. *Reliable Comput.* 9 (2003) 143–159.
- [76] Intel Corporation (2006). Intel® math kernel library. *Reference Manual*. <http://www.intel.com/cd/software/products/asmo-na/eng/perflib/mkl/index.htm>, accessed January 1, 2008.
- [77] D. Lerner. *Quality and Quantity*. The Free Press of Glencoe, Inc., New York, 1961.
- [78] D. Katzner. *Analysis without Measurement*. Cambridge University Press, Cambridge, 1983.
- [79] D. Katzner. *Unmeasured Information and the Methodology of Social Scientific Inquiry*. Kluwer Academic Publishers, Dordrecht, 2001.
- [80] H. Greenberg and J. Maybee. *Computer-Assisted Analysis and Model Simplification*. Academic Press, New York, 1981.
- [81] H. Simon. A behavioral model of rational choice. *Q. J. Econ.* 69 (1995) 99–118. The full electronic text of all his writings are available at Carnegie-Mellon University library: <http://diva.library.cmu.edu/Simon/>.
- [82] A. Tversky. Features of similarity. *Psychol. Rev.* 84(4) (July 1977) 327–352.
- [83] A. Rubinstein. Similarity and decision-making under risk (is there a utility theory resolution to the Allais paradox?). *J. Econ. Theory* 46 (1988) 145–153.
- [84] S. Barbera, W. Bossert, and P. Pattanaik. Ranking sets of objects. In: S. Barbera, P. Hammond, and C. Seidl (eds) *Handbook of Utility Theory: Volume 2, Extensions*. Kluwer Academic Publishers, Dordrecht, 2004, Chapter 17.
- [85] D. Kreps. A representation theorem for ‘preference for flexibility.’ *Econometrica* 47(3) (May 5, 1979) 65–578.
- [86] L. Billard and E. Diday. *Symbolic Data Analysis: Conceptual Statistics and Data Mining*. John Wiley & Sons, Hoboken, NJ, 2007.
- [87] M. Conyon and M. Muldoon. The small world of corporate boards. *J. Bus. Finan. Account.* 33(9–10) (2006) 1321–1343.
- [88] M. Tarrazo. Identifying securities to buy and sell: the heuristic ri/stdi. In: *Meeting of the Decision Sciences Institute*. November 18–21, 2006, San Antonio, Texas.
- [89] T. Astebro, and S. Elhedli. The effectiveness of simple decision heuristics: Forecasting commercial success for early-stage ventures. *Manage. Sci.* 52(3) (March 2006) 395–409.
- [90] R. Hogarth and N. Karelaia. Simple models for multiattribute choice with many alternatives: When it does and does not pay to face trade-offs with binary attributes. *Manage. Sci.* 51(12) (December 2005) 1860–1872.
- [91] J. Smith and R. Keeney. Your money or your life: A prescriptive model for health, safety, and consumption decisions. *Manage. Sci.* 51(9) (November 2005) 1309–1325.
- [92] J. Smith and D. Winterfeldt. Decision analysis in management science. *Manage. Sci.* 50(5) (May 2004) 561–574.
- [93] B. Schwartz. *The Paradox of Choice: Why More Is Less*. Harper Perennial, New York, 2004.
- [94] M. Jerrell. Interval arithmetic for input-output models with inexact data. *Comput. Econ.* 10(1) (February 1997a) 89–100.
- [95] M. Jerrell, Automatic differentiation and interval arithmetic for estimation of disequilibrium models. *Comput. Econ.* 10(3) (August 1997b) 295–316.

- [96] M. Jerrell. Applications of interval computations to regional economic input-output models. In: R. Kearfoot and V. Kreinovich (eds), *Applications of Interval Computations*, Vol. 3 of *Applied Optimization*, Kluwer Academic Publishing, Dordrecht, 1996, pp. 133–142.
- [97] M. Jerrell. Computer programs to demonstrate some hypothesis-testing issues. *Am. Stat.* 42(1) (February 1988) 80–81.
- [98] M. Jerrell and W. Campione. Global optimization of econometric functions. *J. Glob. Optim.* 20(3–4) (August 2001) 273–295.
- [99] J. Jablonsky, Y. Smirlis, and D. Despotis. DEA with interval data: An illustration using the evaluation of branches of a Czech Bank. *Cent. Eur. J. Oper. Res.* 12(4) (December 2004) 323.
- [100] M. Tarrazo and G. Alves. Portfolio optimization under realistic short sales conditions. *Int. J. Bus.* 3(2) (April 1998) 77–93.
- [101] J. Buckley, E. Eslami, and T. Feuring. *Fuzzy Mathematics in Economics and Engineering. Studies in Fuzziness and Soft Computing*, Vol. 91. Physica, Heidelberg and New York, 2002, pp. xi, 272.
- [102] J. Buckley. Solving fuzzy equations in economics and finance. *Fuzzy Sets Syst.* 48 (1992) 289–296.
- [103] C. Ponsard. Fuzzy mathematical models in economics. *Fuzzy Sets Syst.* 28 (1988) 273–283.
- [104] S. Raha and K. Ray. Approximate reasoning with time. *Fuzzy Sets Syst.* 107 (1999) 59–79.
- [105] Z. Xu. On compatibility of interval fuzzy preference relations. *Fuzzy Optim. Decis. Mak.* 3 (2004) 217–225.
- [106] K. Basu. Fuzzy revealed preference theory. *J. Econ. Theory* 32 (1984) 212–227.
- [107] B. De Baets, H. De Meyer, B. De Schuymer, and S. Jenei. Cyclic evaluation of transitivity of reciprocal relations. *Soc. Choice Welfare* 26 (2006) 217–238.
- [108] C.F. Lee, G. Tzeng, and S. Wang. A fuzzy set approach for generalized CRR model: An empirical analysis of S&P 500 index options. *Rev. Quant. Financ. Account.* 25(3) (November 2005) 255–275.
- [109] J. Allen, S. Bhattacharya, and F. Smarandache. Fuzziness and funds allocation in portfolio optimization. *Int. J. Soc. Econ.* 30(5–6) (2003) 619–632.
- [110] C. Shing and H. Nagasawa. Interactive decision system in stochastic multiobjective portfolio selection. *Int. J. Prod. Econ.* 60–61 (1999) 187–193.
- [111] M. Tarrazo. Fuzzy sets and the investment decision. (Technology issue of the Journal of Investing), *Finan. Technol.* June 1999, 37–47.
- [112] M. Tarrazo and W. Murray. Teaching what we know about asset allocation. *Adv. Finan. Educ.* 2 (Spring 2004) 77–103.
- [113] M. Tarrazo and L. Gutierrez. Financial planning, expectations and fuzzy sets. *Eur. J. Oper. Res.* 126(1) (August 2000) 89–105.
- [114] M. Tarrazo and L. Gutierrez. *Perspectives on Financial Planning. Research Papers in Management and Business*. Ecole Supérieure de Commerce, Montpellier-ESKA Editions, Paris, France, 1997, pp. 61–79.
- [115] J. Corbett, D. Hay, and H. Louri. A financial portfolio approach to inventory behavior: Japan and the U.K. *Int. J. Prod. Econ.* 59 (1999) 43–52.
- [116] B. Samanta and S. Al-Araimi. An inventory control model using fuzzy logic. *Int. J. Prod. Econ.* 73 (2001) 217–226.
- [117] G. Kfir. Uncertainty in economics: The heritage of G. L. S. shackle. *Fuzzy Econ. Rev.* 7(2) (November 2002) 3–21.
- [118] C. Kao and S. Liu. A mathematical programming approach to fuzzy efficiency ranking. *Int. J. Prod. Econ.* 86 (2003) 145–154.
- [119] A. Beskese, C. Kahraman, and Z. Irani. Quantification of flexibility in advanced manufacturing systems using fuzzy concept. *Int. J. Product. Econ.* 89(1) (May 2004) 45–56. Special Issue.
- [120] I. Erol and W. Ferrell. A methodology for selection problems with multiple, conflicting objectives and both qualitative and quantitative criteria. *Int. J. Prod. Econ.* 86 (2003) 187–199.
- [121] E. Ringhut and S. Kooths. Modeling expectations with GENEFER – An artificial intelligence approach. *Comput. Econ.* 21(1–2) (April 2003) 173–194.
- [122] N. Tay and S.C. Linn. Complexity and the character of stock returns: empirical evidence and a model of asset prices based upon complex investor learning. *Manage. Sci.* 53(7) (July 2007) 1165–1180.
- [123] N. Tay and S.C. Linn. Fuzzy inductive reasoning, expectation formation, and the behavior of security prices. *J. Econ. Dyn. Control* 25 (2001) 321–361.
- [124] N. Tay and R.F. Lusch. A preliminary test of hunt's general theory of competition: Using artificial adaptive agents to study complex and ill-defined environments. *J. Bus. Res.* 58(9) (2005) 1155–1168.
- [125] H. Schjaer-Jacobsen. Representation and calculation of economic uncertainties: intervals, fuzzy numbers, and probabilities. *Int. J. Prod. Econ.* 78(1) (July 2002) 91–98.

52

Granular Computing Methods in Bioinformatics

Julio J. Valdés

52.1 Introduction

The science of biology is very different from what it was two decades ago. It has become increasingly multidisciplinary, especially after the unprecedented changes introduced by the human genome project. This effort projected a new vision of biology as an information science, thus making biologists, chemists, engineers, mathematicians, physicists, and computer scientists cooperate in the development of mathematical and computational tools, as well as high throughput technologies. The entire field of biology is changing at an enormous rate, like a handful of other disciplines, exerting a boosting effect on the development of many existing technologies and inducing the creation of new ones. As a result, a large impact in human society is expected, with unforeseen consequences. In fact, many specialists and analysts believe that it will change forever the way in which modernity is understood.

Bioinformatics is an emerging and rapidly growing field for which no universally accepted definition may be found. In its broadest sense, it covers the use of computers to handle biological information, understood as the use of applied mathematics and computer science to solve biological problems. Accordingly, it covers a large body of both theoretical and applied methods, with implications in medicine, biochemistry, and many other fields in the life sciences domain. In the same sense, it involves many areas of mathematics (ranging from classical analysis and statistics, to probability theory, graph theory, etc.) and computer science, like automata theory and artificial intelligence, just to mention a few. Bioinformatics research considers general topics like systems biology or modeling of evolution, and more specific ones like gene expression intensities, protein–protein interactions, and other biological problems at the molecular scope. It is common to interpret bioinformatics as computational biology; however, they are recognized as separate fields, although very related and overlapping. According to a Committee of the National Institute of Health, bioinformatics is oriented to the research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral, or health data, including those to acquire, store, organize, archive, analyze, or visualize such data; whereas computational biology focuses on the development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques for the study of biological, behavioral, and social systems. Regardless of whether bioinformatics is interpreted in a broad or narrow sense, a common denominator is the processing of large amounts of biologically derived information, whether DNA sequences or breast X-rays. Machine learning techniques oriented to bioinformatics

in combination with other mathematical techniques (mostly from probability and statistics) are presented in [1]. Within bioinformatics there is a broad spectrum of problems requiring classification, discovery of relations, finding relevant variables, and many others, where granular computing approaches can be applied. Moreover, there are other issues like the characterization and processing of uncertain information and the handling of incomplete data, where rough sets and fuzzy set approaches are particularly appropriate. Granular computing provides a broad range of approaches for the analysis of biological data. Fuzzy and rough-set-based techniques, in particular, are specially suited for handling uncertainties of different kinds. Moreover, within their framework many powerful procedures have been developed for clustering, classification, feature selection, and other important data mining tasks. The rough sets approach is particularly well suited to bioinformatics applications because of its ability to model from uncertain, approximate, and inconsistent data. The generated rule models are easy to interpret by non-experts and they are also minimal in the sense of not using redundant attributes. These techniques have been applied to a wide variety of biological problems and in more recent years to bioinformatics in the genomic and postgenomic era, but bioinformatics textbooks are not yet covering them regularly [2, 3].

In fact, the number of applications of granular computing techniques to bioinformatics is becoming large and is constantly increasing. It is impossible to cover all of these developments here; therefore, only selected topics and examples are presented. The purpose of this chapter is to illustrate the scope, possibilities, and future of the application of granular computing approaches in the domain of modern bioinformatics.

52.2 Genomics: Gene Expression Analysis

Considered now as classical are the tasks of storing, comparing, retrieving, analyzing, predicting, and simulating the structure of biomolecules (including genetic material and proteins). Most large biological molecules are polymers, composed of ordered chains of simpler molecular modules (monomers) which can be joined together to form a single, larger macromolecule. Macromolecules can have specific informational content and/or chemical properties and the monomers in a given macromolecule of DNA or protein can be treated computationally as letters of an alphabet. In specific arrangements, they carry messages or do work in a cell. This explains why, from the mathematical point of view, the interest was concentrated on sequence analysis.

After the completion of the Human Genome Project in 2003, the focus and priorities of bioinformatics started to change rapidly. Actually, they are constantly changing and several new streams within bioinformatics have emerged.

Genomics is the study of genes and their function. The genome is the entire set of hereditarily obtained instructions for building, running, and maintaining an organism, also used for passing life on to the next generation. The genome is made of a molecule called DNA and it contains genes, which are packaged in units called chromosomes and affect specific characteristics of the organism. In *comparative genomics*, multiple genomes are investigated for differences and similarities between the genes of different species. These studies have led to both specific conclusions about species and general considerations about evolution itself.

The identification of gene functions on a large scale and the discovery of their associations are of great importance, which is the purpose of *functional genomics*. The set of proteins encoded by the genome is known as the proteome. The study of the proteome is the domain of *proteomics*, which includes not only all the proteins in any given cell, but also the set of all protein forms and modifications, their interactions, and the structural description of both the proteins and their higher order complexes. The characterization of the many tens of thousands of proteins expressed in a given cell type at a given time involves the storage and processing of very large amounts of data. It is natural that artificial intelligence techniques, in general, and machine learning, in particular, find broad application in bioinformatics because of the need to speed up the process of knowledge discovery. In this sense, data mining on the constantly growing bioinformatics databases is possibly the only way to achieve that goal.

One of the most important fields of modern bioinformatics where granular computing methods have a large potential and where successful applications have already been made is genomics; in particular, the

analysis of DNA microarrays. DNA is the molecule that encodes genetic information. In eukaryotes (all organisms except viruses, bacteria, and bluegreen algae), it is a double-stranded molecule held together by weak bonds between base pairs of nucleotides, namely, adenine (A), guanine (G), cytosine (C), and thymine (T). Base pairs form between A and T and between G and C; thus, the base sequence of each single strand can be obtained from that of the other. RNA is the molecule found in the nucleus and cytoplasm of cells, and it plays an important role in protein synthesis and other chemical activities of the cell. The structure of RNA is related to that of DNA. There are several RNA molecules: messenger RNA, transfer RNA, ribosomal RNA, and others.

According to what is considered the central dogma of biology (Figure 52.1), DNA experiences a process called *transcription* which is the synthesis of an RNA copy from a sequence of DNA (a gene). From the RNA (actually, from the messenger RNA which is the one that serves as a template), a process called *translation* occurs, in which the genetic code carried by mRNA directs the synthesis of proteins from amino acids.

The cell determines through interactions among DNA, RNA, proteins, and other substances when and where genes will be activated and how much gene product (e.g., a protein) will be produced. (The process is called gene regulation.) In this process, genes are activated to produce the specific biological molecule encoded by them (gene expression) following very complex patterns of interactions. Traditionally, molecular biology experiments studied the behavior of an individual gene, thus obtaining a very limited amount of information and missing the more complex picture given by the interrelations of different genes and their functions.

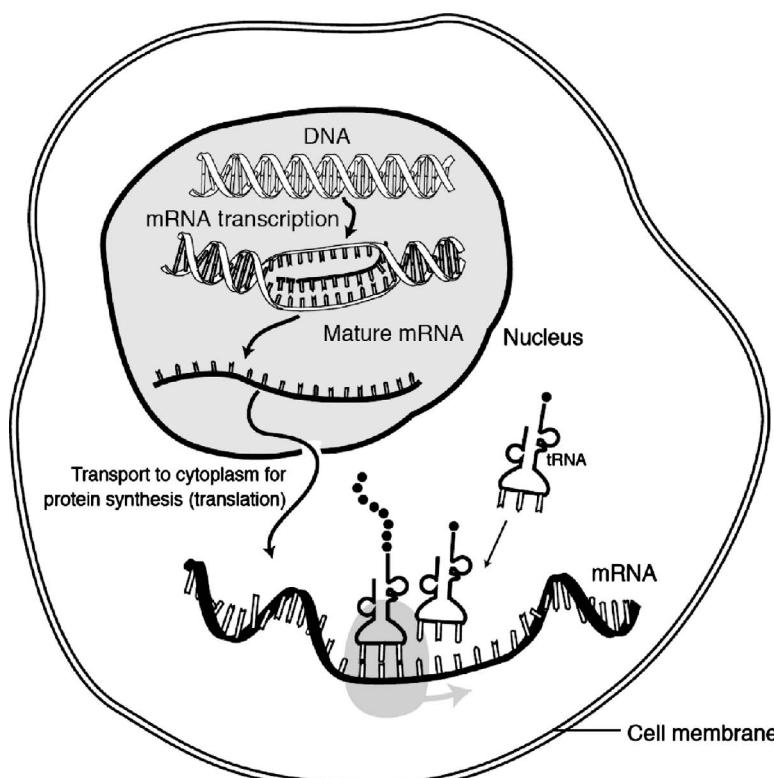


Figure 52.1 The central dogma of biology. DNA leads to mRNA via transcription and then to proteins via translation

Recently, a new technology, called a DNA microarray, has been developed which has attracted tremendous interest among biologists. It allows the study of the behavior of large numbers of genes simultaneously, which potentially can cover the whole genome on a single chip. In this way, researchers can have a broad picture of the interactions among thousands of genes simultaneously [4–7].

Complementary DNA (cDNA) is single-stranded DNA made in the laboratory from a messenger RNA template. It represents the parts of a gene that are expressed in a cell to produce a protein. Often it is used as a probe in the physical mapping of a chromosome. A DNA microarray is a glass slide with cloned cDNA in spots deposited on its surface at fixed locations according to a previously designed layout, in an operation usually controlled by a robotic arm. Target mRNA from two different samples (test and control) are labeled with fluorescent dyes Cy5 and Cy3 (red and green respectively) which hybridize complementary DNA (cDNA). The mRNA degrades and the resulting mixture of cDNA from the test and control samples is applied to the microarray where some strand binds to their complementary probe strands after some time. The plate is washed in order to remove the strands which did not bind to any of the existing spots. Then, the microarray is placed in a black box and scanned with red and green lasers producing two images where the intensity of each spot is proportional to the concentration of mRNA. For each spot a ratio of residual intensity of each of the dyes (i.e., removing the background intensity of the corresponding dye) is computed. Thus, what is obtained is a measure of relative abundance between the two samples. Typically, many thousands of spots can be placed on a microarray surface and (even considering that in many experimental designs duplicate spots are placed), thousands of genes can be studied simultaneously (Figure 52.2).

One common use of microarrays is to determine which genes are activated and which genes are inhibited when two populations of cells are compared. This technology has been used in gene discovery, disease diagnosis, drug discovery (this field is called pharmacogenomics), toxicological research (the field of toxicogenomics), and other biomedical tasks.

Considering the costs involved in producing the microarrays and in conducting the experiments, the typical situation is that of having a relatively small number of objects (experiments) in comparison with the number of attributes describing each of them (genes). Depending on the particular situation, the objects may or may not have labels representing a disease, type of tumor, etc., leading to supervised or unsupervised problems involving classification and/or clustering.

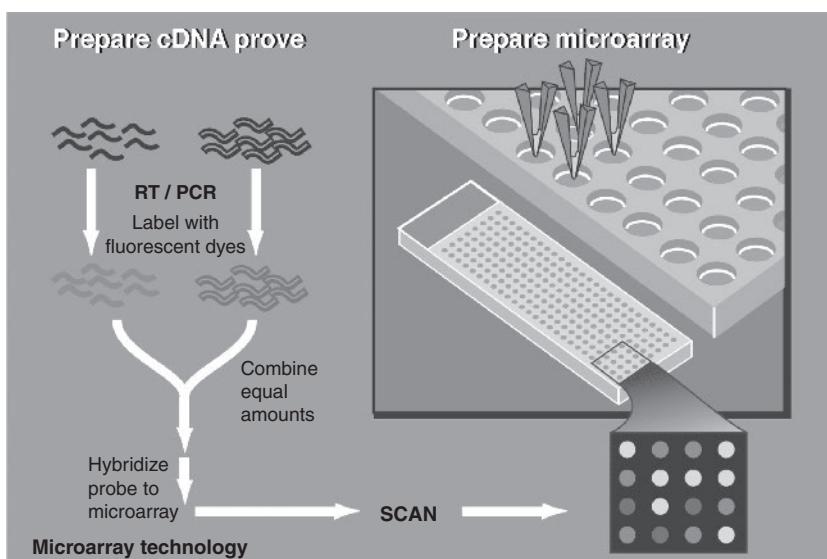


Figure 52.2 Sample preparation for a microarray technology experiment

52.2.1 Fuzzy Methods in Genomics

There are several advantages of applying fuzzy logic to the analysis of gene expression data. Fuzzy logic inherently accounts for noise in the data, as they are understood as categories of a linguistic variable, with gradual boundaries in between. In contradistinction with other algorithms like neural networks, support vector machines (SVM), or elaborated statistical procedures, fuzzy logic results can be communicated and understood with ease to domain experts (biologists, physicians, etc.). Also, fuzzy procedures are computationally fast and efficient.

A relatively simple fuzzy logic approach for the analysis of gene expression data is that of [8], where expression data values were normalized to a [0,1] range and then fuzzified by creating a linguistic variable with three categories (low, medium, and high). Triangular membership functions were used for describing each category with cross-points at 0.5 membership values. Using yeast cell cycle expression data [9], triplets of gene expression values were defined (all taken at the same time point in the yeast growth cycle time series). Nine rules assembled as a decision matrix were defined, each composed of an elementary conjunction of two attribute-value pairs corresponding to the first two genes of the triplet and an attribute-value pair of the third gene in the triplet as the consequent (the value of a given cell in the decision matrix). The rules were formulated with the assumption that the first conjunct is an activator gene and the second a repressor.

Then, an exhaustive search algorithm analyzed the triplets by applying the rules, and comparing the fuzzy predicted value for the third gene of the triplet. The squared difference between the observed value and the defuzzified predicted value was computed and triplets with values smaller than a given threshold (0.015) were accepted. In addition, the variance of the number of hits corresponding to the cells in the decision matrix was computed and used as a second filtering criterion. Some very interesting triplets were found, and in particular those involving the *HAPI* gene were followed up. By assembling the corresponding triplets, a regulatory network was assembled. The predicted network was highly consistent with the experimental data obtained from previous studies. Also, many of the most frequently found pairs of genes appeared to be biologically relevant.

Genes are related in very complex ways and the discovery of their relations is an important goal of gene expression microarray data processing. From an unsupervised perspective, a classical approach has been looking for groups of genes which behave similarly with respect to some predefined measure of similarity or distance using cluster analysis methods. Among them, hierarchical clustering and k -means partitioning cluster are typically applied. However, the crisp nature of the partitions created by hierarchical methods where a similarity or distance value is used as the threshold for group separation is a great limitation. The same happens with k -means methods, where the number of groups to construct has to be fixed in advance. This is especially problematic when analyzing large gene-expression data sets that are collected over many experimental conditions, when many of the genes are likely to be similarly expressed with different groups in response to different subsets of the experiments.

Fuzzy clustering, [10–13] on the other hand, facilitates the identification of overlapping groups of objects by allowing each element to belong to more than one group. The essential difference is that rather than the hard partitioning of standard k -means clustering, where genes belong only to a single cluster, fuzzy clustering considers each gene to be a member of every cluster, with a variable degree of membership. In classical k -means clustering the only parameter to specify is the desired number of clusters (k). However, in fuzzy clustering yet another parameter (m) must be indicated. It controls the fuzziness of the constructed partitions. When m is 1, the result is a hard (crisp) partition like the one produced by classical k -means. The larger the m , the ‘fuzzier’ the resulting partitions are going to be.

There are many variants of the fuzzy c -means algorithms [12, 14, 15] and they have been applied to the analysis of gene expression data. An interesting modification to the Gath and Geva algorithm was introduced in [16] and used for exploring the conditional coregulation in yeast gene expression data. The algorithm was modified in two ways: (i) three successive cycles of fuzzy k -means clustering are performed, with the second and third rounds of clustering operating on subsets of the data; (ii) each clustering cycle is initialized by seeding prototype centroids with the eigen vectors identified by principal component analysis (PCA) of the respective data set. (This is done in order to attenuate the impact of random initialization on the results.)

The first round of clustering is initialized by defining $k/3$ prototype centroids (where k is the total number of clusters and 3 is the number of clustering cycles) as the most informative $k/3$ eigen vectors identified by PCA of the input data set. In the subsequent steps the prototype centroids are refined by assigning to each gene a membership to each of the prototype centroids, based on the Pearson correlation between the gene's expression pattern and the given centroid. Then the centroids are recalculated as the weighted mean of all of the gene-expression patterns in the corresponding group, where each gene's weight is proportionate to its membership in the given cluster. The process is iterated until the centroids become stable. Once this round of fuzzy clustering is performed, centroid pairs whose Pearson correlation is greater than 0.9 are considered duplicate and are averaged. Then, genes with a correlation greater than 0.7 to any of the identified centroids are removed from the data set.

These steps are repeated on this smaller data set to identify patterns missed in the first clustering cycle, and the new centroids are added to the set identified in the first round. The process of averaging replicated centroids and selecting a data subset is repeated, and the third cycle of clustering is performed on the subset of genes with a correlation of less than 0.7 to any of the existing centroids. The newly identified centroids are combined with the previous sets, and replicate centroids are averaged. As a final step, the membership of each gene to each centroid is computed.

When applied to 93 published microarray experiments involving 6200 yeast genes, it was found that this kind of fuzzy clustering method was able to produce clusters of genes that were not identified by hierarchical or classical (crisp) k -means clustering. In addition, it also provided more comprehensive clusters of previously recognized groups of functionally related genes. In many cases, these genes were similarly expressed in only a subset of the experiments, which prevented their association when the data were analyzed with other clustering methods. In general, the flexibility of fuzzy c -means clustering revealed complex correlations between gene-expression patterns and also allowed biologists to advance more elaborated hypotheses of the role and regulation of gene-expression changes.

As mentioned, fuzzy clustering needs an extra fuzziness parameter (m). However, not much has been written in the literature about its choice. A method for the estimation of an upper bound for m and a procedure for choosing it independently of the desired number of clusters has been proposed in [17], which also applied this approach to gene expression microarray data.

The fuzziness parameter m is commonly fixed at a value of 2. However, it has been observed that when applying fuzzy c -means with this value to microarray data, the membership values in the generated partitions are very similar, thus failing to extract any clustering structure. It is known that as m grows, memberships go asymptotically to the reciprocal of the number of clusters (k) [12, p. 73]. In this case it was found that a reasonable estimate for the upper bound of m can be computed from the coefficient of variation (the ratio between the standard deviation and the mean) of the set of object distances. Moreover, a heuristic formula for computing a good value for m is proposed, which ensures high membership values for objects (genes) strongly related to clusters. In this procedure the number of clusters to extract is estimated by using the CLICK algorithm [18], based on graph-theoretic and statistical techniques. This approach was applied to several gene expression data sets: (i) serum data [19], (ii) yeast data [9], and (iii) human cancer data (<http://discover.nci.nih.gov/nature2000/>).

It was found that no single value of the fuzziness parameter m gives good results across the data sets, but rather, that an individual estimate must be used for each of them. Using a clustering criterion based on thresholding the median of the highest membership values of the genes, good results were obtained, which were useful in unraveling complex modes of regulation for some genes. Genes having high memberships to clusters with very different overall expression patterns (as revealed by the values of the second or third highest memberships) might suggest the presence of regulatory pathways. It was shown that the threshold-based selection proposed, preferentially retains genes which are likely to have biological significance in the clusters.

Fuzzy clustering has proved to be particularly effective when used in combination with other techniques. For example, in [20], gene expression profiles are preprocessed by self-organizing maps (SOMs) prior to fuzzy c -means clustering. Then, the prediction of marker genes is performed by visualizing the weighted/mean SOM component plane (manual feature selection) or automatically by a feature selection procedure using pairwise Fisher's linear discriminant analysis. This approach was applied successfully to colon, brain tumor, and cell-line-derived cancer data [21–23]. With this approach the error rates obtained

improved those previously published for the data sets used and in particular, for multiclass problems, they represent approximately a 4% improvement.

Variants of the classical fuzzy clustering scheme have been applied as well, with good results. One example is the so-called fuzzy J -means [24, 25], which is a local search heuristic inspired by a similar procedure developed for crisp clustering. Based on a reformulation of the fuzzy clustering problem in terms of cluster centroids, the idea is to explore all possible centroid-to-pattern relocations and consider the assignment of a single centroid to any unoccupied pattern (a pattern that does not have a centroid coincident with it). Like in standard fuzzy clustering procedures, there is no guarantee that the final solution is a globally optimal one, but this is alleviated by using another heuristic (called variable neighborhood search), to improve further on the solution found. The idea of variable neighborhood search is to systematically explore neighborhoods with a local search algorithm. This algorithm remains near the same locally optimal solution and from it explores increasingly farther regions. New solutions based on random points generated in the neighborhoods are obtained, until one better than the current one is found.

This procedure was applied to simulated breast cancer [26] and human blood data [27], using the method proposed by [17], for the estimation of the fuzziness parameter (m). The study confirmed what has been found in previous applications of fuzzy clustering, namely, that the membership values obtained from the fuzzy methods can be used in different ways. In the first place, the largest membership values can be used to accomplish cluster assignment (allocate each gene into one single cluster, *a la* crisp clustering). In addition, with the membership values it is possible to identify genes most tightly associated to a given cluster and therefore, most likely to be part of only one pathway in all the cases studied.

From the algorithmic point of view, it was found that fuzzy J -means outperformed the standard fuzzy c -means in all data sets studied. From the point of view of computing speed the classical technique is better, but the quality of the results degrades for large data sets and large number of clusters, which is the usual situation in gene expression microarray data.

52.2.2 Rough Sets Methods in Genomics

Rough-set-based classifiers have been applied successfully to a variety of studies using DNA microarray data. In particular, classification using microarray and clinical data in the context of predicting cancer tumor subtypes and clinical parameters from a rough sets perspective is presented in [28]. A data set containing 17 gastric carcinomas was studied with one microarray per tumor and 2504 genes/microarray. (Each probe was printed twice for each array.) The goal was to find genes that allow classification of gastric carcinomas with respect to important clinicopathological parameters (molecular markers) and at the time of the study there were no known molecular markers for the type of tumors considered. Rough-set-based binary classifiers were built for six clinical parameters (Lauren's histopathological classification, localization of the tumor, lymph node metastasis, penetration of the stomach wall, remote metastasis, and serum gastrin).

In a preprocessing stage, feature selection procedures were applied. This is required in most applications of rough sets methods to microarray data because of the very large number of conditional attributes involved (thousands or tens of thousands of genes). Since feature selection and rule construction is often based on reducts and reduct computation is NP-hard [29], heuristics have to be used in order to reduce the cardinality of the set of conditional attributes.

In this case, for a given decision attribute (all binary), the attributes (genes) were selected according to their individual discriminatory power with respect to the two classes involved. A t -statistic was computed in order to evaluate whether the mean values of the ratio values of gene expression intensity for the two classes were significantly different. A bootstrapping procedure was used for estimating the distribution of the standard error of the t -statistic.

Standard rough set methods are not applicable unless discretization is used [30]. In this case the microarray gene expression measurements are continuous attributes. However, recent developments [31] introduce the notion of rough discretization which avoids the difficult problem of discretization and leads to more decision rules, which vote during classification of new observations. This new approach

is particularly oriented to the analysis of gene expression data where genes are used as attributes. In this case the typical situation involves a relatively small number of samples and a large number of attributes (thousands).

In this case, several discretization techniques were applied (frequency binning, naïve, entropy-based, Boolean reasoning, and Bayes-based linear discriminant analysis). The ROSETTA software was used [32].

In [28], three learning algorithms available in the ROSETTA software [32] were applied: genetic reducts [33–35], dynamic reducts [36], and 1R classifier [37]. They achieved classification accuracies between 0.79 and 1.0 (perfect classification) for all of the clinical parameters studied and no strong evidence was found for a given rough classifier to outperform the others. In particular, a comparison with linear and quadratic discriminant analysis resulted favorable to the rough-set-based classifiers from the point of view of performance. Both methods had an area under the ROC curve lower than that of the rough-set-based methods. In particular the performance of quadratic discriminant analysis was poor, with results similar to those of the 1R classifier of ROSETTA. It is conjectured that the underlying assumption of these methods for the data to have a normal distribution might a possible explanation of their poor performance.

Frequency binning, entropy-based, and linear discriminant discretization methods gave good results, as opposed to Boolean reasoning discretization. However, this last technique is known to produce good results in general.

Only a handful of the genes were found to relate to the clinical parameters when consulting the medical literature. For many of the genes, there was no information available at all, or it was not possible to find known associations with the clinical parameters. Therefore, the results obtained by the rough sets analysis are useful in identifying interesting sets of genes deserving further attention.

Rough sets analysis is combined with clustering, within a distributed (grid) computing environment for the analysis of microarray data in [38–40]. Neural networks, genetic programming, and virtual reality visualization techniques [41] are used at a postprocessing stage. The strategy is to create an automated pipelined mining machine as illustrated in Figure 52.3.

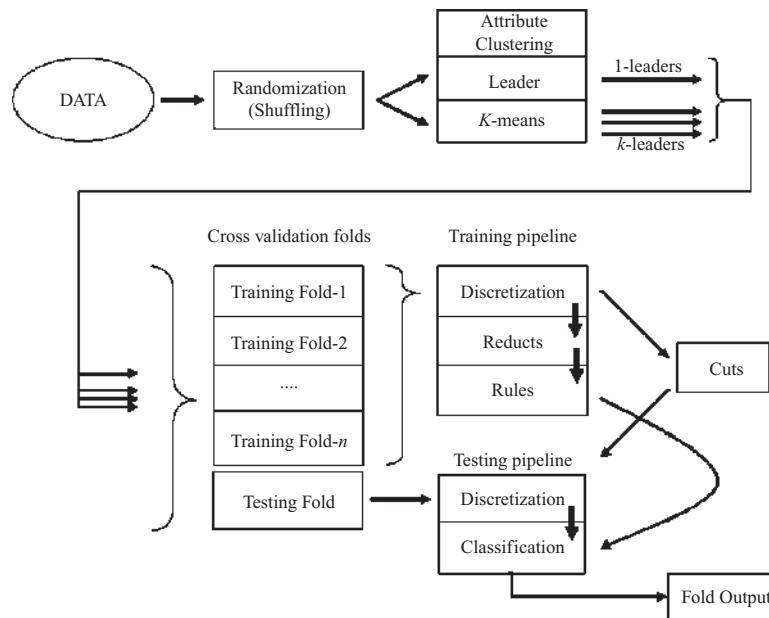


Figure 52.3 Data processing strategy combining clustering with rough sets analysis and cross-validation

In a first step, the objects in the data set are shuffled using a randomized approach in order to reduce possible biases. Then, the attributes of the shuffled data set are clustered using two families of clustering algorithms: the leader (two variants) and k -means (four variants). For a given clustering solution, each of the formed clusters of attributes is represented by exactly one of the original data attributes (the l -leader or k -leader according to the family of clustering algorithm used). For the corresponding clustering scheme, their collection induces a new information system (subset of the original one) amenable to rough sets analysis which proceeds as a n -fold cross-validation process in which for each training fold the following processing is applied: (i) discretization (according to different techniques), (ii) reduct computation, (iii) rule generation. Then the corresponding test fold is (i) discretized using the corresponding cut points found for the training fold and (ii) classified with the set of rules obtained for the training fold. In this way the generalization ability of the generated rules can be evaluated by looking at their min, max, and average performance in the different cross-validation folds.

Cross-validation and bootstrapping are both methods for estimating generalization error based on ‘resampling’ [42–44]. The resulting estimates of generalization error are often used for choosing among various classification or regression models. In k -fold cross-validation, the data is divided into k subsets of approximately equal size. The model is trained k times, each time leaving out one of the subsets from training, but using only the omitted subset to compute whatever error measure is used. If k equals the sample size, this is often called ‘leave-one-out’ cross-validation. A more elaborate and expensive version of cross-validation is called ‘Leave- v -out’ and involves leaving out all possible subsets of v cases.

Each processing stage feeds its results to the next, yielding a pipelined data analysis stream. The whole process is automated using the Condor high-throughput distributed (grid) computing environment [45], (<http://www.cs.wisc.edu/condor/>), with algorithms from the ROSETTA system in batch processing mode embedded [40]. In the first version [38], the RSES system for rough sets processing was used [46].

This approach has been applied to (i) the leukemia gene expression data set reported in [47], consisting of 72 samples from patients with acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML), characterized by 7129 genes [38, 39], and (ii) the breast cancer data set described by [48], which consists of 24 core biopsies taken from patients found to be *resistant* (greater than 25% residual tumor volume) or *sensitive* (less than 25% residual tumor volume) to docetaxel treatment, with 12,625 genes placed onto the microarray [40].

In the leukemia application, two variants of leader clustering with eight different similarity thresholds and four variants of k -means were used (Forgy, Jancey, Convergent, and MacQueen). Rough sets algorithms considered four discretization techniques (Boolean reasoning, entropy, naïve, and seminaïve), with two reduct computation algorithms (Johnson and Holte) on 10 cross-validation folds. In the best experiment, a mean classification accuracy of 0.925 was obtained. Also a set of relevant genes was identified, from which many coincided with those reported [47, 49].

An important research goal is to model the relationships between gene expression as a function of time, the involvement of a gene in a given biological process and the use of the model to predict the biological roles of unknown genes. Rough sets are used [50] to build rule models with minimal features as prediction attributes for gene ontology classes of biological processes. Temporal gene transcript profiles from 24-h fibroblast serum responses data [19] were used in the study. The rule-based classifiers were obtained with the ROSETTA system. Genetic algorithms were used to find approximate reducts (those that preserve only the discriminatory properties for a large fraction of the examples), as they may provide better classification rules and tend to avoid overtraining. Ten-fold cross-validation over the training examples was used to assess the classification quality of the method and 84% of all annotations for the training examples could be classified correctly. A considerable number of the hypothesized new roles for known genes were confirmed by literature search. Moreover, many biological process roles hypothesized for uncharacterized genes were found to agree with assumptions based on additional information.

An important contribution from the point of view of understanding the development of metastatic adenocarcinoma (of unknown origin) and the development of better diagnostic markers is presented in [51]. In that study, expression profiling of 27 candidate markers was done using tissue microarrays and immunohistochemistry. In a first round, 352 primary adenocarcinomas from seven main sites (breast, colon, lung, ovary, pancreas, prostate, and stomach) were considered, including their differential

diagnoses. A combination of rough sets methods (rules found with ROSETTA) and decision trees were used in order to construct a classification scheme. From the original 27 candidate markers, 10 were found important and a classification rate of 88% was obtained using all of the original markers. The same rate was achieved on a test set of 100 primary and 30 metastases tumors using the 10 relevant markers derived from the data analysis process. These results enable better prediction on biopsy material of the primary cancer site in patients with metastatic adenocarcinoma of unknown origin, leading to improved management and therapy.

Another rough-set-based approach for microarray data is presented in [52, 53]. It is illustrated with the leukemia data from [47] with cancer data reported by [54]. The algorithm used is MLEM2, which is part of the LERS data mining system [55–57].

In the first step of processing, the input data is checked for consistency. If the input data is inconsistent, lower and upper approximations of all concepts are computed. Rules induced from the lower approximation of the concept certainly describe the concept and they are called *certain*. Rules induced from the upper approximation of the concept describe the concept only plausibly and they are called *possible*.

The algorithm learns the smallest set of minimal rules describing the concept by exploring the search space of attribute-value pairs. The input data is a lower or upper approximation of a concept, so the algorithm always works with consistent data. The algorithm computes a local covering and then converts it into a rule set. The main underlying concept is that of an attribute-value block, which is the set of objects sharing the same value for a given attribute.

A lower or upper approximation of a concept defined for the decision attribute is said to depend on a set of attribute-value pairs if and only if the intersection of all of its blocks is a subset of the given lower or upper approximation. A set of attribute-value pairs (T) is a *minimal complex* of a lower or upper approximation of a concept defined for the decision attribute (B), if and only if it depends on T and not on any of its proper subsets.

A collection of sets of attribute-value pairs is said to be a *local covering* of B if and only if (i) each member of the collection is a minimal complex of B , and (ii) B can be formed by the union of all of the sets of the collection with minimal cardinality. For a lower or upper approximation of a concept defined for the decision attribute, the LEM2 algorithm produces a single local covering. Its improved version (MLEM2) recognizes integer and real numbers as values of attributes; computing blocks in a different way than for symbolic attributes.

It is interesting that no explicit discretization preprocessing is required due to the way in which blocks are computed for numeric attributes. It combines attribute-value pairs relevant to a concept and creates rules describing the concept. Also it handles missing attribute values during rule induction. Besides the induction of certain rules from incomplete decision tables with missing attribute values interpreted as lost, MLEM2 can induce both certain and possible rules from a decision table with some missing attribute values. They can be of two kinds: ‘lost’ and ‘do not care.’ Another interesting feature of this approach is a mining process based on inducing several rule generations.

The original rule set is the first-generation set. Dominant attributes involved in the first rule generation are excluded from the data set. Then a second rule generation is induced, and so on. The induction of many rule generations is not always feasible, but for microarray data, where the number of attributes (genes) is very large compared to the number of cases, it is. In general, the first rule generation is more valuable than the second rule generation because it is based on a stronger set of condition attributes. Then the second rule generation is more valuable than the third and so on. Rule generations are gradually collected into new rule sets in a process that is repeated until no better sets are obtained in terms of error rates.

When applied to the leukemia data from [47], it was found that the classifiers produced excellent performance. Moreover, many of the genes that were found are relevant to leukemia and coincide with genes found to be relevant in previous studies [15, 38, 47].

The approach was equally successful when applied to the micro-RNA cancer data [54]. All but one case of breast cancer and all cases of ovary cancer were correctly classified using seven attributes (micro-RNAs), from which the functions of four have not yet been determined. For the remaining three with known functions, the connection with certain types of tumors has been clearly established.

52.3 Proteomics

Many researchers consider the forthcoming decades as the *postgenomic era* based on their view that the technical problems for obtaining genomic information have been resolved. However, the understanding of the *proteomes* (all of the proteins in a cell at a given time) poses a big challenge. One main reason is the lack of suitable methods for defining proteomes, which is also related to the increased level of problem complexity. Whilst each of the cells of a given organism has the same DNA, the protein content of a cell depends on the cell type, for which there are many. Moreover, the proteome of a cell changes over time in response to fluctuations in the intra- and extra-cellular environments.

According to the central dogma of biology, a DNA sequence encodes the protein sequence, which determines the three-dimensional (3D) structure of the protein. On the other hand, it is known that protein 3D structure is related with its function. However, proteins are more difficult to analyze than DNA. For proteins there is no chemical process like the polymerase reaction by means of which copies of DNA sequences can be made. Very sensitive and accurate techniques, like mass spectrometry, must be used in order to analyze relatively small numbers of molecules which are produced *in vivo*, in contradistinction with DNA. The information of the DNA (expressed in the four letter language of the nucleotide bases: adenine (A), thymine (T), guanine (G), and cytosine (C)) is converted into a protein which is a sequence of amino acids (20 of them can be used, thus determining a 20-letter alphabet), formed in a way somewhat similar to the nucleotide strand (DNA). Although DNA sequences contain all of the information that is translated into a protein sequence, the converse doesn't hold because in DNA sequences there is information related to the control and regulation of protein expression which can not be extracted from the corresponding protein sequence. Unfortunately, the computational methods available for determining which part of the DNA sequence is translated into a protein sequence and which parts have other possible roles can not provide complete accuracy. Actually, several years after the human genome has been released, there is no reliable estimate of the number of proteins that it encodes. This is a strong reason why known protein sequences should be studied.

Protein strands are much more flexible in space than DNA and form complex 3D structures. The individual amino acids compose a string which makes a protein and are called *residues*. In a process still not understood, the protein folds into a 3D structure. (In fact, sometimes other proteins help a particular protein fold; the so-called *chaperones*.) It is considered that the particularities of this 3D structure determine the functions of the protein. The original chain of residues is called the *primary structure* of the protein. The resulting 3D structure (known as the *tertiary structure* of the protein) is composed of an arrangement of smaller local structures, known as *secondary structures*. They are composed of helices (α -helices, which are right-handed helical folds), strands (β -sheets, which are extended chains with conformations that allow interactions between closely folded parallel segments), and other non-regular regions (Figure 52.4). The *tertiary structure* is the overall 3D structure of the protein, which involves combinations of secondary structure elements in some specific macrostructured ways. Several cases are distinguished: (i) all- α : composed mostly of α -helices, (ii) all- β : composed mostly of β -sheets, (iii) α/β : most regular and common domain structures consist of repeating β - α - β supersecondary units, and (iv) $\alpha+\beta$: there are significant alpha and beta elements mixed, but not exhibiting the regularity found in the α/β type.

Recently, the Human Proteome Initiative has been launched (<http://ca.expasy.org/sprot/hpi/>). So far, *proteomics*, the study of the proteome, has been more difficult than genomics because the amount of information needed is much larger. It is necessary to find what is the molecular function of each protein, what are the biological processes in which a given protein is involved, and where in the cell the protein is located. One specific problem is related to the 3D structure of a protein (structure prediction is one of the most important computational biology problems) and concerted efforts are systematically oriented toward the solution of this problem (<http://predictioncenter.org/casp7/>). Another problem is protein identification, location, and quantification. Individual proteins have a stochastic nature, which needs to be understood in order to assess its effect on metabolic functions.

Proteomics is a rapidly growing field, especially now in the postgenomic era, with methods and approaches which are constantly changing. As with genomics, granular computing is finding its place within the set of computational techniques applied.



Figure 52.4 A visualization of a protein showing structural elements like helices and strands

52.3.1 Fuzzy Methods in Proteomics

Fuzzy sets have been applied to the problem of predicting protein structural classes from amino acid composition. Fuzzy c -means clustering [12] was used in a pioneering work by [78], for classifying globular proteins into the four structural classes (all- α , all- β , α/β , and $\alpha+\beta$), depending on the type, amount, and arrangement of secondary structures present. Each of the structural classes is described by a fuzzy cluster and each protein is characterized by its membership degree to the four clusters and a given protein is classified as belonging to that structural class corresponding to the fuzzy cluster with maximum membership degree. A training set of 64 proteins was studied and the fuzzy c -means algorithm was used for computing the membership degrees. Results obtained for the training set show that the fuzzy clustering approach produced results comparable to or better than those obtained by other methods. A test set of 27 proteins also produced comparable results to those obtained with the training set. This was an unsupervised approach using clustering to estimate the distribution of the training protein data sets. The prediction of the structural class of a given protein was based on a maximal membership function assignment, which is a simple approach.

From a supervised perspective, also using fuzzy methods, the same problem has been investigated in [59], using supervised fuzzy clustering [60]. This is a fuzzy classifier which can be considered as an extension of the quadratic Bayes classifier that utilizes a mixture of models for estimating the class conditional densities. In this case, the overall success rate obtained by the supervised fuzzy c -means (84.4%) improved the one obtained with unsupervised fuzzy clustering by [78]. When applied to another data set of 204 proteins [61], the success rates obtained with jackknifing also improved those obtained with classical fuzzy c -means (73.5% vs. 68.14% and 87.25% vs. 69.12% respectively).

Another direction pursued for predicting the 3D structure of a protein has been the prediction of solvent accessibility and secondary structure as an intermediate step. The reason is that a basic aspect of protein structural organization involves interaction of amino acids with solvent molecules both during the folding process and in the final structure.

The problem of predicting protein solvent accessibility has been approached as a classification task using a wide variety of algorithms like neural networks, Bayesian statistics, SVMs, and others. In particular, a fuzzy k -nearest neighbor technique [62] has been used for this problem [63], which is a simple variant of the classical ‘hard’ k -nearest neighbor classifier where (i) the exponent of the distance between the feature vectors of the query data and its i th nearest reference data is affected by a fuzzy strength

parameter which determines how heavily the distance is weighted when calculating each neighbor's contribution to the membership value, and (ii) the fuzzy membership of the reference vectors to the known classes is used as a weighting factor for the distances.

With this approach, the ASTRAL SCOP data set [64] was investigated. First, leave-one-out cross-validation on 3644 proteins was performed, where one of the 3644 chains was selected for predicting its solvent accessibility. The remaining 3643 chains were used as the reference data set.

Although slight, the fuzzy k -nearest neighbor method exhibited better prediction accuracies than other methods like neural networks and SVMs, which is remarkable, considering the simplicity of the k -nearest neighbor family of classifiers in comparison with the higher degree of complexity of the other techniques.

Clearly, protein identification is a crucial task in proteomics where several techniques like 2D gel electrophoresis, amino acid analysis, and mass spectrometry are used.

Two-dimensional gel electrophoresis is a method for the separation and identification of proteins in a sample by displacement in two dimensions oriented at right angles to one another. This allows the sample to separate over a larger area, increasing the resolution of each component and is a multistep procedure that can separate hundreds to thousands of proteins with high resolution. It works by separating proteins by their *isoelectric point* (which is the pH at which a molecule carries no net electrical charge) in one dimension and by their molecular weight in the second dimension.

Examples of 2D gels from the GelBank database [65] are shown in Figure 52.5, where both the blurry nature of the spots corresponding to protein locations and the deformation effects due to instrumental and other experimental conditions can be observed.

2D gel electrophoresis is generally used as a component of proteomics and is the step used for the isolation of proteins for further characterisation by mass spectroscopy. Another use of this technique is *differential expression*, where the purpose is to compare two or more samples to find differences in their protein expression. For example, in a study looking at drug resistance, a resistant organism is compared to a susceptible one in an attempt to find changes in the proteins expressed in the two samples. Two-dimensional gel electrophoresis is a multistep procedure: (i) the resulting gel is stained for viewing the protein spots, (ii) it is scanned resulting in an image, and (iii) mathematical and computer procedures are applied in order to perform comparison and analysis of replicates of gels. The objective is to determine statistically and biologically meaningful spots.

The uncertainty of protein location in 2D gels, the blurry character of the spots, and the low reproducibility of this technique make the use of fuzzy methods very appealing. A fuzzy characterization of spots in 2D gels is described in [66]. In this approach the theoretical crisp protein location (a point) is replaced by a spot characterization via a two dimensional Gaussian distribution function with independent variances along the two axis. Then, the entire 2D gel is modeled as the sum of the set of Gaussian functions contained and evaluated for the individual cells in which the 2D gel image was digitized. These

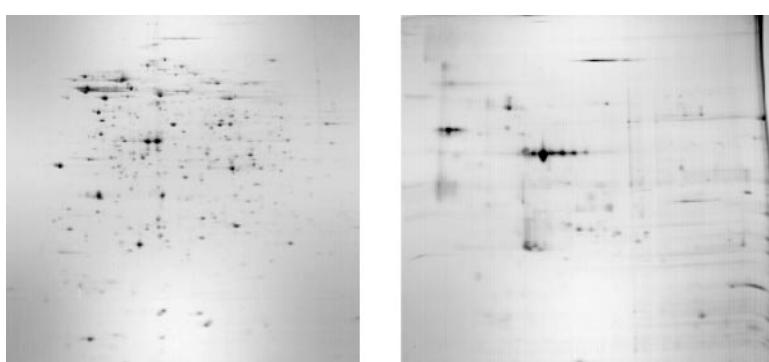


Figure 52.5 GelBank images of 2D gels (<http://gelbank.anl.gov>). The horizontal axis is the isoelectric point and the vertical axis is the molecular weight. Left: S-oneidensis (aerobic growth). Right: P-furiosus (cells grown in the absence of sulfur). Observe the local deformations of the right-hand side image

fuzzy matrices are used as the first step in a processing procedure for comparing 2D gels based on the computation of a similarity index between different matrices. This similarity is defined as a ratio between two overall sums over all of the cells of the two 2D gels compared: the one corresponding to the pairwise minimum fuzzy matrix elements and that of the pairwise maximum fuzzy matrix values [67]. Then, multiple 2D gels are compared by analyzing their similarity matrices by a suite of multivariate methods like clustering, MDS, and others. The application of the method to a complex data set constituted by several 2D maps of sera from rats treated with nicotine (ill) and controls has shown that this method allows discrimination between the two classes.

Another crucial problem associated with 2D gel electrophoresis is the automated comparison of two or more gel images simultaneously. There are many methods for the analysis of 2D gel images but most of the available techniques require intensive user interactions, which creates a major bottleneck and prevents the high-throughput capabilities required to study protein expression in healthy and diseased tissues, where many samples ought to be compared.

An automatic procedure for comparison of the 2D gel images based on fuzzy methods, in combination with global or local geometric transform and brightness interpolation on the images was developed in [68, 69]. The method uses an iterative algorithm, alternating between correspondence and spatial global mapping. The features (spots) are described by Gaussian functions with σ as a fuzziness parameter and the correspondence between two images is represented by a matrix with the rows and columns summing to unit, where its cells measure the matching between the i th spot on image A with the j th spot on image B. These elements are then used as weights in the feature transform. In the process, a starting fuzziness parameter σ is chosen, which is decreased progressively until convergence in the correspondence matrix is obtained. Fuzzy matching is performed for spot coordinates, area and intensity at the maximum; i.e., each spot is described by four parameters; however, spot coordinates are considered as two times more important than the area and intensity. The spatial mapping is performed by bilinear transforms of one image onto the other composed of the inverse and forward transforms. When characterizing the overall geometric distortion between the images one single mapping function can be considered (global transform). However, to deal with local distortions of 2D gel images, piecewise transformations can be used, in this case based on Delaunay triangulation for tessellating the images with linear or cubic interpolation within the resulting triangles. Image brightness is also interpolated and pseudocolor techniques are used for the visualization of matched images.

This method of gel image matching allows efficient automated matching of 2D gel electrophoresis images. Its efficiency is limited by the performance of the fuzzy alignment used to align the sets of the extracted spots. Good results are also obtained with locally distorted 2D gels and the best results are obtained for linear interpolation of the grid and for cubic interpolation of the brightness.

Mass spectrometry is a powerful analytical technique that measures the mass-to-charge ratio (m/z) of ions that is used to identify unknown compounds, to quantify known compounds, and to elucidate the structure and chemical properties of molecules, in particular, proteins (Figure 52.6). Two of the most

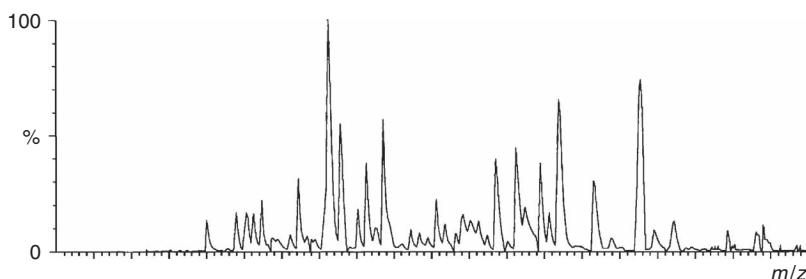


Figure 52.6 Mass spectrum from a sample of mouse brain tissue. The horizontal axis is the mass/charge ratio and the vertical axis is the relative intensity. The individual peaks correspond to different peptides present in the sample, according to their mass/charge ratio

commonly used methods for quantitative proteomics are (i) 2D electrophoresis coupled to either mass spectrometry (MS) or tandem mass spectrometry (MS/MS) and (ii) liquid chromatography coupled to mass spectrometry (LCMS).

With the advances in scientific instrumentation, modern mass spectrometers are capable of delivering mass spectra of many samples very quickly. As a consequence of this high rate of data acquisition, the rate at which protein databases is growing is also high. Therefore the development of high-throughput methods for the identification of peptide fragmentation spectra is becoming increasingly important. But typical analyses of experimental data sets obtained by mass spectrometry on a single processor takes on the order of half a day of computation time (e.g., 30,000 scans against the *Escherichia coli* database). In addition, the search hits are meaningful only when ranked by a relatively computationally intensive statistical significance/relevance score. If modified copies of each mass spectrum are added to the database in order to account for small peak shifts intrinsic to mass spectra owing to measurement and calibration error of the mass spectrometer, combinatorial explosion occurs because of the need of considering the more than 200 known protein modifications.

A ‘coarse filtering-fine ranking’ scheme for protein identification using fuzzy techniques as a fundamental component of the procedure has been introduced recently [70]. It consists of a coarse filter, which is a fast computation scheme that produces a candidate set with many false positives, without eliminating any true positives. The computation is often a lower bound with respect to more accurate matching functions, and it is less computationally intensive. The coarse filtering stage improves on the shared peaks count, followed by a fine filtering stage in which the candidate spectra output by the coarse filter are ranked by a Bayesian scoring scheme.

Mass spectra are represented as high-dimensional vectors of mass/charge values; for convenience, transformed into Boolean vectors. For typical mass ranges, these vectors are $\sim 50,000$ dimensional. Therefore, the similarity measure used is a determining factor of the computational expense of the search. Typically, distance measures for comparison of mass spectra are used and since the specific locations of mass spectra peaks have an associated uncertainty, fuzzy measures are very appropriate. Given two Boolean vectors and a peak mass tolerance (a fuzziness parameter) measured in terms of the mass resolution of the spectra analyzed, a tally measure between two individual mass spectrometry intensities for a given mass/charge ratio is defined. According to this measure, two peaks count as equal (a match) if they lie within a range of vector elements of each other, as determined by the peak mass tolerance. Then a *fuzzy cosine similarity* measure is defined as the ratio between the overall sum of the pairwise match measures and the product of the modules of the two Boolean vectors representing the spectra. This similarity is transformed into a dissimilarity by taking its inverse cosine function, called the *fuzzy cosine distance*, which may fail to fulfill the identity and the triangular inequality axioms of a distance in a metric space.

The precursor mass is the mass of the parent peptide (protein subchains). Another dissimilarity called the *precursor mass distance* is defined as the difference in the precursor masses of two peptide sequences, semithresholded by a precursor mass tolerance factor, which acts as another fuzzification parameter. The idea is that if the absolute precursor mass difference is smaller than the tolerance factor, the precursor mass distance is defined as zero. Otherwise it is set to the absolute precursor mass difference.

This measure is also a semimetric, and the linear combination of the fuzzy cosine distance with the precursor mass distance is the so-called *tandem cosine distance*, carrying the idea of fuzziness in the comparison of the two mass spectra.

This is the measure used by the coarse filter function when querying the mass spectra database. With this ‘coarse filtering-fine ranking’ metric space indexing approach for protein mass spectra database searches, fast, lossless metric space indexing of high-dimensional mass spectra vectors is achieved. The fuzzy coarse filter speeds up searches by reducing both the number of distance computations in the index search and the number of candidate spectra input to a fine filtering stage. Moreover, the measures represent biologically meaningful and computationally efficient distance measures. In fact the number of distance computations is less than 0.5% of the database and the number of candidates for fine filtering to approximately 0.02% of the database.

52.3.2 Rough Sets Methods in Proteomics

The prediction of the protein structure class (all- α , all- β , α/β , and $\alpha+\beta$) is one of the most important problems in modern proteomics and it has been approached using a wide variety of techniques like discriminant analysis, neural networks, Bayes decision rules, SVMs, boost of weak classifiers, and others. Recently, rough sets have been applied as well [71]. In the study, two data sets of protein domain sequences from the SCOP database were used: one consisting of 277 sequences, and another with 498 sequences. In both cases, the condition attribute set was assembled with compositional percentages of the 20 amino acids in primary sequences and 8 physicochemical properties, for a total of 28 attributes. The decision attribute was the protein structure class consisting of the four previously mentioned categories. The ROSETTA system was used for rough sets processing with seminäive discretization and genetic algorithms for reduct computation. Self-consistency and jackknife tests were applied and the rough sets results were compared with other classifiers like neural networks and SVMs. From this point of view, the performance of the rough set approach was on the average equivalent to that of SVM and superior to that of neural networks. For example, for the α/β class, the results obtained with rough sets were the overall best with respect to the other algorithms (93.8% for the first data set composed of 277 sequences and 97.1% for the second composed of 498). It was also proved that amino acid composition and physicochemical properties can be used to discriminate protein sequences from different structural classes, suggesting that a rough set approach may be extended to the prediction of other protein attributes, such as subcellular location, membrane protein type, and enzyme family classification.

Proteomic biomarker identification is another important problem because in the search for early diagnosis in diseases like cancer, it is essential to determine molecular parameters (so-called biomarkers) associated with the presence and severity of specific disease states. Rough sets have been applied to this problem [72] for feature selection in combination with blind source separation [73] in a study oriented to the identification of proteomic biomarkers of ovarian and prostate cancer. The information used was serum protein profiles as obtained by mass spectrometry in a data set composed of 19 protein profiles belonging to two classes: myeloma (a form of cancer) and normal. Each profile was initially described by 30,000 values of mass-to-charge ratio (the attributes), as obtained from the mass spectrometer. Then, they were reduced to a subsequence of 100 by choosing those with the highest Fisher discriminant power. Blind source separation separated the subsequence into five source signals, further reduced to only two when reducts were computed. In order to verify the effect of the use of a reduced set of attributes in the classification, a neural network consisting of a single neuron was used. Average testing errors revealed that there was a generalization improvement with the use of a smaller number of selected attributes. Despite being in its early stages and hindered by the problem of determining the optimal number of sources to extract, this approach showed the advantages of combining rough sets with signal processing techniques.

Drug design is another important problem and the development of the so-called G-protein-coupled receptors (GPCRs) are among the most important targets. Their 3D structure is very difficult to find experimentally. Hence, computational methods for drug design have relied primarily on techniques such as 2D substructure similarity searching and quantitative structure activity relationship modeling [74]. Very recently this problem has been approached from a rough sets perspective [75].

A ligand is a molecule that interacts with a protein, by specifically binding to the protein via a non-covalent bond, while a receptor is a protein that binds to the ligand. Protein–ligand binding has an important role in the function of living organisms and is one method that the cell uses to interact with a wide variety of molecules. The modeling of the receptor–ligand interaction space is made using descriptors of both receptors and ligands. These descriptors are combined and associated with experimentally measured binding affinity data. From them, associations between receptor–ligand properties can be derived. In all of the three data sets investigated the condition attributes were descriptors of receptors and ligands and the decision attribute was a two category class of binding affinity values (low and high). The goal was to induce models separating high and low binding receptor–ligand complexes formulated as a set of decision rules obtained using the ROSETTA system. Three data sets were studied, and each was randomly divided into a training set of 80% (with 32, 48, and 105 objects respectively) and an external test set composed of 20% of the objects (with 8, 12, and 26 objects respectively). The number of condition attributes for the three data sets was 6, 8, and 55 respectively.

Object related reducts were computed using Johnson's algorithm [76] and rules were constructed from them. They were used for validation and interpretation of the induced models. Approximate reducts were computed with genetic algorithms for an implicit ranking of attributes. Mean accuracy and area under the ROC curve (receiver operating characteristic) served as measures of the discriminatory power of the classifiers evaluated by cross-validation. The rough set models provided good accuracies in the training set, with mean 10-fold cross-validation accuracy values in the 0.81–0.87 range for the three data sets and in the 0.88–0.92 range for the independent test set. These results complement those obtained for the same data sets using the partial least squares technique [77] for the analysis of ligand–receptor interactions. Besides quality and robustness, rough sets models have advantages like their minimality with respect to the number of attributes involved and their interpretability. All of them are very important because they provide a deeper understanding of ligand–receptor interactions.

Rough sets models have been proved to be successful and robust in, for example, fold recognition, prediction of gene function from time series expression profiles, and the discovery of combinatorial elements in gene regulation. From the point of view of rough sets software tools used in bioinformatics, ROSETTA [32] is the one which has been mostly used, followed by RSES [46] and LERS [55, 56]. It is important to observe that the effectiveness of rough set approaches increases when used in combination with other computational intelligence techniques like neural networks, evolutionary computation, support vector machines, statistical methods, etc.

52.4 Conclusion

All of these examples indicate that granular computing methods have a large potential in bioinformatics. Their capabilities for uncertainty handling, feature selection, unsupervised and supervised classification, and their robustness, among others, make them very powerful tools, useful for the problems of interest to both classical and modern bioinformatics. So far, fuzzy and rough sets methods have been the preferred granular computing techniques used in bioinformatics and they have been applied either alone or in combination with other mathematical procedures. Most likely this is the best strategy. The number of applications in this domain is growing rapidly and this trend should continue in the future.

References

- [1] P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach*. MIT Press, Cambridge, MA, 1999.
- [2] A.M. Campbell and L.J. Heyer. *Discovering Genomics, Proteomics and Bioinformatics*. CSHL Press, Pearson Education Inc., New York, 2003.
- [3] A.D. Baxevanis and B.F. Ouellette. *Bioinformatics. A Practical Guide to the Analysis of Genes and Proteins*. John Wiley, Hoboken, NJ, 2005.
- [4] M. Shena, D. Shalon, R. Davis, and P. Brown. Quantitative monitoring of gene expression patterns with a complementary microarray. *Science* 270(5235) (1995) 467–470.
- [5] R. Ekins and F.W. Chu. Microarrays: Their origins and applications. *Trends Biotechnol.* 17 (1999) 217–218.
- [6] P. Gwynne and G. Page. Microarray analysis: the next revolution in molecular biology. *Science* 285 (1999) 911–938.
- [7] D.J. Lockhart and E.A. Winzeler. Genomics, gene expression and DNA arrays. *Nature* 405(6788) (2000) 827–836.
- [8] P.J. Woolf and Y. Wang. A fuzzy logic approach to analyzing gene expression data. *Physiol. Genomics* 3 (2000) 9–15.
- [9] R.J. Cho, M.J. Campbell, E.A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T.G. Wolfsberg, A.E. Gabrielian, D. Landsman, D.J. Lockhart, and R.W. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell* 2(1) (1998) 65–73.
- [10] E.H. Ruspini. A new approach to clustering. *Inf. Control* 15(1) (1969) 22–32.
- [11] J.C. Bezdek. *Fuzzy Mathematics in Pattern Classification*. Cornell University, Ithaca, NY, 1973.
- [12] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function*. Plenum Press, NY, 1981.
- [13] J.C. Bezdek and S.K. Pal. *Fuzzy Models for Pattern Recognition Method that Search for Structures in Data*. IEEE Press, New York, 1992.

- [14] I. Gath and A.B. Geva. Unsupervised optimal fuzzy clustering. *Trans. Pattern Anal. Mach. Intell.* 11 (1989) 773–781.
- [15] E.E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In: *Proceedings of the IEEE CDC*, San Diego, CA, 1979, pp. 761–766.
- [16] A.P. Gasch and M.B. Eisen. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biol.* 3(11) (2002) 1–22.
- [17] D. Dembele and P. Kastner. Fuzzy C-means method for clustering microarray data. *Bioinformatics* 19(8) (2003) 973–980.
- [18] R. Sharan and R. Shamir. CLICK: A Clustering algorithm with application to gene expression analysis. In: *Proceedings of 8th International Conference on Intelligent Systems for Molecular Biology (AAAI-ISMB)*, UC San Diego, La Jolla, CA, August 19–23, 2000. AAAI Press, Menlo Park, CA, 2000, pp. 307–316.
- [19] V.R. Iyer, M.B. Eisen, D.T. Ross, G. Schuler, T. Moore, J.C. Lee, J.M. Trent, L.M. Staudt, J. Hudson, and M.S. Boguski. The transcriptional program in the response of human fibroblasts to serum. *Science* 283 (1999) 83–87.
- [20] J. Wang, T.H. Bø, I. Jonassen, O. Myklebost, and E. Hovig. Tumor classification and marker gene prediction by feature selection and fuzzy c-means clustering using microarray data. *BMC Bioinform.* 4(60) (2003) pp. 1471–2105.
- [21] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. U.S.A.* 96(67) (1999) 45–6750.
- [22] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, M.E. McLaughlin, J.Y.H. Kim, L.C. Goumnerova, P.M. Black, C. Lau, J.C. Allen, D. Zagzag, J.M. Olson, T. Curran, C. Wetmore, J.A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D.N. Louis, J.P. Mesirov, E.S. Lander, and T.R. Golub. Prediction of central nervous system embryonal tumor outcome based on gene expression. *Nature* 415 (2002) 436–442.
- [23] T.D. Ross, U. Scherf, M.B. Eisen, C.M. Perou, C. Rees, P. Spellman, V. Iyer, S.S. Jeffrey, M.V.D. Rijn, M. Waltham, A. Pergamenschikov, J.C.F. Lee, D. Lashkari, D. Shalon, T.G. Myers, J.N. Weinstein, D. Bostein, and P.O. Brown. Systematic variation in gene expression patterns in human cancer cell lines. *Nat. Genet.* 24 (2000) 227–235.
- [24] N. Belacel, P. Hansen, and N. Mladenovic. Fuzzy J-means: A new heuristic for fuzzy clustering. *Pattern Recognit.* 35 (2002) 2193–2200.
- [25] N. Belacel, M. Čuperlović-Cufic, M. Laflamme, and R. Ouellette. Fuzzy J-means and VNS methods for clustering genes from microarray data. *Bioinformatics* 20 (2004) 1690–1701.
- [26] T. Sorlie, C.M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M.B. Eisen, M. van de Rijn, and S.S. Jeffrey. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc. Natl. Acad. Sci. U.S.A.* 98 (2001) 10869–10874.
- [27] A.R. Whitney, M. Diehn, S.J. Popper, A.A. Alizadeh, J.C. Boldrick, D.A. Relman, and P.O. Brown. Individuality and variation in gene expression patterns in human blood. *Proc. Natl. Acad. Sci. U.S.A.* 100 (2003) 1896–1901.
- [28] H. Midelfart, J. Komorowski, K. Nørsett, F. Yadetie, A. Sandovik, and A. Lægreid. Learning rough set classifiers from gene expressions and clinical data. *Fundam. Inf.* 53(2) (2002) 155–183.
- [29] J. Wróblewski. Ensembles of classifiers based on approximate reducts. *Fundam. Inf.* 47 (2001) 351–360.
- [30] J. Bazan, H.S. Nguyen, S.N. Nguyen, P. Synak, and J. Wróblewski. Rough set algorithms in classification problem. In: *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems*. Physica-Verlag, Heidelberg, 2000, pp. 49–88.
- [31] D. Slezak and J. Wróblewski. Rough discretization of gene expression data. In: *Proceedings of 2006 International Conference on Hybrid Information Technology*, Cheju Island, Korea, November 9–11, 2006.
- [32] A. Øhrn, J. Komorowski, and A. Skowron. The design and implementation of a knowledge discovery toolkit based on rough sets: The ROSETTA system. In: *Rough Sets in Knowledge Discovery 1: Methodology and Applications*, Vol. 18 of *Studies in Fuzziness and Soft Computing*, Physica-Verlag, Germany, 1998, pp. 376–399.
- [33] J. Wróblewski. Finding minimal reducts using genetic algorithms. In: *Proceedings of Second International Conference on Information Sciences*, Wrightsville Beach, NC, 1995, September 28–October 1, pp. 186–189.
- [34] J. Wróblewski. Genetic algorithms in decomposition and classification problems. In: L. Polkowski and A. Skowron (eds.), *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*, Vol. 19 of *Studies in Fuzziness and Soft Computing*, Physica-Verlag, Germany, 1998, pp. 471–487.
- [35] S. Vinterbo and A. Øhrn. Minimal approximate hitting sets and rule templates. *Int. J. Approx. Reason.* 25(2) (2000) 123–143.
- [36] J.G. Bazan. Dynamic reducts and statistical inference. In: *Proceedings of Sixth International Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96)*, July 1–5, 1996, Granada, Spain, Vol. 3, 1996.

- [37] R.C. Holte. Very simple classification rules perform well on most commonly used data sets. *Mach. Learn.* 11(1) (1993) 63–91.
- [38] J.J. Valdés and A.J. Barton. Gene discovery in leukemia revisited: A computational intelligence perspective. In: *The Seventeenth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA/AIE 2004)*, Ottawa, Ontario, Canada, May 17–20, 2004.
- [39] J.J. Valdés and A.J. Barton. Relevant attribute discovery in high dimensional data based on rough sets applications to leukemia gene expressions. In: *Tenth International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC 2005)*, Regina, Saskatchewan, Canada, August 31–September 3, 2005. *Lecture Notes in Computer Sciences/Lecture Notes in Artificial Intelligence. LNAI 3642*, 2005, pp. 362–371.
- [40] J.J. Valdés and A.J. Barton. Relevant attribute discovery in high dimensional data: Application to breast cancer gene expressions. In: *First International Conference on Rough Sets and Knowledge Technology (RSKT 2006)*, Chongqing, P.R. China, July 24–26, 2006. *Lecture Notes in Computer Sciences/Lecture Notes in Artificial Intelligence. LNAI 4062*, 2006, pp. 482–489.
- [41] J.J. Valdés. Virtual reality representation of information systems and decision rules: An exploratory technique for understanding data knowledge structure. In: *The 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'2003)*, Chongqing, China. *Lecture Notes in Artificial Intelligence LNAI 2639*, Springer-Verlag, Heidelberg, May 26–29, 2003, pp. 615–618.
- [42] S.M. Weiss and C.A. Kulikowski. *Computer Systems That Learn*. Morgan Kaufmann, San Matco, CA, 1991.
- [43] B. Efron and R.J. Tibshirani. Improvements on cross-validation: The .632+ bootstrap method. *J. Am. Stat. Assoc.* 92 (1997) 548–560.
- [44] J.S.U. Hjorth. *Computer Intensive Statistical Methods Validation, Model Selection, and Bootstrap*. Chapman & Hall, London, 1994.
- [45] D. Thain, T. Tannenbaum, and M. Livny. Distributed computing in practice: The condor experience. *Concurrency and Computation: Practice and Experience*. 17 (2–4) (2005) 323–356.
- [46] J.G. Bazan, S. Szczuka, and J. Wróblewski. A new version of rough set exploration system. In: *Third International Conference on Rough Sets Current Trends in Computing RSCTC 2002*, Malvern, PA, USA, October 14–17, 2002. *Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence Series) LNCS 2475*. Springer-Verlag, Heidelberg, 2002, pp. 397–404.
- [47] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286 (1999) 531–537.
- [48] J.C. Chang, E.C. Wooten, A. Tsimelzon, S.G. Hilsenbeck, M.C. Gutierrez, R. Elledge, S. Mohsin, C.K. Osborne, G.C. Chamness, D.C. Allred, and P. O'Connell. Gene expression profiling for the prediction of therapeutic response to docetaxel in patients with breast cancer: Mechanisms of disease. *Lancet* 362(9381) (2003) 362–369.
- [49] F. Famili and J. Ouyang. Data mining: Understanding data and disease modeling. In: *Proceedings of 21st IASTED International Conference of Applied Informatics*, Innsbruck, Austria, February 2003, pp. 32–37.
- [50] A. Lægreid, T.R. Hvidsten, H. Midelfart, J. Komorowski, and A. K. Sandvik. Predicting gene ontology biological process from temporal gene expression patterns. *Genome Res.* 13 (2003) 965–979.
- [51] J.L. Dennis, T.R. Hvidsten, E.C. Wit, J. Komorowski, A.K. Bell, I. Downie, J. Mooney, C. Verbeke, C. Bellamy, W.N. Keith, and K.A. Oien. Markers of adenocarcinoma characteristic of the site of origin: Development of a diagnostic algorithm. *Clin. Cancer Res.* 11(10) (2005) 3766–3772.
- [52] J. Fang and J.W. Grzymala-Busse. Mining of microRNA expression data-A rough set approach. In: *First International Conference on Rough Sets and Knowledge Technology (RSKT 2006)*, Chongqing, P.R. China. July 24–26, 2006. *Lecture Notes in Computer Sciences/Lecture Notes in Artificial Intelligence. LNAI 4062*, 2006, pp. 758–765.
- [53] J. Fang and J.W. Grzymala-Busse. Leukemia prediction from gene expression data|A rough set approach. In: *Proceedings of ICAISC'2006, the Eighth International Conference on Artificial Intelligence and Soft Computing*, Zakopane, Poland, June 25–29, 2006. *Lecture Notes in Artificial Intelligence, 4029*. Springer-Verlag, Heidelberg, 2006.
- [54] J. Lu, G. Getz, E.A. Miska, E. Alvarez-Saavedra, J. Lamb, D. Peck, A. Sweet-Cordero, B.L. Ebet, R.H. Mak, A.A. Ferrando, J.R. Downing, T. Jacks, H.R. Horvitz, and T.R. Golub. MicroRNA expression profiles classify human cancers. *Nature* 435 (2005) 834–838.
- [55] J.W. Grzymala-Busse. LERS: A system for learning from examples based on rough sets. In: R. Slowinski (ed.), *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory*. Kluwer Academic Publishers, Dordrecht, 1992, pp. 3–18.
- [56] J.W. Grzymala-Busse. A new version of the rule induction system LERS. *Fundam. Inf.* 31 (1997) 27–39.
- [57] J.W. Grzymala-Busse. MLEM2: A new algorithm for rule induction from imperfect data. In: *Proceedings of 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002*, Annecy, France, July 1–5, 2002, pp. 243–250.

- [58] C.T. Zhang, K.C. Chou, and G.M. Maggiora. Predicting protein structural classes from amino acid composition: Application of fuzzy clustering. *Protein Eng.* 8(5) (1995) 425–435.
- [59] H.B. Shen, J. Yang, X.J. Liu, and K.C. Chou. Using supervised fuzzy clustering to predict protein structural classes. *Biochem. Biophys. Res. Commun.* 334 (2005) 577–581.
- [60] J. Abonyi and F. Szeifert. Supervised fuzzy clustering for the identification of fuzzy classifiers. *Pattern Recognit. Lett.* 24(14) (2003) 2195–2207.
- [61] K.C. Chou. A key driving force in determination of protein structural class. *Biochem. Biophys. Res. Commun.* 264 (1999) 216–224.
- [62] J.C. Bezdek, L.O. Hall, and L.P. Clark. Review of MR image segmentation techniques using pattern recognition. *Med. Phys.* 20 (1993) 1033–1048.
- [63] J. Sim, S.Y. Kim, and J. Lee. Prediction of protein solvent accessibility using fuzzy k -nearest neighbor method. *Bioinformatics* 21(12) (2005) 2844–2849.
- [64] ASTRAL SCOP: The ASTRAL Compendium for Sequence and Structure Analysis. <http://astral.berkeley.edu>, accessed 2007.
- [65] GelBank database. Argonne National Lab. Protein Mapping Group. at <http://gelbank.anl.gov>, accessed 2007.
- [66] E. Marengo, E. Robotti, V. Gianotti, and P.G. Righetti. A new approach to the statistical treatment of 2D-maps in proteomics using fuzzy logics. *Annali di Chim.* 93 (2003) 105–115.
- [67] E. Marengo, E. Robotti, V. Gianotti, P.G. Righetti, D. Cecconi, and E. Domenici. A new integrated statistical approach to the diagnostic use of two-dimensional maps. *Electrophoresis* 24 (2003) 225–236.
- [68] K. Kaczmarek, B. Walczak, S. de Jong, and B.G.M. Vandeginste. Feature based fuzzy matching of 2D gel electrophoresis images. *J. Chem. Inf. Comput. Sci.* 42 (2002) 1431–1442.
- [69] K. Kaczmarek, B. Walczak, S. de Jong, and B.G.M. Vandeginste. Matching 2D gel electrophoresis images, *J. Chem. Inf. Comput. Sci.* 43 (2003) 978–986.
- [70] S.R. Ramakrishnan, R. Mao, A.A. Nakorchevskiy, J.T. Prince, W.S. Willard, W. Xu, E.M. Marcotte, and D.P. Miranker. A fast coarse filtering method for peptide identification by mass spectrometry. *Bioinformatics* 22(12) (2006) 1524–1531.
- [71] Y. Cao, S. Liu, L. Zhang, J. Qin, J. Wang, and K. Tang. Prediction of protein structural class with rough sets. *BMC Bioinform.* 7 (2006) 20.
- [72] G.M. Boratyn, T.G. Smolinski, J.M. Zurada, M. Mariofanna Milanova, S. Bhattacharyya, and L.J. Suva. Hybridization of blind source separation and rough sets for proteomic biomarker identification. In: *Proceedings of Seventh International Conference. Artificial Intelligence and Soft Computing (ICAISC 2004), Zakopane, Poland, June 7–11, 2004, (Lecture Notes in Artificial Intelligence Series) LNAI 3070*, 2004, pp. 486–491.
- [73] J.F. Cardoso. Blind signal separation: Statistical principles. *Proc. IEEE* 9(10) (1998) 2009–2025.
- [74] J. Bajorath. Integration of virtual and high-throughput screening. *Natl. Rev. Drug Discovery* 1 (2002) 882–894.
- [75] H. Strömbärgsson, P. Prusis, H. Midelfart, M. Lapinsh, J. Wikberg, and J. Komorowski. Rough set-based proteochemometrics modeling of Gprotein-coupled receptor-ligand interactions. *Proteins Struct. Funct. Bioinform.* 63 (2006) 24–34.
- [76] D.S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.* 9 (1974) 256–278.
- [77] P. Prusis, R. Mucaniece, P. Andersson, C. Post, T. Lundstedt, and J. Wikberg. PLS modeling of chimeric MS04/MSH-peptide and MC1/MC3-receptor interactions reveals a novel method for the analysis of ligand-receptor interactions. *Biochim. Biophys. Acta* 1544(1–2) (2001) 350–357.

Index

A

- Actor-critic method 680
- Adaptive judgment 335
- Aggregation operator 240
- Agent 425, 432, 460
- Alpha-cut (α -cut) 104, 250, 526, 608
- Analogy-based reasoning 1037
- Approximate methods 50
- Approximate reasoning 472, 508, 801
- Approximate reasoning network 479
- Approximation space 433, 453, 482, 534, 671, 676, 1063
- Approximation 293, 425
- Association analysis 893
- Attribute 299
- Automated planning 789, 793

B

- Behavioral graph 782
- Boolean reasoning 301
- BZMV algebra 614

C

- Case-based reasoning (CBR) 367, 1005
- CADNA 44
- Cartesian product 229
- CESTAC 35
- Chromosome 664
- Clustering 1100
- Clustering and taxonomy 154, 191
- Cluster validity 161
- Compositional rule of inference 227
- Complete lattice 736
- Computing with Words 286, 510, 567
- Cognition 632
- Concept approximation 436

C

- Conjunction 206
- Conflict theory 303, 1056
- Consensus 921
- Constraint propagation 83
- Control 581, 582

D

- Data error 44
- Data mining 889
- Decision making 1070
- Decision system 298
- Decomposition 465
- Defect correction 69
- Degree of compatibility 504
- Descriptive model 848
- Diagnostic classification 848
- Dichotomy 97
- Direct measurement 9
- Discernibility 301
- Discretization 895
- Differential evolution 275
- Discrete stochastic arithmetic (DSA) 42
- Disjunction 207
- Distance 154, 505, 1044
- Document clustering 842, 988, 992
- Document indexing 838, 840
- Dominance 348, 360
- Dominance-based rough sets 351
- Duality in fuzzy linear programming 703
- Dynamic system 778

E

- ECG classification 849
- Eigen fuzzy set 863
- Enclosure 58, 66
- Entropy 15, 199, 505

- Epsilon inflation 69
 Error propagation 35
 Ethogram 672, 677
 Extension principles 57
 Interval extension principle 60
 Fuzzy extension principle 73, 253
 Evolutionary computing 768
 Expert uncertainty 11
- F**
 Feasible solution 697
 Feature reduction 1007
 Feature selection 896
 Floating point (FP) error 34
 Food web 91
 Fuzzification 560, 579
 Fuzzy sets 71, 98, 537, 690
 Normal 102
 Support 103
 Core 103
 Cardinality 105
 Fuzzy
 Arithmetic 74, 262
 Associative memory 733, 734, 741, 746
 Clustering 121, 157, 176, 183, 192, 899
 Cognitive map 130, 755, 756
 Control 226, 931
 Decision tree 128
 Decoding (defuzzification) 171, 173, 176, 246
 Dilation 646
 Encoding (fuzzification) 171, 173
 Integration and differentiation 276
 Interval 72
 Linear programming 690, 695
 Majority 916
 Number 253, 254, 259
 Regression 719, 723
 Relational equations 227
 Preferences 912, 915
 Relation 690, 863
 System 557, 576
 Tournament 914
 Fuzzy neural networks 129
 Fuzzy rough set 538
- G**
 Genetic operators 664
 Genetic optimization 659, 664
 Genomics 1094, 1097, 1099
- Granularity 633
 Granular computing 450, 510
 Granular Intensional logic 387, 388
 Granular system 428
 Syntax 429
 Semantics 429
 Granulation of information 109, 518
 Group decision making 912, 916
- H**
 Hierarchical granulation 806
 Hierarchical reasoning 477, 483
 Hebbian learning 761, 763
 Hierarchy 407
 Hierarchical learning 816
- I**
 Image processing 509, 864
 Implication 208, 214, 537
 Indetermination index 505
 Indirect measurement 9
 Indiscernibility relation 295
 Information granule 330, 334, 449
 Information system 378, 451, 473
 Information map 473
 Information retrieval 836, 843
 Interpolation 233, 240
 Interval analysis 59, 61, 1069, 1078
 Interval arithmetic 58, 81
 Constraint interval arithmetic 63
 Specialized interval arithmetic 65
 Interval computing 13
 Historic overview 19
 Applications 27
 Interval linear programming 710
 Interval Newton method 84
 Interval-valued fuzzy sets 491
 Interval-valued implication 501
 Iterative methods 48
- J**
- K**
 K-Means clustering 996
 Knowledge fusion 391
- L**
 Lattice 206
 Least commitment principle 101
 Level fuzzy set 608
 Linearization 18

Linguistic data 720, 729
Linguistic quantifier 909

M

Machine Learning 889, 894, 898
Mathematical morphology 643, 644
Membership function estimation 112
Mereology 311, 379, 427
Measurement 518
Measurement theory 142
Measurement uncertainty 519, 521
Membership function estimation 141
Missing attribute value 873
Modular approach 661
Morphological neural network 734
Multiagent system 463, 933
Multilevel granularity 403
Multiple-criteria decision 358

N

Negation 206, 492
NP completeness 13
Neural network 421, 658
Neuro-fuzzy system 595

O

Object 287
Ontology 944
Ontology matching 825
Ordered weighted average (OWA) 908, 910
Outlier 823, 831

P

Pairwise comparison 146
Partition 189, 199
Perception 631
Phase equilibrium 85
Polling 143
Possibilistic clustering 159
Prediction model 950, 952
Probability-possibility defuzzification 180
Problem solving 403
Proteomics 1103

Q

Quotient-based model 418
Quotient space 412, 414
Query language 840, 841
Query refinement 534, 545

R

Rough set 291, 293, 347, 377
Rough sets 426
Rough fuzzy 444
Ranking 358
Rule extraction 666
Rough-neuro hybridization 658, 659
Rough set 987, 1006
Relational equation 1076
Reinforcement learning 339, 671
Risk management 1086
Requirements engineering 1059
Rough sets 657, 675
Rough sets 883
Rough set 824
Rough set 534
Rough
Classifier 803
Clustering 970
Inclusion 380
K-Means 73
Mereology 380
Support vector clustering 975
Rough-neural computing 392
Rule induction 892
Rule induction 885
Road traffic 813
RBF neural network 855

S

Satisfiability 452, 456
Scaling 144
Shadowed set 604, 617
Self-organizing map 973
Search engine 835
Stirling number 155
Software 41
Spatial reasoning 629, 630
Spatial relation 638, 640, 647
Spatiotemporal reasoning 471
Stream flow forecasting 957
Stochastic arithmetic 37
Storage capacity 747
Structure assessment 726
Structural pattern recognition 649
Symbolic algebra 1076

T

T-conorm 497
T-norm 213, 497, 693, 738
Temporal pattern 780

Text mining 991
Time series 594, 597
Tolerance relation 431
Transitive closure 547
Transition state analysis 88
Triangular norm 537
Type-2 fuzzy sets 492, 553, 556,
 576
Twiddle rule 674

U
User centricity 99

V
Vagueness 290
Validation 136
Verification 133
Vocabulary matching 722
Voting paradoxes 924

W
Wisdom technology 331, 338

Y
Z