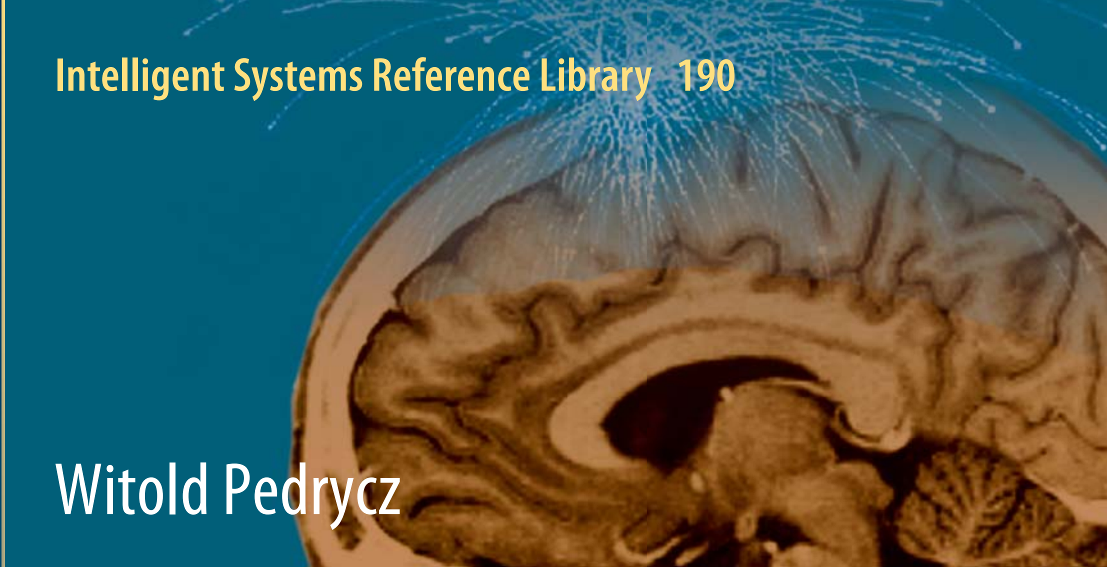
Intelligent Systems Reference Library 190



Witold Pedrycz

An Introduction to Computing with Fuzzy Sets

Analysis, Design, and Applications

Springer

Intelligent Systems Reference Library

Volume 190

The aim of this series is to publish a Reference Library, including novel advances and developments in all aspects of Intelligent Systems in an easily accessible and well structured form. The series includes reference works, handbooks, compendia, textbooks, well-structured monographs, dictionaries, and encyclopedias. It contains well integrated knowledge and current information in the field of Intelligent Systems. The series covers the theory, applications, and design methods of Intelligent Systems. Virtually all disciplines such as engineering, computer science, avionics, business, e-commerce, environment, healthcare, physics and life science are included. The list of topics spans all the areas of modern intelligent systems such as: Ambient intelligence, Computational intelligence, Social intelligence, Computational neuroscience, Artificial life, Virtual society, Cognitive systems, DNA and immunity-based systems, e-Learning and teaching, Human-centred computing and Machine ethics, Intelligent control, Intelligent data analysis, Knowledge-based paradigms, Knowledge management, Intelligent agents, Intelligent decision making, Intelligent network security, Interactive entertainment, Learning paradigms, Recommender systems, Robotics and Mechatronics including human-machine teaming, Self-organizing and adaptive systems, Soft computing including Neural systems, Fuzzy systems, Evolutionary computing and the Fusion of these paradigms, Perception and Vision, Web intelligence and Multimedia.

** Indexing: The books of this series are submitted to ISI Web of Science, SCOPUS, DBLP and Springerlink.

More information about this series at http://www.springer.com/series/8578

Witold Pedrycz

An Introduction to Computing with Fuzzy Sets

Analysis, Design, and Applications

Springer

Witold Pedrycz
Department of Electrical and Computer
Engineering
University of Alberta
Edmonton, AB, Canada

ISSN 1868-4394 ISSN 1868-4408 (electronic)
Intelligent Systems Reference Library
ISBN 978-3-030-52799-0 ISBN 978-3-030-52800-3 (eBook)
https://doi.org/10.1007/978-3-030-52800-3

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Since their inception 55 years ago, fuzzy sets have emerged as a novel conceptual framework to facilitate developments of human-centric systems. Human centricity is one of the pivotal features of intelligent systems. The pursuits in data mining, data analytics, image understanding and interpretation, recommender systems, explainable artificial intelligence (XAI) and inherently associated with human centricity are at the forefront of advanced technologies being under intensive studies. Fuzzy sets have been evolving over the half century fostering new interactions and exploring uncharted territories within the discipline of computational intelligence. With this regard, fuzzy sets are important examples of information granules. Granular computing offers processing principles that opens new frontiers in computing with fuzzy sets and builds an overarching and conceptually appealing processing environment.

The principles and practice of fuzzy sets evolve in response to the challenges of real-world problems. This makes the evident shift in the way how the material becomes exposed here. The leitmotiv of the book is that the fuzzy set technology delivers fundamentals instrumental in supporting analysis and synthesis of intelligent systems promoting their applications to a variety of disciplines of science and engineering. In the organization of the material, we put equal emphasis on analysis and synthesis. We offer convincing motivation behind concepts and constructs of fuzzy sets. Equally importantly, we discuss how introduced constructs are built, evaluated and refined (optimized).

This textbook can serve as a prudently orchestrated body of knowledge, and the material is suitable for covering over a one-term graduate course. The book could also serve as a sound text suitable for senior undergraduate courses. It could be helpful to those interested to gain preliminary yet working knowledge about fuzzy sets. The pedagogy taken in the organization of the material engages a top-down strategy: We start with fundamentals, motivate them and deliver ways of their construction; finally, we include a series of illustrative examples. The objective is to deliver a systematic exposure by starting with an origin of the concept, sound and well-elaborated analysis, and then proceeding with its realization pointing at ensuing practical implications.

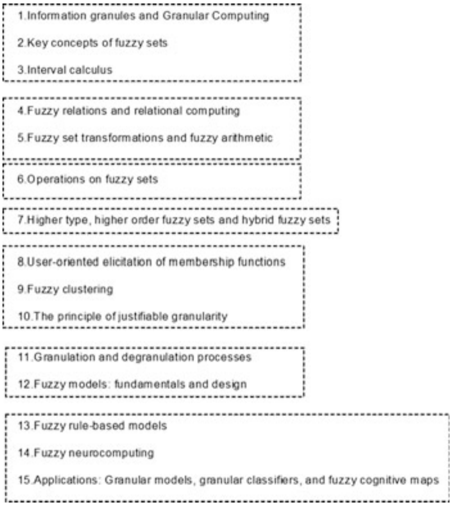


Fig. 1 Road map of the book with modules identified

It is obvious that neither any book nor textbook could completely cover the rich landscape of fuzzy set concepts, methodologies, algorithms and applications. We adhere to the principle of delivering the essentials that help the reader gain a sound picture *how* and *why* fuzzy sets work and acquire a working knowledge of how to use them. The book contains 15 chapters organized into several modules. Depending on the interest of the audience, several paths could be taken as succinctly illustrated in Fig. 1. The highlighted modules group chapters similar in terms of the underlying concepts.

The shortest path, which offers the most concise coverage of the ideas and working knowledge, consists of Chaps. 1, 2, 4, 5, 6, 8, 9, 11, 12 and 13. Depending on time allowed, one could add chapters on granular computing, and higher-order and higher-type fuzzy sets (Chap. 7), and expand studies on membership function estimation (Chap. 10). Those readers interested in application studies and advanced architectures of granular and fuzzy models can study Chap. 15.

There are several key features of the book worth highlighting, in particular

- casting concepts and algorithms of fuzzy sets in a general setting of granular computing, emphasizing diverse ways of building fuzzy sets (determining membership functions),
- delivering a systematic and comprehensive exposure to fuzzy modeling and fuzzy models,
- presenting direct linkages of the technology of fuzzy sets to the design and analysis of intelligent systems.

As the textbook, the material includes a wealth of carefully organized illustrative examples and systematic design guidelines. At the end of each chapter, we include a suite of problems of varying levels of difficulty with some of them could be regarded as an interesting research problem worth detailed investigation. Some sample examinations are included.

The book is self-contained. To assure broad accessibility of the material, the required prerequisites are limited to some introductory algebra, optimization and mathematical analysis. Appendix A brings about a brief and focused exposure to gradient-based optimization and population-based optimization in the form of particle swarm optimization. Appendix B includes samples of final examinations. References are limited to the most representative entries that the reader can easily access and gain a more thorough exposure to the subject matter.

In the realization of the project, I am indebted to a number of individuals. Professor Janusz Kacprzyk, Editor of the series, has been instrumental by offering his enthusiastic support and encouragement. Dr. Thomas Ditzinger, Editorial Director, has been a key person always eager to help realize the project in an efficient and timely manner. The professionals at Springer were always supportive by making sure that the project runs smoothly and on schedule.

Edmonton, Canada
May 2020

Witold Pedrycz

Contents

Chapter 1
Information Granules and Granular Computing

Abstract A tendency, which is being witnessed more visibly nowadays, concerns human centricity of intelligent systems. Data science and big data, Artificial Intelligence, computer vision and interpretation revolve around a two-way efficient interaction with users. The key facets of user centricity engage key features such as data quality, actionability, transparency. They are of highly relevance and are to be provided in advance. With this regard, information granules emerge as a sound conceptual and algorithmic vehicle owing to their way of delivering a more general view at data, ignoring irrelevant details and supporting a suitable level of abstraction aligned with the nature of the problem at hand. The objective of this chapter is to provide a general overview of Granular Computing, identify the main items on its agenda and highlight key application directions. We show linkages with the applied areas such as Explainable Artificial Intelligence.

Keywords Granular computing · Fuzzy sets · Transparency · Explainable artificial intelligence · User centricity

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_1

1.1 Information Granules and Information Granularity

Information granules are intuitively appealing constructs, which play a pivotal role in human cognitive faculties, system modeling and decision-making activities [1–5]. Briefly, information granules compose elements based on their observed similarity or resemblance or closeness. We perceive complex phenomena by organizing existing knowledge along with available experimental evidence and structuring them in a form of some meaningful, semantically sound entities, which are central to all ensuing processes of describing the world, reasoning about the environment, and support decision-making activities.

The terms information granules and information granularity themselves have emerged in different contexts and numerous areas of application. Information granule carries various meanings. One can refer to Artificial Intelligence (AI) in which case information granularity is central to a way of problem solving through problem decomposition, where various subtasks could be formed and solved individually. Information granules and the area of intelligent computing revolving around them being termed Granular Computing are quite often presented with a direct association with the pioneering studies by Zadeh [5]. He coined an informal, yet highly descriptive and compelling concept of information granules. Generally, by information granules one regards a collection of elements drawn together by their closeness (resemblance, proximity, functionality, etc.) articulated in terms of some useful spatial, temporal, or functional relationships. Subsequently, Granular Computing is about representing, constructing, processing, and communicating information granules. The concept of information granules is omnipresent and this becomes well documented through a series of applications.

Granular Computing exhibits a variety of conceptual developments; one may refer here to selected pursuits collected in Table 1.1 while some applications are shown in Table 1.2.

No matter which problem is taken into consideration, we usually set it up in a certain conceptual framework composed of some generic and conceptually meaningful entities—information granules, which we regard to be of relevance to the problem formulation, further problem solving, and a way in which the findings are communicated to the community. Information granules realize a framework in which we formulate generic concepts by adopting a certain level of generality.

Table 1.1 Selected conceptual developments of Granular Computing

Granular graphs
Information tables
Granular mappings
Knowledge representation
Micro and micro models
Association discovery and data mining

Table 1.2 Selected applications of Granular Computing

Forecasting time series
Prediction problems
manufacturing
Concept learning
optimization
Credit scoring
Analysis of microarray data
Autonomous vehicle

Information granules naturally emerge when dealing with data, including those coming in the form of data streams. The ultimate objective is to describe the underlying phenomenon in an easily understood way and at a certain suitable level of abstraction. The level of abstraction is crucial to problem solving: too many details, see Fig. 1.1, hamper the understanding of the problem and preventing from delivering an efficient, computationally sound and reliable solution.

This requires that we use a vocabulary of commonly encountered terms (concepts) and discover relationships between them as well as reveal possible linkages among the underlying concepts.

Information granules facilitate realizations of abstractions. As such they naturally give rise to hierarchical structures: the same problem or system can be perceived at different levels of specificity (detail) depending on the complexity of the problem, available computing resources, and particular needs to be addressed. A hierarchy of information granules is inherently visible in processing of data and knowledge. The level of captured details (which is represented in terms of the size of information granules) becomes an essential facet facilitating a way a hierarchical processing of information with different levels of hierarchy indexed by the size of information granules.

Even such commonly encountered and simple examples presented above are convincing enough to lead us to ascertain that

(a) information granules are the key components of knowledge representation and processing,

Fig. 1.1 Focusing on details and losing a global view at the recognition problem

(b) the level of granularity of information granules (their size, to be more descriptive) becomes crucial to the problem description and an overall strategy of problem solving,
(c) hierarchy of information granules supports an important aspect of perception of phenomena and delivers a tangible way of dealing with complexity by focusing on the most essential facets of the problem,
(d) there is no universal level of granularity of information; commonly the size of granules is problem-oriented and user dependent.

Human-centricity comes as an inherent feature of intelligent systems. It is anticipated that a two-way effective human-machine communication is imperative. Human perceive the world, reason, and communicate at some level of abstraction. Abstraction comes hand in hand with non-numeric constructs, which embrace collections of entities characterized by some notions of closeness, proximity, resemblance, or similarity. These collections are referred to as information granules. Processing of information granules is a fundamental way in which people process such entities. Granular Computing has emerged as a framework in which information granules are represented and manipulated by intelligent systems. The two-way communication of such intelligent systems with the users becomes substantially facilitated because of the usage of information granules.

It brings together the existing plethora of formalisms of set theory (interval analysis, fuzzy sets, rough sets, etc.) under the same banner by clearly visualizing that in spite of their visibly distinct underpinnings (and ensuing processing), they exhibit some fundamental commonalities. In this sense, Granular Computing establishes a stimulating environment of synergy between the individual approaches. By building upon the commonalities of the existing formal approaches, Granular Computing helps assemble heterogeneous and multifaceted models of processing of information granules by clearly recognizing the orthogonal nature of some of the existing and well-established frameworks (say, probability theory coming with its probability density functions and fuzzy sets with their membership functions). Granular Computing fully acknowledges a notion of variable granularity, whose range could cover detailed numeric entities and very abstract and general information granules. It looks at the aspects of compatibility of such information granules and ensuing communication mechanisms of the granular worlds. Granular Computing gives rise to processing that is less time demanding than the one required when dealing with detailed numeric processing. The concept of modularity is emphasized: to tackle a difficult problem, it is usually partitioned into a collection of subproblems and dealt with each of them individually.

1.2 Frameworks of Information Granules

There are numerous formal frameworks of information granules; for illustrative purposes, we recall some selected alternatives.

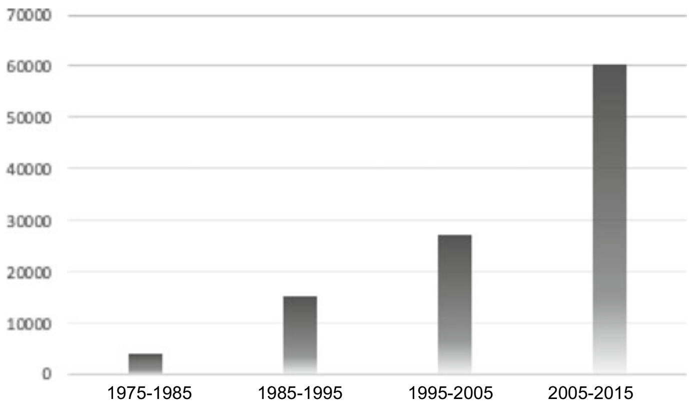


Fig. 1.2 Publications in the area of fuzzy sets (Science Direct https://www.sciencedirect.com/)

Sets (*intervals*) realize a concept of abstraction by introducing a notion of dichotomy: we admit element to fully belong to a given information granule or to be excluded from it. Along with the set theory comes a well-developed discipline of interval analysis [6–8].

Fuzzy sets deliver an important conceptual and algorithmic generalization of sets [9–11]. By admitting partial membership of an element to a given information granule, we bring an important feature which makes the concept to be in rapport with reality. As shown in Fig. 1.2, fuzzy sets have demonstrated a dynamic growth. They help working with the notions, where the notion of dichotomy is neither justified, nor advantageous [12–14].

Shadowed sets offer an interesting description of information granules by distinguishing among three categories of elements [15, 16]. Those are the elements, which (i) fully belong to the concept, (ii) are excluded from it, and (iii) their belongingness is completely *unknown.*

Rough sets are concerned with a roughness phenomenon, which arises when an object (pattern) is described in terms of a limited vocabulary of certain granularity [17–19]. The description of this nature gives rise to a so-called lower and upper bound forming the essence of a rough set.

The list of formal frameworks is quite extensive; as interesting examples, one can recall here probabilistic sets [20] and axiomatic fuzzy sets [21].

Example 1 Consider a collection of one-dimensional numeric data. Describe it by an information granule expressed in the form of a histogram.

Having data $\{x_1, x_2, \ldots, x_N\}$, we place them into p bins of the same length. In each of them falls n_i data. Thus the description of the information granule can be formed as a c-dimensional vector of probabilities $\boldsymbol{p} = [p_1\ p_2 \ldots p_c]$ where $pi = n_i/N$.

Alternatively, we can form bins of varying length requiring that each of them includes the same proportion of data c. Here the information granule is described by providing the cut-off points of the bins $\boldsymbol{z} = [z_1\ z_2 \ldots z_c]$.

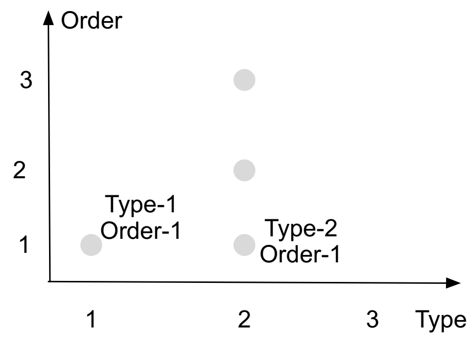


Fig. 1.3 Diversity of types and orders of information granules

Hybrid constructs are developed to cope with the diversity and complexities of real-world phenomena. Consider a statement

$$high \text{ probability of } low \text{ precipitation and above } normal \text{ temperatures}$$

Here different facets of information granularity are dealt with and give rise to constructs where both probability and fuzzy sets are considered at the same time.

There are two important directions of generalizations of information granules, namely information granules of higher *type* and information granules of higher *order*. The essence of information granules of higher type means that the characterization (description) of information granules is described in terms of information granules rather than numeric entities. Well-known examples are fuzzy sets of type-2, granular intervals, imprecise probabilities. For instance, a type-2 fuzzy set is a fuzzy set whose grades of membership are not single numeric values (membership grades in [0,1]) but fuzzy sets, intervals or probability density functions truncated to the unit interval. There is a hierarchy of higher type information granules, which are defined in a recursive manner. Therefore we talk about type-0, type-1, type-2 fuzzy sets, etc. In this hierarchy, type-0 information granules are numeric entities, say, numeric measurements. With regard to higher order information granules are granules defined in some space whose elements are information granules themselves. The diversity of the landscape of information granules is clearly visible as outlined in Fig. 1.3.

1.3 Explainable Artificial Intelligence

In the recent years, Artificial Intelligence has emerged as an important, timely, and far reaching research discipline with a plethora of advanced and highly visible applications. With the rapid progress of concepts and methods of AI, there is a recent and vivid trend to augment the paradigm by bringing aspects of explainability. With the ever growing complexity of AI constructs their relationships with data analytics (and inherent danger of cyberattacks and the presence of adversarial data)

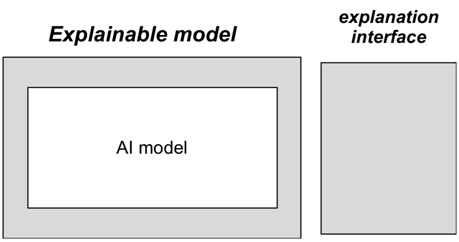


Fig. 1.4 Explanation module and explanation interface of AI architectures

and the omnipresence of demanding applications in various criteria domains, there is a growing need to associate the results with sound explanations All of these factors have given rise to the most recent direction of Explainable AI (XAI for brief) [22–25] to foster the developments of XAI architectures and augment AI with the facets of human centricity, which becomes indispensable. To make the results interpretable and furnish with the required facet of explainability, one argues that the findings have to be delivered at a certain level of abstraction (general perspective)–as such information granularity and information granules play here a pivotal role.

It is desirable that the models of AI are transparent so that the results being produced have to be easily interpretable and explainable. There have been a number of studies emphasizing that opaque constructs of artificial neural networks including deep learning and ways of bringing the aspect of transparency. In a nutshell, this research direction is ultimately aimed at pursuing a new original avenue of XAI by bringing the rich methodology and a plethora of algorithmic developments of Granular Computing to the area of AI and delivering concepts of granular models. A general architecture is portrayed in Fig. 1.4 where we stress the role of Granular Computing by visualizing the formation of explainable models and an explanation interface formed on top of the generic AI constructs.

We advocate that the main faculties of explainable AI architectures involve three key features which are the main contributors to the functional modules identified in Fig. 1.4:

(i) formation of a well-delineated logic blueprint of the AI model; the logic-oriented skeleton of the architecture plays a pivotal role,
(ii) mapping data (numeric evidence) into symbols of far higher level of abstraction,
(iii) incorporation of an adjustable level of abstraction being in rapport with the formulated problem.

The aspect of interpretation (according to Merriam-Webster dictionary, interpret—to explain or to present in understandable terms to a human) associates directly to cognitive chunks that can be represented as information granules and regarded as the generic units building an interpretation layer.

With this regard, several aspects are to be raised, for instance:

Form of cognitive chunks. What are the basic units of the explanation? Are they raw or transformed features? If the transformed (derived) features are considered, could they exhibit any semantics

Composition and hierarchical aspects Are the cognitive chunks organized in a structured way? Could this structured way come with some limits? An explanation may involve defining a new unit (a chunk) that is a function of raw units, and then provide an explanation in terms of that new unit.

1.4 Image Understanding and Computer Vision

Image understanding is one of the core problems of computer vision. Its goal is to support understanding and labeling input image given a set of labels. Scene recognition is also referred to as scene classification. It has gained importance and produced a wide range of applications to remote sensing, image processing, and intelligent monitoring, among others. The challenge in the area comes because of the two evident and commonly encountered reasons:

(i) image contains a lot of objects, and
(ii) final labeling is subjective and usually of non-binary form which is very much in rapport with a human perception.

Firstly, the data involved in any scene recognition task are more complicated. Each image contains a variety of objects, background information, and scene attributes. There are also some relationship among different objects. As shown in Fig. 1.5, the "standard" classification problem involves only a single object (aircraft). In contrast, scene classification is far more demanding: there are a lot of objects difficult to recognize, there are numerous relationships (relations) among them. All of these items have to be dealt with in a comprehensive manner. There is a diversity of objects and this makes the classification to fall under the rubric of multi-label classification. Quite commonly the labeling is not binary (yes-no) but associated with some quantification in terms of partial membership.

1.5 Data Analytics

Data analytics is commonly regarded as a conceptual and algorithmic faculty that starts with a usually vast amount of raw data in order to make actionable conclusions about information contained there. The critical feature of data analytics is the transparency and a high level of comprehension of the generated findings. Several categories of main data analytics tasks are considered

(i) Descriptive analytics describes what has occurred over a given period of time.
(ii) Diagnostic analytics attempts to reveal why some situation happened
(iii) Predictive analytics is concerned with prediction (forecasting) about the future behavior of the system
(iv) Prescriptive analytics offers some decisions as to the future course of action.

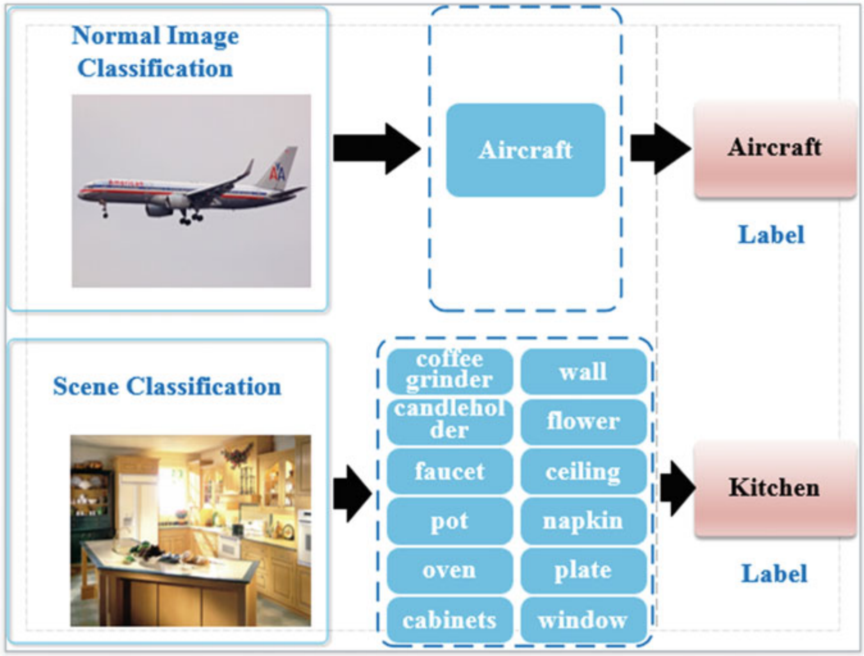


Fig. 1.5 Image classification versus scene interpretation

It is apparent that in all these four categories identified above the user centricity plays a visible role. Sound descriptive analytics requires a summarization of historical data so that main relationships could be easily understood. The same applies to diagnostic analytics. Prediction activities are required to be carried out at a certain level of generality; the level of abstraction is present here.

In summary, the results delivered by methods of data analytics as well as the context being established to carry out a variety of tasks of data analytics, AI and image processing and understanding associate with information granules as highlighted in Fig. 1.6. The interfaces composed of information granules are problem-oriented and user-oriented with intent to deliver interesting and actionable results.

For instance, information granules help deliver the results that are easily interpretable and strongly supported by existing experimental evidence, say *high* rising costs of real estate expected in the first *few* months of the coming year. The two information granules concern the quantification of attribute (costs) and time (months). The user can formulate a path of interesting analysis. Instead of forming a general request such as

describe a structure of customers of financial institution

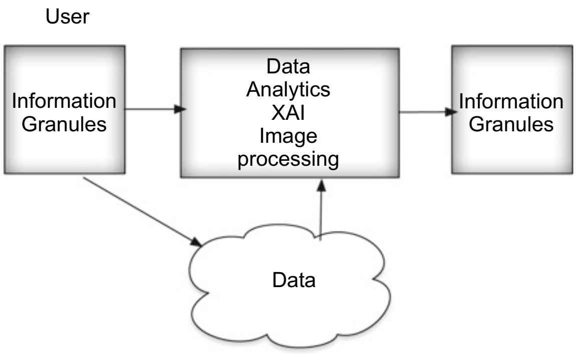


Fig. 1.6 A general processing framework with interfaces composed of information granules

where the task is carried out based on collected data, one expresses the more focused (context-based) request.

describe a structure of $medium$ income customers of financial institutions

An idea and algorithms of linguistic summarization [26, 27] play a visible role in this setting.

1.6 Conclusions

The introductory notes identify the essence of information granules, the idea of information granularity and processing of information granules. The diverse landscape of formal frameworks delivers a wealth of approaches that help establish a suitable level of abstraction and focus on processing information granules. The visible role of information granules has been identified in a spectrum of applications, in particular in the three strategically positioned areas of Artificial Intelligence, data analytics, and computer vision.

Problems

1. Provide some examples of information granules encountered in real-world environment and suggest a suitable formal way of their representation.
2. How could you formalize an expression *low* probability of *high* electricity consumption.
3. Suggest 2–3 convincing arguments behind the emergence of XAI.
4. Which of the concepts could be described in terms of sets (intervals): odd numbers, low approximation error, speed of 55 km/h.

References

1. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Dordrecht (2003)
2. Bargiela, A., Pedrycz, W.: Toward a theory of granular computing for human-centered information processing. IEEE Trans. Fuzzy Syst. **16**(2), 320–330 (2008)
3. Pedrycz, W.: Granular Computing. CRC Press, Boca Raton, Fl (2013)
4. Pedrycz, W.: Granular computing for data analytics: a manifesto of human-centric computing. IEEE/CAA J. Autom. Sinica **5**, 1025–1034 (2018)
5. Zadeh, L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst. **90**, 111–117 (1997)
6. Alefeld, G., Herzberger, J.: Introduction to Interval Computations. Academic Press, New York (1983)
7. Moore, R.: Interval Analysis. Prentice Hall, Englewood Cliffs (1966)
8. Moore, R., Kearfott, R.B., Cloud, M.J.: Introduction to Interval Analysis. SIAM, Philadelphia (2009)
9. Pedrycz, W., Gomide, F.: Fuzzy Systems Engineering: Toward Human-Centric Computing. John Wiley, Hoboken, NJ (2007)
10. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 33–353 (1965)
11. Zadeh, L.A.: The concept of linguistic variables and its application to approximate reasoning I, II, III. Inform. Sci. **8** 199–249, 301–357, 43–80 (1975)
12. Dubois, D., Prade, H.: Outline of fuzzy set theory: An introduction. In: Gupta, M.M., Ragade, R.K., Yager, R.R. (eds.) Advances in Fuzzy Set Theory and Applications, pp. 27–39. North-Holland, Amsterdam (1979)
13. Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice-Hall, Upper Saddle River (1995)
14. Nguyen, H., Walker, E.: A First Course in Fuzzy Logic. CRC Press, Boca Raton, Chapman Hall (1999)
15. Pedrycz, W.: Shadowed sets: representing and processing fuzzy sets. IEEE Trans. Syst. Man Cybern. Part B **28**, 103–109 (1998)
16. Pedrycz, W.: Interpretation of clusters in the framework of shadowed sets. Pattern Recogn. Lett. **26**(15), 2439–2449 (2005)
17. Pawlak, Z.: Rough sets. Int. J. Inf. Comput. Sci. **11**(15), 341–356 (1982)
18. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers, Dordrecht (1991)
19. Pawlak, Z.: Rough sets and fuzzy sets. Fuzzy Sets Syst. **17**(1), 99–102 (1985)
20. Hirota, K.: Concepts of probabilistic sets. Fuzzy Sets Syst. **5**(1), 31–46 (1981)
21. Liu, X., Pedrycz, W.: Axiomatic Fuzzy Set Theory and Its Applications. Springer, Berlin (2009)
22. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
23. Escalante, H.J., et al. (eds.): Explainable and Interpretable Models in Computer Vision and Machine Learning. Springer, Cham (2018)
24. Gunning, D.: Explainable artificial intelligence (XAI), Defense Advanced Research Project Agency (DARPA) (2017)
25. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2014). arXiv:1412.6572
26. Kacprzyk, J., Zadrozny, S.: Computing with Words is an implementable paradigm: fuzzy queries, linguistic data summaries, and natural-language generation. IEEE Trans. Fuzzy Syst. **18**, 461–472 (2010)
27. Kacprzyk, J., Yager, R.R., Merigo, J.M.: Towards Human-centric aggregation via ordered weighted aggregation operators and linguistic data summaries: a new perspective on Zadeh's inspirations. IEEE Comput. Intell. Mag. **14**, 15–30 (2019)

Chapter 2
Key Concepts of Fuzzy Sets

Abstract This chapter covers the main concepts of fuzzy sets, offers illustrative examples and brings some algorithmic developments essential to the characterization and processing fuzzy sets.

Keywords Sets · Interval analysis · Fuzzy sets · Characterization · Dichotomy · Membership function · Representation theorem · Entropy · Energy · Specificity

2.1 Sets and Interval Analysis

Sets and set theory are the fundamental notions of mathematics, science, and engineering. They are in common usage when describing a wealth of concepts, identifying relationships, and formalizing solutions. The underlying fundamental notion of set theory is that of *dichotomy*: a certain element belongs to a set or is excluded from it. A universe of discourse $\boldsymbol{X}$ over which a set or sets are formed could be very diversified depending upon the nature of the problem.

Given a certain element in a universe of discourse $\boldsymbol{X}$, a process of dichotomization (binarization) imposes a binary, *all-or-none* classification decision: we either accept

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_2

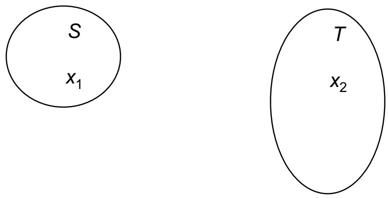


Fig. 2.1 Examples of sets S and T

or reject the element as belonging to a given set. For instance, consider the set S shown in Fig. 2.1. The point x_1 belongs to S whereas x_2 does not, that is, $x_1 \in S$ and $x_2 \notin S$. Similarly, for set T we have $x_1 \notin T$, and $x_2 \in T$.

If we denote the acceptance decision about the belongingness of the element by 1 and the reject decision (non-belongingness) by 0, we express the classification (assignment) decision through a characteristic function. In general, a characteristic function of set A defined in $\boldsymbol{X}$ assumes the following form

$$A(x) = \begin{cases} 1, & \text{if } x \in \mathrm{A} \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

The empty set Ø has a characteristic function that is identically equal to zero, Ø(x) = 0 for all x in $\boldsymbol{X}$. The universe $\boldsymbol{X}$ itself comes with the characteristic function that is identically equal to one, that is $\boldsymbol{X}(x) = 1$ for all x in $\boldsymbol{X}$. Also, a singleton $A = \{a\}$, a set comprising only a single element, has a characteristic function such that $A(x) = 1$ if $x = a$ and $A(x) = 0$ otherwise.

Characteristic functions A: $\mathbf{X} \to \{0, 1\}$ induce a constraint with well-defined binary boundaries imposed on the elements of the universe $\mathbf{X}$ that can be assigned to a set A. By looking at the characteristic function, we see that all elements belonging to the set are non-distinguishable—they come with the same value of the characteristic function so by knowing that $A(x_1) = 1$ and $A(x_2) = 1$ we cannot tell these elements apart. The operations of union, intersection, and complement are easily expressed in terms of the characteristic functions. The characteristic function of the union comes as the maximum of the characteristic functions of the sets involved in the operation. The complement of A denoted by $\overline{A}$, comes with a characteristic function equal to $1 - A(x)$.

Two-valued logic and sets are isomorphic. Logic statements (propositions) assume truth values of false (0) and true (1). Logic operations of *and* and *or* are isomorphic with the operations of union and intersection defined for sets. The negation flips the original truth value of the proposition.

2.2 Fuzzy Sets: A Departure from the Principle of Dichotomy

Conceptually and algorithmically, fuzzy sets constitute one of the most fundamental and influential notions in science and engineering. The notion of a fuzzy set is highly intuitive and transparent since it captures what really becomes an essence of a way in which a real world is being perceived and described in our everyday activities. We are faced with objects whose belongingness to a given category (concept) is always a matter of degree. There are numerous examples, in which we encounter elements whose allocation to the concept we want to define can be satisfied to some degree. One may eventually claim that continuity of transition from full belongingness and full exclusion is the major and ultimate feature of the physical world and natural systems. For instance, we may qualify an in-door environment as *comfortable* when its temperature is kept *around* 20 °C. If we observe a value of 19.5 °C it is very likely we still feel quite *comfortable*. The same holds if we encounter 20.5 °C—humans usually do not discriminate between changes in temperature within the range of one degree Celsius. A value of 20 °C would be fully compatible with the concept of *comfortable* temperature yet 0 or 30 °C would not. In these two cases as well for temperatures close to these two values, we would describe them as being *cold* and *warm*, respectively. We could question whether the temperature of 25 °C is viewed as *warm* or *comfortable* or, similarly, if 15 °C is *comfortable* or *cold*. Intuitively, we know that 25 °C is somehow between *comfortable* and *warm* while 15 °C is between *comfortable* and *cold*. The value 25 °C is partially compatible with the term *comfortable* and *warm*, and somewhat compatible or, depending on observer's perception, incompatible with the term of *cold* temperature. Similarly, we may say that 15 °C is partially compatible with the *comfortable* and *cold* temperature, and slightly compatible or incompatible with the warm temperature. In spite of this highly intuitive and apparent categorization of environment temperatures into the three classes, namely *cold*, *comfortable* and *warm*, we note that the transition between the classes is not instantaneous and sharp (binary). Simply, when moving across the range of temperatures, these values become gradually perceived as *cold*, *comfortable* or *warm*. These simple and apparent examples support a clear message: the concept of dichotomy does not apply when defining even simple concepts. The reader can easily supply a huge number of examples as they commonly appear in real-world situations.

The fundamental idea of fuzzy set is to relax this requirement by admitting intermediate values of class membership. Therefore, we may assign intermediate values between 0 and 1 to quantify our perception on how compatible these values are with the class (concept) with 0 meaning incompatibility (complete exclusion) and 1 compatibility (complete membership). Membership values thus express the degrees to which each element of the universe is compatible with the properties distinctive to the class. Intermediate membership values underline that no "natural" threshold exists and that elements of a universe can be members of a class and at the same time belong to other classes with different degrees. Allowing for gradual, hence less strict non-binary membership degrees is the crux of fuzzy sets.

Formally, a fuzzy set A [4] is described by a membership function mapping the elements of a universe $\boldsymbol{X}$ to the unit interval [0, 1].

$$A : \boldsymbol{X} \rightarrow [0, 1] \tag{2.2}$$

The membership functions are therefore synonymous of fuzzy sets. In a nutshell, membership functions generalize characteristic functions in the same way as fuzzy sets generalize sets.

The choice of the unit interval for the values of membership degrees is usually a matter convenience and semantics. The specification of very precise membership values (up to several decimal digits), say $A(4) = 0.9865$, is not crucial or even counter-productive. We should stress here that in describing membership grades we are predominantly after the reflecting an order of the elements in A in terms of their belongingness to the fuzzy set [1].

Being more descriptive, we may view fuzzy sets as elastic constraints imposed on the elements of a universe. As emphasized before, fuzzy sets deal primarily with the concept of elasticity, graduality, or absence of sharply defined boundaries. In contrast, when dealing with sets we are concerned with rigid boundaries, lack of graded belongingness and sharp, binary boundaries. Gradual membership means that no natural boundary exists and that some elements of the universe of discourse can, contrary to sets, belong to different fuzzy sets with different degrees of membership.

In light of definition (2), fuzzy sets can be also be viewed as a family of ordered pairs of the form $\{x, A(x)\}$ where x is an element of $\boldsymbol{X}$ and $A(x)$ denotes its corresponding degree of membership. For a finite universe of discourse $\boldsymbol{X} = \{x_1, x_2,\ldots, x_n\}$, A can be represented by a n-dimensional vector $A = [a_1, a_2,\ldots, a_n]$ where $a_i = A(x_i)$. Figure 2.2 illustrates a fuzzy set whose membership function captures the concept of integer *around* 5. Here $n = 7$ and $A = [0, 0, 0, 0.2, 0.5, 1.0, 0.5, 0.2, 0, 0, 0]$. An equivalent notation of A which explicitly includes the elements of $\boldsymbol{X}$ of can be read as $A = [0/1, 0/2, 0/3, 0.2/4, 0.5/4, \ldots, 0/10]$.

For the n elements space $\boldsymbol{X}$, sets and fuzzy sets come with an interesting geometric interpretation as points in the $[0, 1]^n$ hypercube. In the simplest case when $\boldsymbol{X} = \{x_1, x_2\}$; $n = 2$, the family of sets $P(\boldsymbol{X})$ consists of the empty set Ø, $\{x_1\}$, $\{x_2\}$, and $\{x_1, x_2\}$. The cardinality of $\boldsymbol{X}$ is $2^2 = 4$. Each of the four elements of this family can be represented by a two dimensional vector, say Ø = [0, 0], $\{x_1\} = [1, 0]$, $\{x_2\} = [0,$

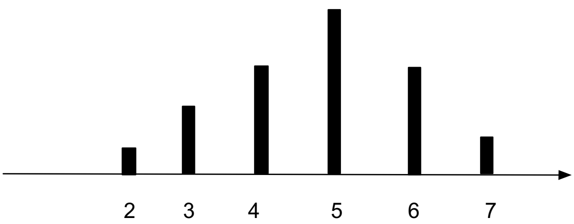


Fig. 2.2 Fuzzy set A defined in finite space $\boldsymbol{X}$

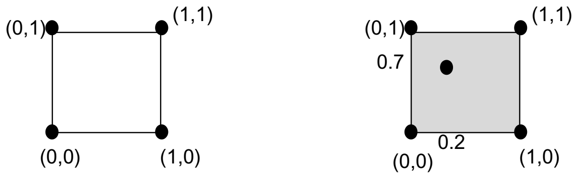


Fig. 2.3 Geometric interpretation of sets and fuzzy sets

1], and $\{x_1, x_2\} = [1, 1]$ located at the corners of the unit square, as illustrated in Fig. 2.3.

In contrast, fuzzy sets being two-dimensional vectors are distributed throughout the entire unit square. For instance, fuzzy set A is represented as a vector [0.2 0.7]. A family of fuzzy sets $F(\boldsymbol{X})$ occupies the whole shaded area, including the interior and the corners of the unit square. The same geometric interpretation occurs when n assumes higher values.

2.3 Classes of Membership Functions

Formally speaking, any function A: $\mathbf{X} \rightarrow [0, 1]$ could be qualified to serve as a membership function describing the corresponding fuzzy set. In practice, the form of the membership functions should be reflective of the problem at hand for which we construct fuzzy sets. They should reflect our perception (semantics) of the concept to be represented and further used in problem solving, the level of detail we intend to capture, and a context, in which the fuzzy set are going to be used. It is also essential to assess the type of a fuzzy set from the standpoint of its suitability when handling the ensuing optimization procedures. It also needs to accommodate some additional requirements arising as a result of further needs optimization procedures such as differentiability of membership functions. Given these criteria in mind, we elaborate on the most commonly used categories of membership functions. All of them are defined in the universe of real numbers, that is $\boldsymbol{X} = \boldsymbol{R}$.

Triangular membership functions. The fuzzy sets are expressed by their piecewise linear segments described in the form

$$A(x;\ a,\ m,\ b) = \begin{cases} 0, & \text{if } x \le a \\ \frac{x-a}{m-a}, & \text{if } x \in [a, m] \\ \frac{b-x}{b-m}, & \text{if } x \in [m,\ b] \\ 0, & \text{if } x \ge b \end{cases} \tag{2.3}$$

The meaning of the parameters is straightforward: m denotes a modal (typical) value of the fuzzy set while a and b stand for the lower and upper bounds, respectively. They could be sought as the extreme elements of the universe of discourse that delineate the elements belonging to A with nonzero membership degrees. Triangular fuzzy sets (membership functions) are the simplest possible models of membership

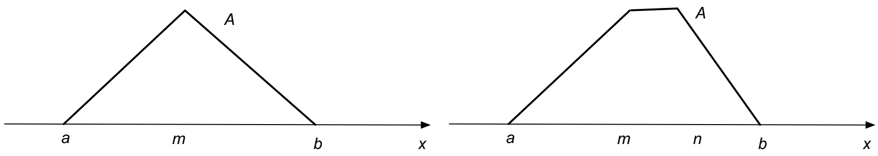


Fig. 2.4 Example of triangular and trapezoidal membership functions

functions. They are fully defined by only three parameters. As mentioned, the semantics is evident as the fuzzy sets are expressed on a basis of knowledge of the spreads of the concepts and their typical values. The linear change in the membership grades is the simplest possible model of membership one could think of (Fig. 2.4).

Trapezoidal membership functions. They are piecewise linear function characterized by four parameters, namely a, m, n, and b specifying a location of the four linear parts of the membership function. The membership functions assume the following form:

$$A(x;\ a,\ m,\ b) = \begin{cases} 0, & \text{if } x \le a \\ \frac{x-a}{m-a}, & \text{if } x\ \in [a, m] \\ 1, & \text{if } x\ \in [m, n] \\ \frac{b-x}{b-n}, & \text{if } x\ \in [n, b] \\ 0, & \text{if } x\ \ge b \end{cases} \tag{2.4}$$

The elements falling within the range $[m, n]$ are indistinguishable as this region is described by a characteristic function.

S-membership functions. These are the functions described in the following way

$$A(x; a, b) = \begin{cases} 0, & \text{if } x \le a \\ 2\left(\frac{x-a}{b-a}\right)^2 & \text{if } x \in [a, m] \\ 1 - 2\left(\frac{x-b}{b-a}\right)^2, & \text{if } x \in [m, b] \\ 1, & \text{if } x \ge b \end{cases} \tag{2.5}$$

The point $m = (a + b)/2$ is referred to as the crossover point.

Gaussian membership functions. These membership functions are described by the following relationship:

$$A(x; m, \sigma) = \exp\left(-(x - m)^2/\sigma^2\right) \tag{2.6}$$

Gaussian membership functions are characterized by the two parameters. The modal value m represents the typical element of A (with the membership value equal to 1) while σ denotes a spread of A. Higher values of σ corresponds to larger spreads of the fuzzy sets.

Piecewise membership functions. As the name stipulates, these membership functions are described by pieces of linear functions as illustrated in Fig. 2.5. One regard

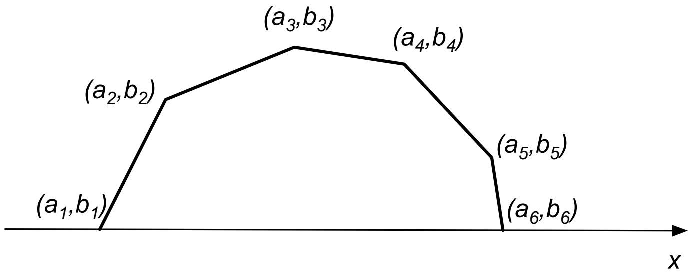


Fig. 2.5 Example of piecewise membership function $A = A(x; (a_1, b_1), (a_2, b_2),\ldots, (a_6, b_7))$

that some points are specified during the elicitation of the membership function as having some specific membership grades. In-between these points, we consider the simplest linear way of changes in the membership grades. The description of A is given by listing the locations and values of the corresponding membership grades (a_i, b_i).

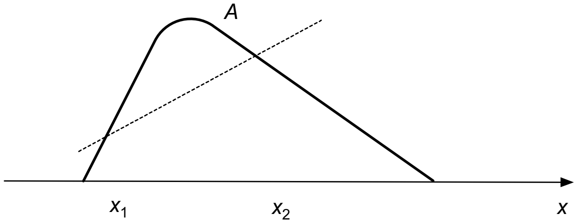
Fuzzy sets can be categorized as convex or nonconvex fuzzy sets A fuzzy set is convex if its membership function satisfies the following condition: For any $x_1, x_2 \in$ $\boldsymbol{X}$ and all $\lambda \in [0, 1]$ one has

$$A(\lambda x_1 + (1 - \lambda)x_2)\min(A(x_1), A(x_2)) \tag{2.7}$$

The above inequality states that, whenever we choose a point x on a line segment in-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 Fig. 2.6.

Membership functions can be looked at from the perspective of their property of sensitivity which quantifies how much changes in membership grades are encountered when looking at different regions of the space. As expected, the sensitivity measure of A can be regarded as a derivative of A (assuming that A is differentiable), namely $dA(x)/dx$ or the absolute value $|dA(x)/dx|$. For instance, for the triangular membership we have two regions of X with the sensitivity implied by the slope of the corresponding segments of the linear function. The piecewise membership function offers more flexibility as the sensitivity associates with the slopes of the linear parts of the membership functions.

Fig. 2.6 An example of a convex fuzzy set A



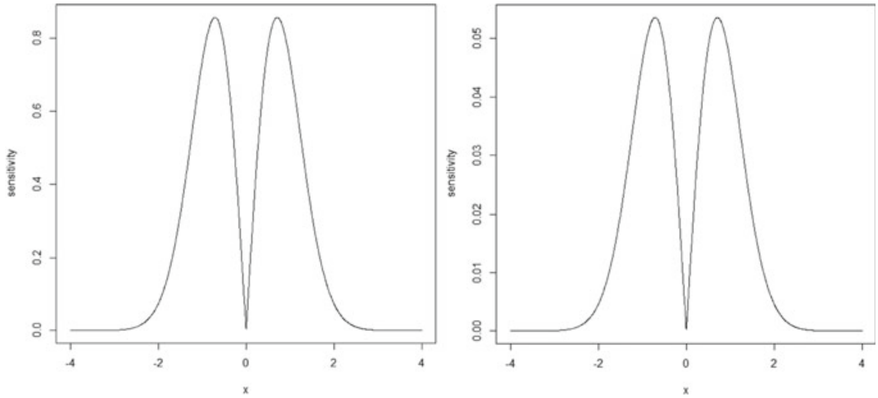


Fig. 2.7 Sensitivity of the Gaussian membership function (a) $\sigma = 1$, and (b) $\sigma = 2$

Example 1 Determine the sensitivity measure of the Gaussian membership function $\exp(-x^2/\sigma^2)$ for $\sigma = 1$ and $\sigma = 2$. Interpret the obtained results.

The sensitivity is computed as $|dA/dx| = 2A(x)x/\sigma^2$. The plots of this measure are shown in Fig. 2.7. It is apparent that sensitivity assumes its highest values at two points x_1 and x_2 located symmetrically around the origin. Those are two points where the membership grade is in-between 0 and 1 and whose determination is impacted by the highest level of sensitivity. The sensitivity assumes a zero value for $x = 0$ where also the maximal value of membership function is observed. The sensitivity starts to decline when we move towards higher and lower values of x. Note also that the pattern of sensitivity does not change when varying the values of σ however the values of sensitivity start to become lower for increasing values of the spread.

Example 2 Determine a membership function of A knowing that the change of degree "x is A" is proportional to the degree of x is A and the degree of x is not A.

The requirement translates into a differential equation in the form

$$dA(x)/dx = A(x)(1 - A(x)) \tag{2.8}$$

One can easily demonstrate that the sigmoid membership function $A(x) = 1/(1 + \exp(x))$ satisfies the above differential equation.

2.4 Selected Descriptors of Fuzzy Sets

Given the enormous diversity of potentially useful (semantically sound) membership functions, there are certain concise 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.

Normality: We say that the fuzzy set A is *normal* if its membership function attains 1, that is

$$\sup_{x \in X} A(x) = 1 \tag{2.9}$$

If this property does not hold, we call the fuzzy set *subnormal*. The supremum operation (sup) standing in the above expression is also referred to as a height of the fuzzy set A, hgt(A) = $\sup_{x \in X} A(x) = 1$.

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 one states that there is at least one element in **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.

The normalization operation, norm(A) is a transformation mechanism that is used to convert a subnormal nonempty 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

$$\mathrm{norm}(A) = \frac{A(x)}{\mathrm{hght}(A)} \tag{2.10}$$

Support Support of a fuzzy set A, denoted by supp(A), is a set of all elements of $\boldsymbol{X}$ with nonzero membership degrees in A

$$\mathrm{supp}(A) = \{x \in \boldsymbol{X} | A(x) > 0\} \tag{2.11}$$

Core The core of a fuzzy set A, core(A), is a set of all elements of the universe that are typical for A viz. they come with membership grades equal to 1,

$$core(A) = \{x \in \boldsymbol{X} | A(x) = 0\} \tag{2.12}$$

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 be also interested in characterizing sets of elements that come with some intermediate membership degrees. Both the support and core of any fuzzy sets are sets (Fig. 2.8).

A notion of an 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 α ∈ [0, 1] [2]. Formally speaking, we describe the α-cut of A as a set

$$A_a = \{x \in X | A(x) \geq \alpha\} \tag{2.13}$$

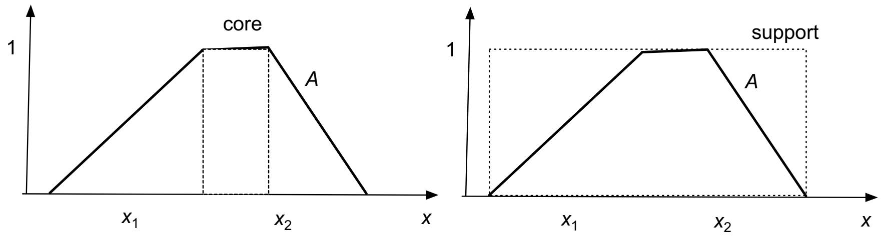


Fig. 2.8 Core and support of membership functions: characteristic functions of core and support shown by dotted lines

A strong α-cut differs from the α-cut in the sense that it identifies all elements in $\boldsymbol{X}$ for which we have the following equality $A_{a+} = \{x \in X | A(x) > \alpha\}$

An illustration of the concept of the α-cut and strong α-cut is presented in Fig. 2.9. Both support and core are limit cases of α-cuts and strong α-cuts. For α = 0 and the strong α-cut, we arrive at the concept of the support of *A*. The threshold α = 1 states that the corresponding α-cut is the core of *A*.

We can characterize fuzzy sets by counting their elements and brining 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 universe $\boldsymbol{X}$, its cardinality, denoted by card(*A*), is expressed as the following sum

$$\mathrm{card}(A) = \sum_{x \in \boldsymbol{X}} A(x) \tag{2.14}$$

or alternatively as the following integral

$$\mathrm{card}(A) = \int_{x \in \boldsymbol{X}} A(x) \tag{2.15}$$

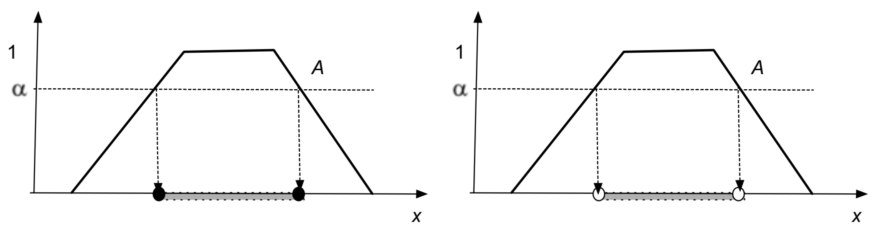


Fig. 2.9 Examples of α-cut and strong α-cut of A

(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 card(A) = |A| and refer to it as a sigma count (σ-count) of A.

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 cardinality come with the higher level of abstraction (generalization) and the lower values of granularity (specificity). The concept of cardinality is linked the notion of specificity of an information granule.

Specificity The notion of specificity is linked with a way of expressing a level of detail conveyed by the fuzzy set. Specificity quantifies a level of detail conveyed by information granule. Before introducing specificity of a fuzzy set, let us formulate a suitable measure for an interval. For a given interval A, we consider a specificity measure to be a decreasing function of the length of A, sp(A) = g(length(A)). Furthermore specificity for a single element A, $A = \{x_0\}$ attains maximal value equal to 1. For instance, a simple example is a linearly decreasing function of the length,

$$\mathrm{sp}(A) = 1 - \mathrm{length}(A)/\mathrm{range} \tag{2.16}$$

where *range* is a length of the entire universe of discourse considered in the problem, say an interval of feasible values of the variable encountered in the problem, see Fig. 2.10. For (2.16) $\mathrm{sp}(\{x_0\}) = 1$ and sp(*range*) = 0; refer to Fig. 2.11.

Some other example of g can be taken as exp(−length(A)).

The specificity of the fuzzy set can be introduced by taking into account the representation theorem: we determine specificity of α-cut of A and then integrate (or sum up the results)

Fig. 2.10 Determining specificity of A

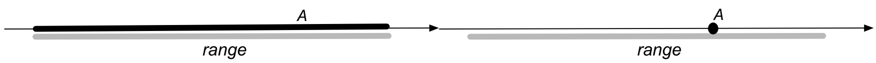


Fig. 2.11 Boundary examples of A

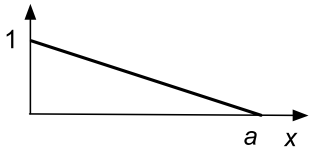


Fig. 2.12 Triangular membership function

$$\mathrm{sp}(A) = \int_0^1 sp(A_\alpha)d\alpha \tag{2.17}$$

Example 3 Determine specificity of the triangular fuzzy set A shown in Fig. 2.12.

We start with the determination of the specificity of some *a*-cut. We have $1 - x_\alpha/a = \alpha$.

So $x_\alpha = a(1 - \alpha)$ and $\mathrm{sp}(A_\alpha) = 1 - a(1 - \alpha)/range$. Integrating over all values of α ranging from 0 to 1, one derives $\mathrm{sp}(A) = 1 - a/(2range) = 1 - \frac{1}{2}\frac{\mathrm{a}}{\mathrm{range}}$..

So far we discussed properties of a single fuzzy set. The operations to be studied look into the characterizations of relationships between two fuzzy sets.

Equality: We say that two fuzzy sets A and B defined in the same space $\boldsymbol{X}$ are equal if and only if their membership functions are identical, meaning that

$$A(x) = B(x)\forall \mathrm{x} \in \mathbf{X} \tag{2.18}$$

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 (Fig. 2.13)

$$A(x) \le B(x)\forall \mathrm{x} \in \mathbf{X} \tag{2.19}$$

Fuzzy sets can be characterized by counting their elements and brining 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.

Energy measure of fuzziness of a fuzzy set A in $\mathbf{X}$, denoted by $E(A)$, is a functional of the membership degrees

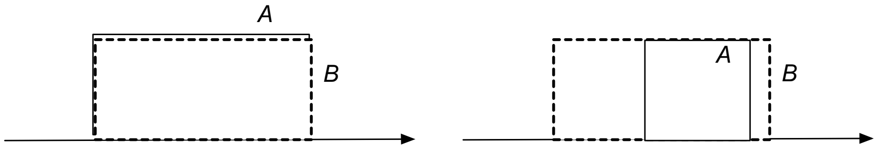


Fig. 2.13 Characteristic functions of equality and inclusion of sets

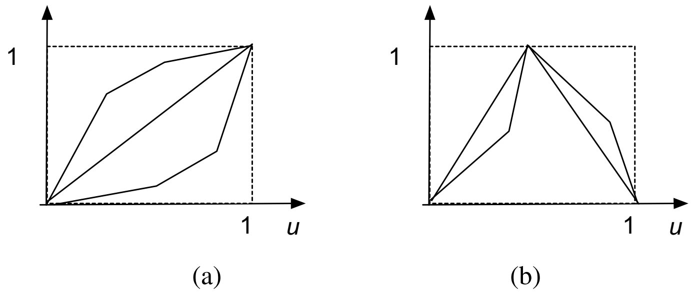


Fig. 2.14 Examples of functions e and h used in the functionals of energy measure and entropy measures of fuzziness

$$E(A) = \sum_{x \in \boldsymbol{X}} e[A(x)] \tag{2.20}$$

if card ($\boldsymbol{X}$) = n. In the case of the infinite space, the energy measure of fuzziness is expressed as the following integral (we assume it does exist)

$$E(A) = \int_{x \in \boldsymbol{X}} e[A(x)]dx \tag{2.21}$$

The mapping e: [0, 1] → [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 (Fig. 2.14).

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, the expressions (2.20) and (2.21) reduce to the cardinality of A,

$$\mathrm{E}(A) = \sum_{i=1}^{n} A(x_i) = \mathrm{card}(A) \tag{2.22}$$

The energy measure of fuzziness forms a convenient way of expressing a total mass of the fuzzy set. Since card(Ø) = 0 and card ($\boldsymbol{X}$) = n, the more a fuzzy set differ from the empty set, the larger its mass is. Indeed, rewriting (2.22) we obtain

$$E(A) = \sum_{i=1}^{n} A(x_i) = \sum_{i=1}^{n} |A(x_i) - \emptyset(x_i)| = d(A, \emptyset) = \mathrm{card}(A) \tag{2.23}$$

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 $e(u) = \sin\left(\frac{\pi}{2}u\right)$. Note that by choosing a certain form of the functional, we accentuate a varying contribution of different membership grades. For instance, depending upon 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.

Entropy measure of fuzziness of A, denoted by $H(A)$, is built upon the entropy functional (h) and comes in the form [xx]

$$\mathrm{H}(A) = \sum_{x \in \boldsymbol{X}} h(A(x)) \tag{2.24}$$

or in the continuous case of **X**

$$H(A) = \int\limits_{x \in \boldsymbol{X}} h(A(x))dx \tag{2.25}$$

where h: $[0, 1] \rightarrow [0, 1]$ is a functional such that (i) it is monotonically increasing in [0, ½] and monotonically decreasing in [½, 1] and (ii) comes with the boundary conditions $h(0) = h(1) = 0$ and $h(½) = 1$. This functional emphasizes membership degrees around ½; in particular the value of ½ is stressed to be the most *unclear* (causing the highest level of hesitation with its quantification by means of the proposed functional).

Example 4 Determine cardinality of the Gaussian membership function $A(x) = \exp(-x^2/2)$.

We calculate the integral of the membership function $\int_{\boldsymbol{R}} A(x)dx$. Note that the integral of the following expression (probability of normal density function) $\frac{1}{\sqrt{2\pi}} \int_R \exp\left(-x^2/L_2\right)$ is 1, thus the above integral is $\sqrt{2\pi}$.

Example 5 Determine the energy and entropy measures of fuzziness of $B = [1.0\ 0.9\ 0.6\ 0.3\ 0.2\ 0.0]$. Consider that $e(u) = u$ and $h(u)$ is piecewise linear such that $h(u) = 2u$ for u in [0, ½] ang $\mathrm{h}(u) = 2(1 - u)$ for u in [1/2, 1].

The energy measure of fuzziness is the sum of the membership grades,

$$E(B) = e(1.0) + e(0.9) + e(0.6) + e(0.3) + e(0.2) + e(0.0) = 3.0$$

The entropy measure of fuzziness is computed as follows

$$\begin{aligned} H(B) &= h(1.0) + h(0.9) + h(0.6) + h(0.3) + h(0.2) \\ &\quad + h(0.0) = 0.0 + 0.2 + 0.8 + 0.6 + 0.4 = 2.0 \end{aligned}$$

2.5 A Representation Theorem

Fuzzy sets offer an important conceptual and operational feature of information granules by endowing their formal models by gradual degrees of membership. We are interested in exploring relationships between fuzzy sets and sets. While sets come with the binary (yes-no) model of membership, it could be worth investigating whether they are indeed some special cases of fuzzy sets and if so, in which sense a set could be treated as a suitable approximation of some given fuzzy set. This could shed light on some related processing aspects. To gain a detailed insight into this matter, we recall here a concept of an α-cut and a family of α-cuts and show that they relate to fuzzy sets in an intuitive and transparent way [3]. Let us re-visit the semantics of α-cuts: an α-cut of A embraces all elements of the fuzzy set whose degrees of belongingness (membership) to this fuzzy set are at least equal to α. In this sense, by selecting a sufficiently high value of α, we identify (tag) elements of A that belongs to it to a significant extent and thus could be sought as those substantially representative of the concept conveyed by A. Those elements of $\boldsymbol{X}$ exhibiting lower values of the membership grades are suppressed so this allows us to selectively focus on the elements with the highest degrees of membership while dropping the others.

For α-cuts A_α the following properties hold

$$\begin{aligned}&\text{(i)}\quad A_0 = \boldsymbol{X}\\&\text{(ii)}\quad \text{If } \alpha \le \beta \in \quad \text{then } A_\alpha \supseteq A_\beta\end{aligned} \tag{2.26}$$

The first property states that if we allow for the zero value of α, then all elements of $\boldsymbol{X}$ are included in this α-cut (0-cut, to be more specific). The second property underlines the monotonic character of the construct: higher values of the threshold imply that more elements are accepted in the resulting α-cuts. In other words, we may say that the level sets (α-cuts) A_α form a nested family of sets indexed by some parameter (α). If we consider the limit value of α, that is $\alpha = 1$, the corresponding α-cut is nonempty if and only if A is a normal fuzzy set.

It is also worth to remember that α-cuts, in contrast to fuzzy sets, are sets. We showed how for some given fuzzy set, its α-cut could be formed. An interesting question arises as to the construction that could be realized when moving into the opposite direction. Could we “reconstruct” a fuzzy set on a basis of an infinite family of sets? The answer to this problem is offered in what is known as the representation theorem for fuzzy sets [1, 2]

Theorem Let $\{A_\alpha\}$ $\alpha \in [0, 1]$ be a family of sets defined in $\boldsymbol{X}$ such that they satisfy the following properties

(a) $A_0 = \boldsymbol{X}$
(b) If $\alpha \le \beta$ then $A_\alpha \supseteq A_\beta$
(c) For the sequence of threshold values $\alpha_1 \le \alpha_2 \le \cdots$ such that $\lim \alpha_n = \alpha$ we have $A_\alpha = \bigcap_{n=1}^{\infty} A_{\alpha_n}$

Then there exists a unique fuzzy set B defined in $\boldsymbol{X}$ such that $B_\alpha = A_\alpha$ for each $\alpha \in [0, 1]$.

In other words, the representation theorem states that any fuzzy set A can be uniquely represented by an infinite family of its α-cuts. The following reconstruction expression shows how the corresponding α-cuts contribute to the formation of the corresponding fuzzy set

$$A = \bigcup_{\alpha > 0} \alpha A_\alpha \tag{2.27}$$

that is

$$A(x) = \sup_{\alpha \in (0,1]} [\alpha A_\alpha(x)] \tag{2.28}$$

where A_α denotes the corresponding α-cut.

The essence of this construct is that any fuzzy set can be uniquely represented by the corresponding family of nested sets (viz. ordered by the inclusion relation). The illustration of the concept of the α-cut and a way in which the representation of the corresponding fuzzy set becomes realized becomes visualized in Fig. 2.15.

More descriptively, we may say that fuzzy sets can be reconstructed with the aid of a family of sets. Apparently, we need a family of sets (intervals, in particular) to capture the essence of a single fuzzy set. The reconstruction scheme illustrated in Fig. 2.15 is self-explanatory with this regard. In more descriptive terms, we may look at the expression offered by (2.27)–(2.28) as a way of decomposing A into a series of layers (indexed sets) being calibrated by the values of the associated levels of α.

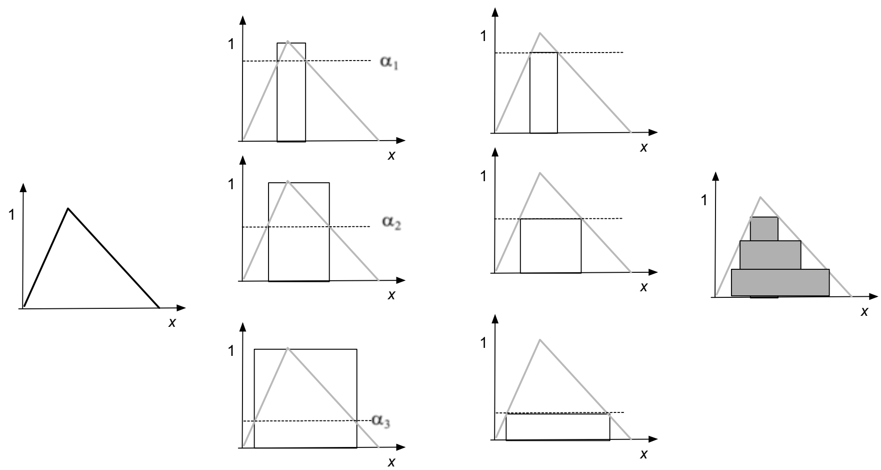


Fig. 2.15 Fuzzy set A, examples of some of its α-cuts and a representation of A through the corresponding family of sets (α-cuts)

For the finite universe of discourse, dim $(\boldsymbol{X}) = n$, we encounter a finite number of membership grades and subsequently a finite number of α-cuts. This finite family of α-cuts is then sufficient to fully represent or reconstruct the original fuzzy set. From the practical perspective, it is helpful in the development and implementation of algorithms with fuzzy sets in case there are well-established algorithms for intervals. The algorithm is realized for several α-cuts and then the partial results are combined to produce a solution in the form of a fuzzy set.

Example 6 Determine α-cut of the Gaussian membership function $A(x) = \exp(-x^2/\sigma^2)$.

We determine the bounds of the α-cut by solving the following equation $\exp(-x^2/\sigma^2) = \alpha$. This produces $-x^2/\sigma^2 = \ln\alpha$ so $A_\alpha = \left[-\sigma\sqrt{\ln\alpha}, \sigma\sqrt{\ln\alpha}\right]$.

Example 7 Determine α-cuts of the following fuzzy set $A = [0.3\ 0.2\ 1.0\ 0.5\ 0.1]$.

As there is a finite number of membership grades, there is a finite number of α-cuts. We have

$$A_1 = \left[0\ 0\ 1\ 0\ 0\right] A_{0.5} = \left[0\ 0\ 1\ 1\ 0\right], A_{0.3} = \left[1\ 0\ 1\ 1\ 0\right]$$
$$A_{0.2} = \left[1\ 1\ 1\ 1\ 0\right], A_{0.1} = \left[1\ 1\ 1\ 1\ 1\right].$$

Example 8 To illustrate the essence of α-cuts and the ensuing reconstruction, let us consider a fuzzy set with a finite number of membership grades, $A = \left[0.8\ 1.0\ 0.2\ 0.5\ 0.1\ 0.0\ 0.0\ 0.7\right]$. The corresponding α-cuts of A are equal to

$$\begin{aligned}
\alpha &= 1.0 A_{1.0} = \left[0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\right] \\
\alpha &= 0.8 A_{0.8} = \left[1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\right] \\
\alpha &= 0.7 A_{0.7} = \left[1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\right] \\
\alpha &= 0.5 A_{0.5} = \left[1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\right] \\
\alpha &= 0.2 A_{0.2} = \left[1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\right] \\
\alpha &= 0.1 A_{0.1} = \left[1\ 1\ 1\ 1\ 1\ 0\ 0\ 1\right]
\end{aligned}$$

We clearly see the layered character of the consecutive α-cuts indexed by the sequence of the increasing values of α. Because of the finite number of membership grades, the reconstruction realized in terms of (2.28) returns the original fuzzy set (which is possible given the finite space over which the original fuzzy set has been defined) $A(x) = \max\,(1.0A_{1.0}(x), 0.8A_{0.8}(x), 0.7A_{0.7}(x), 0.5A_{0.5}(x), 0.2A_{0.2}(x), 0.1A_{0.1}(x))$.

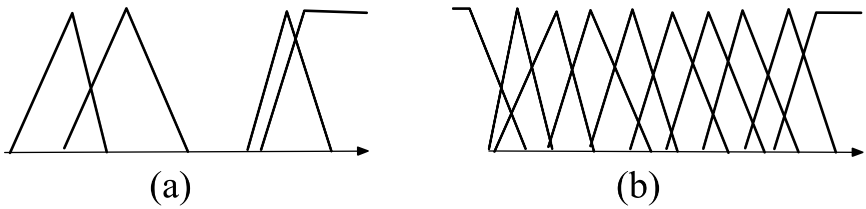


Fig. 2.16 Examples of families of fuzzy sets not forming a frame of cognition: **a** lack of coverage and high overlap, **b** high number of fuzzy sets

2.6 Families of Fuzzy Sets: A Frame of Cognition

A family of fuzzy sets (frame of cognition) $\boldsymbol{A} = \{A_1, A_2, \ldots, A_c\}$ can be sought a comprehensive description of a given variable defined over some space $\boldsymbol{X}$ and forming its abstract characterization. To make sure that $\boldsymbol{A}$ comes with a well-formed semantics, we put the following requirements

i. *coverage* We require that A_is "covers" the entire space, namely for any x in $\boldsymbol{X}$ there is at least one fuzzy set A_i so that $A_i(x) > 0$. This makes the fuzzy sets fully describing the entire space. The coverage requirement could be made stronger by imposing a condition $A_i(x) > \varepsilon$ where ε is a level of coverage.
ii. *limited number of fuzzy* sets As each fuzzy set comes with a clear semantics that is it maintains its meaning say *small*, *medium*, etc., this number has to be kept low. Typically, the maximal number of fuzzy sets in $\boldsymbol{A}$ should not exceed 9; one can refer here to the magic 7 ± 2 number.
iii. *unimodality* of membership functions of A_is. In this way, one retains the focus of each fuzzy set by identifying the corresponding regions of $\boldsymbol{X}$.
iv. *restricted overlap* between fuzzy sets We require that for any pair (i, j) $\max(\min(A_i(x), A_j(x)) > \delta$ where δ is a predefined tolerance level. If there were too high overlap, the corresponding fuzzy sets start losing their meaning by becoming not distinguishable enough.

Some illustrative examples are included in Fig. 2.16; here we show a way where there is a lack of satisfaction of any of the above requirements.

2.7 Linguistic Modifiers and Linguistic Quantifiers

The semantics of fuzzy sets is further augmented by a concept of linguistic quantifiers while linguistic approximation helps enhance to deliver semantics of any fuzzy set (typically resulting from computing using fuzzy sets).

Linguistic quantifiers referred to as linguistic hedges modify the semantics of the original fuzzy set by quantifying it by terms encountered in natural language such as *very, more or less*, etc. These terms modify the original membership function A

in the following way

$$\begin{aligned} very\ A(x) &= A(x)^2 \\ more\ or\ less\ A(x) &= A(x)^{0.5} \end{aligned} \tag{2.29}$$

In general, the modifiers are modeled by some power function (p) of A, A^p. They are also referred to as concentration operations ($p > 1$) and dilation operations ($p < 1$). For $p = 0$ $A^p(x) = 1$ and in this case we come up with a concept *unknown* (expressed as the entire space).

Notice that the modifiers have an intuitive interpretation: hedges like *very* produce a concentration effect; the result becomes more specific. The modification *very very A* with the membership function $A(x)^4$ amplifies the concentration effect. In contrast, *more or less A* makes the result less specific than A.

2.8 Linguistic Approximation

When fuzzy set B is a result of some computing with fuzzy sets and we would like to express it in terms of some fuzzy set coming from $\boldsymbol{A}$, we formulate the problem as linguistic approximation. In a nutshell, we approximate B by some A_i coming from $\boldsymbol{A}$ by involving linguistic modifiers coming from a set of modifiers $\boldsymbol{\tau} = \{\tau_1, \tau_2,\ldots, \tau_p\}$. In a formal way, we pose the problem as follows

$$\tau_j(x \text{ is } B) \approx A_i \tag{2.30}$$

Because of the finite number of fuzzy sets in $\boldsymbol{A}$ and the set of modifiers $\boldsymbol{\tau}$, the minimization of (2.30) gives rise to a combinatorial optimization problem: we select A_i and τ_j such that the distance between A_i and τ_j (x is B) is minimized

$$(i_0, j_0) = \arg\ \mathrm{Min}_{\mathrm{A},\boldsymbol{\tau}} \left\| \tau_j(x \text{ is } B) - A_i \right\| \tag{2.31}$$

where $\|\cdot\|$ is a distance between the fuzzy sets. For instance, the distance could be taken as the Hamming distance, say

$$\left\| \tau_j(x \text{ is } B) - A_i \right\| = \int\limits_X \left| \tau_j(B(x)) - A_i(x) \right| dx \tag{2.32}$$

2.9 Linguistic Variables

One can often deal with variables describing phenomena of physical or human systems assuming a finite, quite small number of descriptors.

We often describe observations about a phenomenon by characterizing its states which we naturally translate in terms of the idea of some variable. For instance, we can qualify the environment condition through the variable of temperature with values chosen in a range such as the interval $\boldsymbol{X} = [0, 40]$. The temperature can be described by terms such as cold, warm, and hot, all of them being represented in terms of some fuzzy sets. This brings an idea of a linguistic variable [5] which offer an interesting way to computing with words, viz. computing carried out at the level of information granules.

In general, linguistic variables may assume values consisting of words or sentences expressed in a certain language (Zadeh 6). Formally, a linguistic variable is characterized by a quintuple

$<X, T(X), \boldsymbol{X}, G, M>$ where the components are as follows

- X the name of the variable;
- $T(\boldsymbol{X})$ a term set of $\boldsymbol{X}$ whose elements are labels L of linguistic values of X;
- G a grammar that generates the names of X;
- M a semantic rule that assigns to each label $L \in T(\boldsymbol{X})$ some meaning expressed through the corresponding membership function

The following example serves as an illustration of the elements of the grammar.

Example 9 . Let us consider the linguistic variable of temperature. This linguistic variable is formalized by explicitly identifying all the components of the formal definition:

$X =$ temperature, $\boldsymbol{X} = [0, 40]$.
T(temperature) $= \{cold, warm, hot\}$
G is a grammar composed of production rules following which the names of X are formed; usually linguistic modifiers and logic operators are involved, say *very cold*, *not very hot*, etc.
Semantics associates with fuzzy sets defined for the labels, namely $M(cold)$, $M(warm)$ and
$M(hot)$. See Fig. 2.17.

The grammar is formulated by providing a collection of syntax rules. Here a standard way is realized in terms of the Backus-Naur form (BNF) using terminal and non-terminal symbols. The production list is used to verify correctness of a given expression.

An example production list is shown below

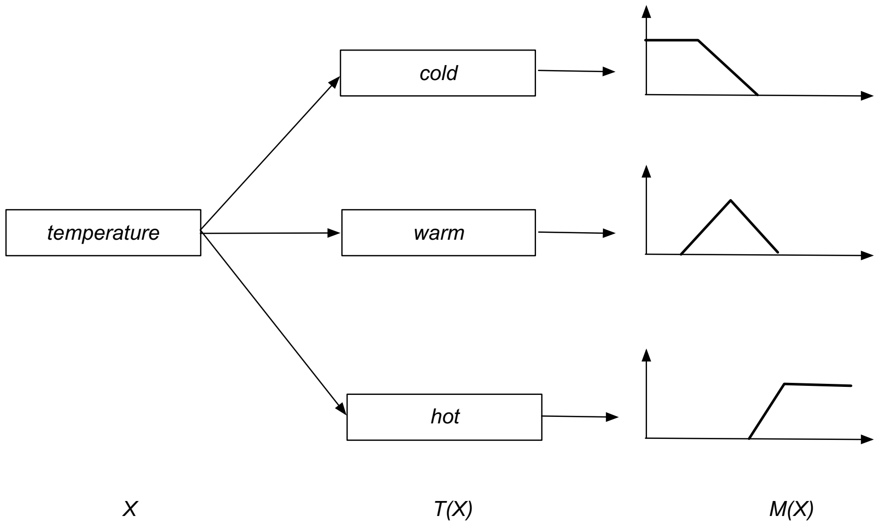


Fig. 2.17 An example of the linguistic variable of *temperature*

$$\begin{aligned}
&< \text{expression} >::=< \text{expression} >< \text{operator} >< \text{expression} >\\
&< \text{expression} >::=< \text{linguistic quantifier} >< \text{linguistic term} >\\
&\quad | < \text{negation} >< \text{expression} >\\
&< \text{linguistic quantifier} >::= \text{very}|\text{more or less}\\
&< \text{linguistic term} >::= \text{small}|\text{medium}|\text{high}\\
&< \text{operator} >::= \text{and}|\text{or}\\
&< \text{negation} >::= \text{not}
\end{aligned}\tag{2.33}$$

Using it, we build a derivation tree for the following expression *not very low and not high* as shown in Fig. 2.18. As moving from the expressing using the production rules, we reach the non-terminal symbol expression, the expression is syntactically correct.

The notion of the linguistic variable plays a major role in applications of fuzzy sets. In fuzzy logic and approximate reasoning truth values can be viewed as linguistic variables whose truth values form the term set as, for example, *true*, *very true*, *false*, *more or less true*, and the like.

2.10 Conclusions

Information granules are construct with a well-defined semantics, which help formalize a notion of abstraction. Different formal approaches emphasize this fundamental facet in different ways. Sets stress the notion of dichotomy. Fuzzy sets depart from the dichotomy emphasizing the idea of partial membership and in this way offer

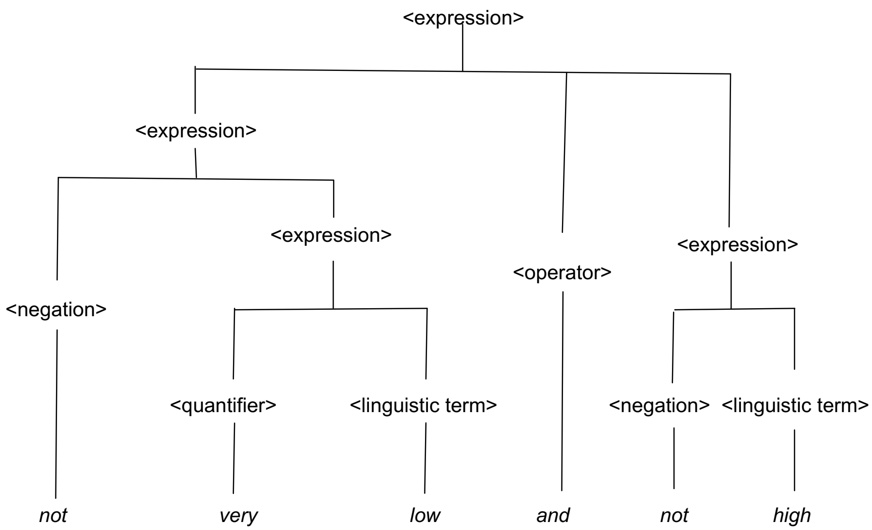


Fig. 2.18 Construction of derivation tree

a possibility to deal with concepts where the binary view at them of the underlying concept is not suitable or overly restrictive. Rough sets bring another conceptual and algorithmic perspective by admitting an inability of a full description of concepts in presence of a limited collection of information granules. Roughness comes as a manifestation of the limited descriptive capabilities of the vocabulary using which the description is realized. Shadowed sets can be sought as a concise, less numerically driven characterization of fuzzy sets (and induced by fuzzy sets themselves). One could also take another view at them as forming a bridge between fuzzy sets and rough sets. In essence, the shadows can be regarded as regions where uncertainty is been accumulated.

The choice of a suitable formal framework of information granules is problem-oriented and has to be done according to the requirements of the problem, ways of acquisition of information granules (both on a basis of domain knowledge and numerical evidence), and related processing and optimization mechanisms available in light of the assumed formalism of information granularity.

Problems

1. Discuss which of the concepts shown below you can describe with the use of fuzzy sets or sets:low mileage, capital city, solutions to quadratic equation, acceptable tax rate, high corporate profit, significant fluctuations of exchange rate
   Propose the corresponding membership or characteristic functions; define the universes of discourse on which these information granules are formed.
2. What is the sensitivity of the piecewise membership function
3. What is the support of the Gaussian membership function
4. Show if the parabolic membership function is convex

5. Determine the sensitivity measure of the S membership function.
6. Consider the fuzzy set discussed in Example 3. Now take an interval $[0, b]$. What is the value of b which makes the equality sp(A)=sp(B) satisfied?
7. Plot a Γ membership function

$$A(x) = \begin{cases} 0, & \text{if} x \leq a \\ 1 - \exp(-k(x-a)^2, & \text{if} x > a \end{cases}$$

where $k > 0$. Offer its interpretation.
8. What is the specificity of a triangular fuzzy set $A(x; a, m, b)$ and its modified version *very* (A) and *more or less* (A).

References

1. Dubois, D., Prade, H.: Outline of fuzzy set theory: An introduction. In: Gupta, M.M., Ragade, R.K., Yager, R.R. (eds.) Advances in Fuzzy Set Theory and Applications, pp. 27–39. North-Holland, Amsterdam (1979)
2. Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice-Hall, Upper Saddle River (1995)
3. Nguyen, H., Walker, E.: A First Course in Fuzzy Logic. CRC Press, Boca Raton, Chapman Hall (1999)
4. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 33–353 (1965)
5. Zadeh, L.A.: The concept of linguistic variables and its application to approximate reasoning I, II, III. Inf. Sci. **8**, 199–249, 301–357, 43–80 (1975)
6. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets Syst. **1**, 3 (1978)

Chapter 3
Interval Calculus

Abstract Interval analysis has emerged with the inception of digital computers and was mostly motivated by the models of computing studies therein. Computing is carried out for intervals (sets) implied by the finite number of bits used to represent any number on a (digital) computer. This interval nature of the arguments (variables) implies that the results are also intervals. This raises awareness about the granular nature of the results. Interval analysis is instrumental in the analysis of propagation of granularity residing within the original arguments (intervals). The intent of this chapter is to elaborate on the fundamentals of interval calculus. It will become apparent that the principles of interval computing will be helpful in the development of the algorithmic fabric of other formalisms of information granules. In particular, this concerns fuzzy sets given the representation theorem in which calculus of fuzzy sets is transformed to computing with a family of their alpha cuts.

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_3

3.1 Operations on Intervals

We briefly recall the fundamental notions of numeric intervals and their operations. Given are two intervals $A = [a, b]$ and $B = [c, d]$.

Equality

A and B are equal if their bounds are equal, $a = c$ and $b = d$.

Inclusion

A is included in B if $a \geq b$ and $c \leq d$

A degenerate interval $[a, a]$ is a single real number.

There are two categories of operations on intervals [1, 4, 5], namely set-theoretic and algebraic operations.

Set-theoretic operations

Assuming that the intervals are not disjoint (overlap, have some common non-empty intervals), the intersection and union are defined as follows

Intersection

$$\{z|z \in A \text{ and } z \in B\} = [\max(a, c)\min(b, d)] \tag{3.1}$$

Union

$$\{z|z \in A \text{ or } z \in B\} = [\min(a, c), \max(b, d)] \tag{3.2}$$

The illustration of these operations is displayed in Fig. 3.1.

Algebraic operations on intervals

The generic algebraic operations on intervals are quite intuitive. The results of addition, subtraction, multiplication, and division are expressed as follows

$$A + B = [a + c, b + d]$$

$$A - B = [a - d, b - c]$$

$$A^* B = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$

$$A/B = [a, b]^*[1/d, 1/c] (\text{it is assumed that } 0 \text{ is not included in the interval } [c, d]) \tag{3.3}$$

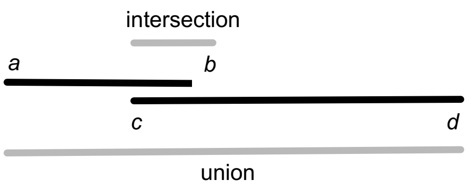


Fig. 3.1 Examples of set-theoretic operations on numeric intervals

Note that the result $A - B$ is determined by applying the multiplication formula for the product $[-1, -1]$*$[c, d]$. We have

$$\begin{aligned} A - B &= A + [-1 \ \ -1]^{*} B = [a, b] + [-1, -1]^{*}[c, d] \\ &= [a, b] + [-d, -c] = [a - d, b - c] \end{aligned}$$

All these formulas result from the fact that the above functions are continuous on a compact set; as result they take on the largest and the smallest value as well as the values in-between. The intervals of the obtained values are closed—in all these formulas one computes the largest and the smallest values. All the operations are conservative in the sense the length of the resulting interval is the highest.

Example 1 Carry out algebraic operations on the following intervals $A = [-1, 5]$ and $B = [3, 6]$ and determine the specificity of the obtained results considering the space $\boldsymbol{X}$ being a subset of real numbers $\boldsymbol{R}$, $\boldsymbol{X} = [-25, 25]$.

The specificity values of the arguments are $\mathrm{sp}(A) = 1 - 6/50 = 1 - 0.12 = 0.88$ and $\mathrm{sp}(B) = 1 - 3/50 = 0.0.94$.

Proceeding with formulas (3.3), we obtain

$$A + B = [2, 11], \mathrm{sp}(A + B) = 1 - 9/50 = 1 - 0.18 = 0.82$$

$$A - B = [-7, -2], \mathrm{sp}(A - B) = 1 - 9/50 = 1 - 0.18 = 0.82$$

$$\begin{aligned} A^{*}B &= [\min(\mathrm{ac, ad, bc, bd}), \max(\mathrm{ac, ad, bc, bd})] \\ &= [\min(-3, -6, 15, 30), \max(-3, -6, 15, 30] \\ &= [-6, 30], \mathrm{sp}\big(A^{*}B\big) = 1 - 36/50 = 1 - 0.72 = 0.28 \end{aligned}$$

$$\begin{aligned} A/B &= [-1, 5]^{*}[1/6, 1/3] = [-1/6, 5/3] \\ &= [-0.17, 1.67, \mathrm{sp}(A/B) = 1 - 1.83/50 = 0.96 \end{aligned}$$

It is interesting to note that the specificity of the results of the sum, subtraction, and multiplication is lower the specificity of the arguments.

Example 2 Consider a simple electric circuit consisting of resistors arranged in parallel and in series whose resistance is described by intervals, see Fig. 3.2. If the current I is granular in the form of the interval $I = [1, 3]$ Ampers

We determine the equivalent resistance of the circuits and the drop of voltage.

The resistance intervals are $R_1 = [1.0, 1.5]$ and $R_2 = [3.2, 3.5]$. When the resistors are connected in series, one has $R = R_1 + R_2 = [4.2, 5.0]$ and the drop of voltage U is $[4.2, 5.0]$*$[1, 3] = [4.2, 15.0]$ Volts.

For the resistors in the parallel configuration we have

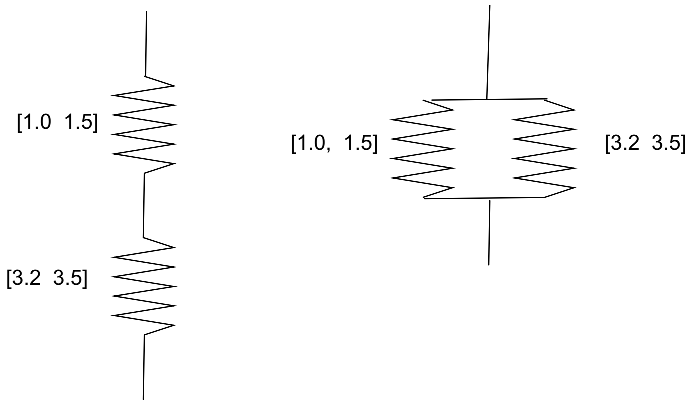


Fig. 3.2 Examples of electric circuits

$$
\begin{aligned}
R &= (R_1 + R_2)/\left(R_1^* R_2\right) = ([1\ 1.5] + [3.2, 3.5])/[3.2, 5.25] \\
&= [4.2, 5.0]^*[1/5.25, 1/3.2] \\
&= [4.2/5.25, 5.0/3.2] = [0.8, 1.6].
\end{aligned}
$$

and then the voltage is [1, 3]*[0.8, 1.6] = [0.8, 2.08] Volts.

3.2 Distance Between Intervals

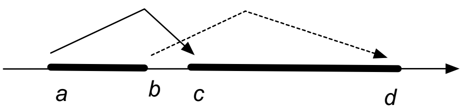
The concept of distance between two intervals A and B is more challenging than computing of distance between two real numbers or numeric vectors. This entails a number of alternatives.

Distance based on bounds of the intervals is computed in the form

$$d(A, B) = \max(|a - c|, |b - d|) \tag{3.4}$$

viz. in the calculations one considers the bounds of the intervals, see Fig. 3.3. One can easily show that the properties of distances are satisfied: $d(A, B) = d(B, A)$ (symmetry), $d(A, B)$ is nonnegative with $d(A, B) = 0$ (non-negativity) if and only if $A = B, d(A, B) d(A, C) + d(B, C)$ (triangle inequality). For real numbers the distance reduced to the Hamming one.

Fig. 3.3 Distance based on bounds of the intervals



Overlap measure takes into consideration a level of overlap between A and B and is expressed by taking a ratio of the length of the intersection $A \cap B$ over the length of the union $A \cup B$, namely

$$\text{overlap}(A, B) = \frac{\text{length}(A \cap B)}{\text{length}(A \cup B)} \tag{3.5}$$

If $A \cap B \neq \emptyset$ then $\text{overlap}(A, B) = \frac{|\min(b,d) - \max(a,c)|}{|\max(b,d) - \min(a,c)|}$ otherwise the overlap is equal to zero.

Hausdorff distance involves the maximum-minimum (max-min) calculations

$$D_H(A, B) = \max[\sup_{\boldsymbol{x} \in A} \inf_{\boldsymbol{y} \in B} d(\boldsymbol{x}, \boldsymbol{y}), \sup_{\boldsymbol{y} \in B} \inf_{\boldsymbol{x} \in A} d(\boldsymbol{x}, \boldsymbol{y})] = \max[h(A, B), h(B, A)] \tag{3.6}$$

where $h(A, B)$ and $h(B, A)$ are the one-sided Hausdorff distance from A to B and B to A, respectively. The detailed calculations are carried out as follows

$$\begin{aligned} h(A, B) &= \sup_{\boldsymbol{x} \in A} \inf_{\boldsymbol{y} \in B} d(\boldsymbol{x}, \boldsymbol{y}) \\ h(B, A) &= \sup_{\boldsymbol{y} \in B} \inf_{\boldsymbol{x} \in A} d(\boldsymbol{x}, \boldsymbol{y}) \end{aligned} \tag{3.7}$$

d is a distance computed for individual points (say, the Euclidean distance) and $\boldsymbol{a}$ and $\boldsymbol{b}$ stand for some points in A and B, respectively.

$h(A, B)$ finds the point in A that is the most distant from any point in B and identifies the distance from to its nearest neighbor in B [2, 3]. That is to say, one first ranks each point in A according to its distance to the nearest neighbor in B and then takes the largest one as the one-sided Hausdorff distance. The calculation process of $h(B, A)$ proceeds in a similar way. In essence, the overall distance quantifies the degree of mismatch between two sets by considering the spatial position of each individual point in these two sets.

Example 3 Determine the Hausdorff distance between A and B specified as follows

$$\begin{aligned} A &= \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\} = \{(1, 3)(5, 6)(-1, 0)\} \\ B &= \{\boldsymbol{y}_1, \boldsymbol{y}_2\} = \{(3, 4)(-2, 2)\} \end{aligned}$$

We compute

$$\begin{aligned} h(A, B) &= \max_{\boldsymbol{x} \in \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}}[\min_{\boldsymbol{y} \in \{\boldsymbol{y}_1, \boldsymbol{y}_2\}} d(\boldsymbol{x}, \boldsymbol{y})] \\ &= \max\{\min(d(\boldsymbol{x}_1, \boldsymbol{y}_1), d(\boldsymbol{x}_1, \boldsymbol{y}_2)), \\ &\qquad \min(d(\boldsymbol{x}_2, \boldsymbol{y}_1), d(\boldsymbol{x}_2, \boldsymbol{y}_2)), \min(d(\boldsymbol{x}_3, \boldsymbol{y}_1), d(\boldsymbol{x}_3, \boldsymbol{y}_2))\} \end{aligned}$$

$$h(B, A) = \max_{y \in \{y_1, y_2\}}[\min_{x \in \{x_1, x_2, x_3\}} d(\boldsymbol{x}, \boldsymbol{y})]$$
$$= \max\{\min\big(d\big(\boldsymbol{x}_1, \boldsymbol{y}_1\big), d\big(\boldsymbol{x}_2, \boldsymbol{y}_1\big), d\big(\boldsymbol{x}_3, \boldsymbol{y}_1\big)\big),$$
$$\min\big(d\big(\boldsymbol{x}_1, \boldsymbol{y}_2\big), d\big(\boldsymbol{x}_2, \boldsymbol{y}_2\big), d\big(\boldsymbol{x}_3, \boldsymbol{y}_2\big)\big)\}$$

If we take the distance between the points as a the Hamming distance, then $d(\boldsymbol{a}, \boldsymbol{b}) = |a_1 - b_1| + |a_2 - b_2]\boldsymbol{a} = (a_1, a_2), \boldsymbol{b} = (b_1, b_2)$.

3.3 Mapping of Interval Arguments

Given a certain function $f : \boldsymbol{R} \to \boldsymbol{R}$ and the input in the form of some interval A, the result is an interval $B = f(A)$. The question is how to determine B.

If f is monotonic (increasing or decreasing) the calculations are straightforward as illustrated in Fig. 3.4.

The result depends only on the bounds of A: if f is increasing, we obtain

$$B = f(A) = \left[\min_x f(x), \max_x f(x)\right] = [f(a), f(b)] \tag{3.8}$$

where the minimum (maximum) are taken for all xs belonging to A. The result is the interval

$$B = f(A) = [f(b), f(a)] \tag{3.9}$$

for the decreasing function f.

In general, the result of mapping is obtained through so-called extension principle expressed in the form

$$f(A)(y) = \sup_{x: f(x)=y} A(x) \tag{3.10}$$

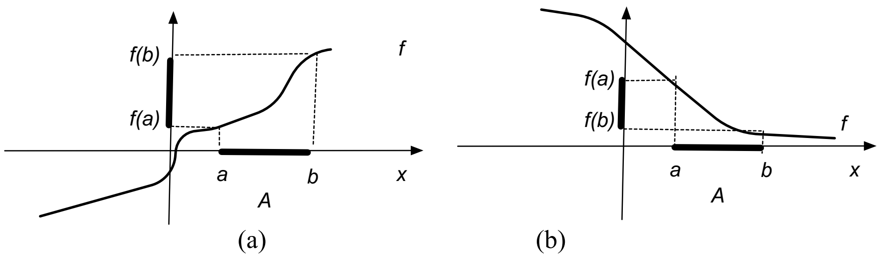


Fig. 3.4 Mapping of A when (**a**) f is increasing, (**b**) f is decreasing

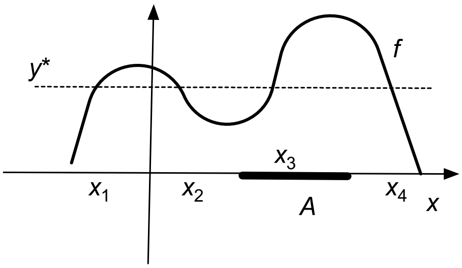


Fig. 3.5 Determination of $B(y)$ with the use of the extension principle

The principle produces an "optimistic" solution in the following sense: the supremum (maximum) operation is taken over all the solutions to the equation $y = f(x)$ for y given and taking the maximal value of the characteristic function of A. Say, x_1, x_2, x_3, and x_4 are solutions to $f(x) = y^*$. Then one considers the maximal value among $A(x_1) = 0$, $A(x_2) = 0$, $A(x_3) = 1$, and $A(x_4) = 0$. Refer to Fig. 3.5. Thus $B(y^*) = 1$.

The determination of the characteristic function of B through the extension principle is a nonlinear optimization problem with constraint specified by f

$$B(y) = \sup_x A(x)$$
$$\text{subject to}$$
$$f(x) = y \tag{3.11}$$

The generalization to a multivariable function f is straightforward. For a two variable function, compute $B = f(A, C)$, one has

$$B(y) = \sup_{x,z}[\min(A(x), C(z))]$$
$$\text{subject to}$$
$$y = f(x, z) \tag{3.12}$$

Example 4 The function is given as $f(x) = \sin(x)$ and $A = [\pi/4, 3\pi/4]$. Determine $B = f(A)$.

The solution to this optimization problem is illustrated in Fig. 3.6.

The bounds of B are determined by taking minimum and maximum of f over [π/4, 3π/4] and this entails $[\sin(\pi/4), \sin(3\pi/4)] = [\sqrt{2}/2, 1]$..

3.4 Relations and Relation Calculus

Relations are subsets of a Cartesian product of some spaces ($\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$) and describe some relationship between the elements of the coordinates of the product. For

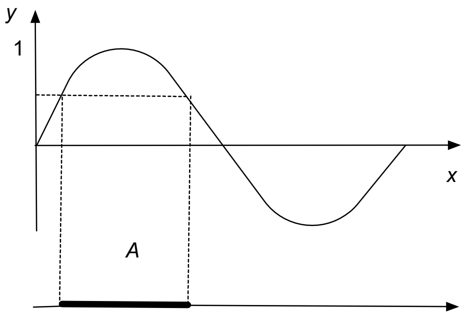


Fig. 3.6 Mapping A through a nonlinear function

instance if $\boldsymbol{X} = \{1, 2, \ldots 10\}$, the relation R *equal* defined in $\boldsymbol{X} \times \boldsymbol{X}$ comes with the characteristic function R

$R(x, y) = 1$ if $x = y$ and 0 otherwise. Examples of some other relations are *less than*, *larger than*, etc. A circle is a relation, $R = \left\{(x, y) | x^2 + y^2 = 4\right\}$ viz. a set of pairs (x, y) satisfying the relationship $x^2 + y^2 = 4$. Relations are commonly encountered in a variety of real-world situations. For instance, for $\boldsymbol{X}$ being a set of countries and Y being a set of currencies, the relation describes linkages country-currency.

In contrast to functions where we distinguish between input (independent) variables and output (dependent) variables, Relations are constructs that are direction-free; there is a profound distinction between input and output variables. Functions are special cases of relations. When R satisfies the condition that for any x in $\boldsymbol{X}$ there is only a single value of y in $\boldsymbol{Y}$ where $R(x, y) = 1$, this relation is a function from $\boldsymbol{X}$ to $\boldsymbol{Y}$.

The operations on relations are expressed in the same way as operations on sets. Given characteristic functions of R and G defined in $\boldsymbol{X} \times \boldsymbol{Y}$ one has

$$\begin{aligned} &\text{Union} \quad (R \cup G)(x, y) = \max(R(x, y), G(x, y)) \\ &\text{Intersection} \quad (R \cap G)(x, y) = \min(R(x, y), G(x, y)) \\ &\text{Complement} \quad \bar{R}(x, y) = 1 - R(x, y) \end{aligned} \tag{3.13}$$

As relations are defined in some Cartesian products of spaces, there are some operations whose essence is to modify their dimensionality

cylindric extension If A is defined in $\boldsymbol{X}$, its cylindric extension formed over the Cartesian product of $\boldsymbol{X}$ and $\boldsymbol{Y}$, $\text{cyl}_y(A)(x, y) = A(x)$ for any pair (x, y). Now cyl(A) and R are of the same form being the two relations defined over $\boldsymbol{X} \times \boldsymbol{Y}$.

Projection of relation is a process in which one reduces the dimensionality of R, see Fig. 3.7. If R is a relation in $\boldsymbol{X} \times \boldsymbol{Y}$, there are two projection operations

- projection on $\boldsymbol{X}$ $\quad \text{Proj}_{\boldsymbol{X}} R(x) = \sup_y R(x, y)$
- projection on $\boldsymbol{Y}$ $\quad \text{Proj}_{\boldsymbol{Y}} R(y) = \sup_x R(x, y)$

In both cases the reduction of dimensionality of R takes place.

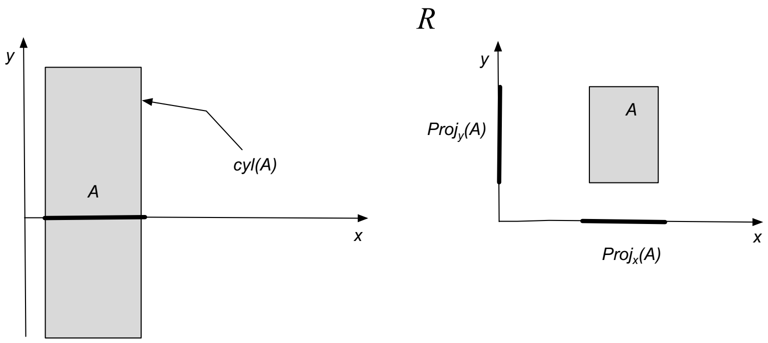


Fig. 3.7 Cylindric extension and projection operations

3.5 Composition Operations

There are several main composition operators on relations. For R defined in $\boldsymbol{X} \times \boldsymbol{Z}$ and G in $\boldsymbol{Z} \times \boldsymbol{Y}$ the result of composition gives rise to some relation T defined in $\boldsymbol{X} \times \boldsymbol{Y}$. Two composition operations are defined

sup-min composition $T = R \circ G$

$$T(x, y) = \sup_z[\min(R(x, z), G(z, y)] \tag{3.14}$$

inf-max composition $T = R\star G$

$$T(x, y) = \inf_z[\max(R(x, z), G(z, y)] \tag{3.15}$$

In particular, if one of the relations is a set, the sup-inf composition $B = A \circ R$, with A defined in $\boldsymbol{X}$ and R expressed over $\boldsymbol{X} \times \boldsymbol{Y}$, then the characteristic function of B reads as

sup-min composition

$$B(y) = \sup_x[\min(A(x), R(x, y)] \tag{3.16}$$

One can define the inf-max composition

$$B(y) = \inf_x[\max(A(x), R(x, y)] \tag{3.17}$$

Note that if A is the entire space, $A(x) = 1$ for all x in $\boldsymbol{X}$, then the above sup-min composition is the projection operation.

For finite spaces, relations are matrices and the composition operations are operations on matrices.

Example 5 Consider two relations $R = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$. Determine their max-min and min-max compositions.

We have

$$R \circ G = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$R{\star}G = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} {\star} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

In general, Cartesian products are multidimensional and can have a number of coordinates, say $\boldsymbol{X} \times \boldsymbol{Y} \times \boldsymbol{Z} \times \boldsymbol{W}$, etc. Then relations are tensors. Operations of cylindric extension and projection can involve different subsets of coordinates. If R is defined in $\boldsymbol{X}_1 \times \boldsymbol{X}_2 \times \cdots \times \boldsymbol{X}_p$, one defines its projection on a Cartesian product of spaces $\boldsymbol{X}_I$s, $\mathrm{Proj}_{XI}R$, where I is a subset of indexes $I = \{i_1, i_2, \ldots, i_r\}, r < p$ whose characteristic function reads as follows

$$\mathrm{Proj}_{XI} R(x_{i1}, x_{i2}, \ldots, x_{ir}) = \sup_{xj;\, j \in J} R\big(x_1, x_2, \ldots, x_p\big) \tag{3.18}$$

where $J = \{1, 2, \ldots, p\} - I$.

In the same way, we can talk about cylindric extensions as well as composition operations.

Example 6 Given are relations A defined in $\boldsymbol{X} \times \boldsymbol{Y}$, B defined in $\boldsymbol{X} \times \boldsymbol{Z}$ and C defined in $\boldsymbol{X} \times \boldsymbol{Y} \times \boldsymbol{Z}$. Carry out union and intersection of these relations.

First, the relations need to be unified in terms of their dimensionality. For instance, we make all of them defined in $\boldsymbol{X} \times \boldsymbol{Y} \times \boldsymbol{Z}$ by running cylindric extension so that all of them are defined in $\boldsymbol{X} \times \boldsymbol{Y} \times \boldsymbol{Z}$ and then the required operations are completed,

$$R = \mathrm{cyl}_z(A) \cap \mathrm{cyl}_y(B) \cap C$$
$$T = \mathrm{cyl}_z(A) \cup \mathrm{cyl}_y(B) \cup C$$

One can eliminate some coordinates of the Cartesian product through projections however this operation leads to information loss,

$$G = \mathrm{Proj}_y(A) \cap \mathrm{Proj}_z(B) \cap \mathrm{Proj}_{y,z} C$$
$$U = \mathrm{Proj}_y(A) \cup \mathrm{Proj}_z(B) \cup \mathrm{Proj}_{y,z} C$$

3.6 Conclusions

Intervals are examples of information granules. We have discussed the basic calculus of intervals as well identified different ways of determining their closeness (distance). As fuzzy sets arise as a generalization of sets (by augmenting a way in which a quantification of belongingness is completed), the calculus of fuzzy sets can help use the computing with sets (intervals) through the use of α cuts present in the representation theorem.

Problems

1. Complete the min-max composition of R and G

$$R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad G = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

2. Using R and G coming from Problem 1, and T

$$T = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

complete the following compositions $R \circ (G \star T)$, $(R \circ G) \star T$, $R \circ (G \star T)$, $R \star (G \circ T)$

3. Determine distance and overlap between $A = [-1, 7]$ and $B = [-5, -1]$.
4. An (artificial) neuron with two inputs x_1 and x_2 and a single output y is governed by the following function

$$y = f(w_0 + w_1 x_1 + w_2 x_2)$$

where $f(u) = 1/(1 + \exp(-u))$ and $w0$, w_1, and w_2 are the weights of the neuron.

(i) Determine the output y for $w_0 = 3$, $w_1 = [-3, 2]$, $w_2 = [1, 4]$ and $x_1 = 0.5$, $x_2 = -4.0$.
(ii) Repeat the calculations for the same weights as in(i) but with $x_1 = [0.2, 0.7]$, $x_2 = [-5.0, -1.0]$.

References

1. Alefeld, G., Herzberger, J.: Introduction to Interval Computations. Academic Press, New York (1983)
2. Attouch, H., Lucchetti, R., Wets, R.J.: The topology of the ρ-Hausdorff distance. Annalidi Matematica Puraed Applicata **160**, 303–320 (1991)
3. Conci, A., Kubrusly, C.: Distances between sets-a survey. Adv. Math. Sci. Appl. **26**, 1–18 (2017)
4. Moore, R.: Interval Analysis. Prentice Hall, Englewood Cliffs (1966)
5. Moore, R., Kearfott, R.B., Cloud, M.J.: Introduction to Interval Analysis. SIAM, Philadelphia (2009)

Chapter 4
Fuzzy Relations and Relational Computing

Abstract Relations represent and quantify associations between objects. They provide a fundamental vehicle to describe interactions and dependencies between variables, components, modules, etc. Fuzzy relations generalize the concept of relations in the same manner as fuzzy sets generalize the fundamental idea of sets [4, 7, 8]. Fuzzy relations are highly instrumental in problems of information retrieval, pattern classification, control and decision-making. In this chapter, we introduce the idea of fuzzy relations, present some illustrative examples, discuss the main properties of fuzzy relations and provide with some interpretation. Fuzzy relational equations describe relationships among objects; several solutions are posed along with their solutions.

4.1 Fuzzy Relations

Fuzzy relations generalize the concept of relations by admitting the notion of partial association (linkages) between elements of universes. Given two universes $\boldsymbol{X}$ and

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_4

$\boldsymbol{Y}$, a fuzzy relation R is any fuzzy subset of the Cartesian product of $\boldsymbol{X}$ and $\boldsymbol{Y}$ [8]. Equivalently, a fuzzy relation on $\boldsymbol{X} \times \boldsymbol{Y}$ is a mapping

$$R : X \times Y \rightarrow [0, 1] \tag{4.1}$$

The membership function of R for some pair (x, y) set to 1, $R(x, y) = 1$, states that the two elements x and y are fully related. On the other hand, $R(x, y) = 0$ means that these elements are unrelated while the values in-between 0 and 1, underline a partial association between x and y. For instance, if d_{fs}, d_{nf}, d_{ns}, d_{gf} are documents whose subjects involve fuzzy systems (*fs*), neural fuzzy systems (*nf*), neural systems (*ns*) and genetic fuzzy systems (*gf*). The description of topics is provided by keywords w_f, w_n and w_g, respectively. Then a relation R on $\boldsymbol{D} \times \boldsymbol{W}$, $\boldsymbol{D} = \{d_{fs}, d_{nf}, d_{ns}, d_{gf}\}$ and $\boldsymbol{W} = \{w_f, w_n, w_g\}$ can assume the matrix form with the following entries

$$R = \begin{bmatrix} 1 & 0 & 0.6 \\ 0.8 & 1 & 0 \\ 0 & 1 & 0 \\ 0.8 & 0 & 1 \end{bmatrix}$$

Since the universes are discrete, R can be represented as a 4×3 matrix (4 documents and 3 keywords) and entries, e.g., $R(d_{fs}, w_f) = 1$ means that the document content d_{fs} is fully compatible with the keyword $\mathrm{w_f}$ whereas $R(d_{fs}, w_n) = 0$ and $R(d_{fs}, w_g) = 0.6$ indicates that d_{fs} does not mention neural systems, but includes genetic systems as part of its content, Fig. 4.1.

As with relations, when $\boldsymbol{X}$ and $\boldsymbol{Y}$ are finite with card($\boldsymbol{X}$) $= n$ and card($\boldsymbol{Y}$) $= m$, then R can be can be arranged into a certain $n \times m$ matrix $R = [r_{ij}]$, with $r_{ij} \in [0, 1]$ being the corresponding degree of association between x_i and y_j.

Example 1 Discuss how a digital image can be represented formally.

A digital image can be represented as a fuzzy relation R where $R(x, y)$ is a level of brightness of pixel at coordinates (x, y). Typically, the range is 0–255 which is normalized to the unit interval.

Example 2 Discuss how to describe a three-dimensional phenomenon of solar radiation.

A point in the three-dimensional space (x, y, z) is a voxel whose value is a level of radiation assuming values in [0,1]. $R(x, y, z)$ is a value of the three-dimensional fuzzy relation R.

Fuzzy relations defined on some continuous spaces such as $\boldsymbol{R}^2$, say *much smaller than*, *approximately equal*, and *similar* could, for instance, be characterized by the following membership functions

$$x\, much\, smaller\, than\, y\ R(x, y) = \begin{cases} 1 - \exp(-|x - y|), \text{if} x \leq y \\ 0, \text{otherwise} \end{cases}. \tag{4.2}$$

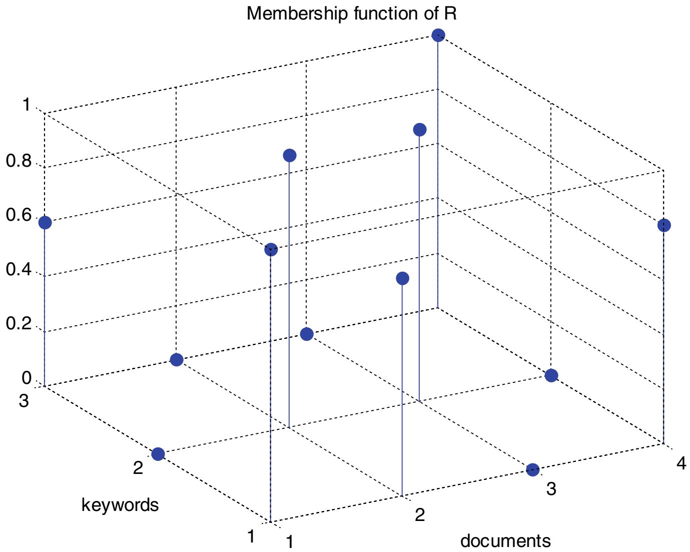


Fig. 4.1 Fuzzy relation of documents- keywords

$$x\ and\ y\ approximately\ equal\ R(x, y) = \exp(-|x - y|/\alpha), \alpha > 0 \tag{4.3}$$

$$x\ and\ y\ similar(\mathrm{x, y})R(x, y) = \begin{cases} \exp\left(-\frac{|x-y|}{\beta}\right) \text{ if } |x - y| \le 5 \\ 0 \text{ otherwise} \end{cases}, \beta > 0 \tag{4.4}$$

α and β are the parameters controlling the shape of the membership function. Figure 4.2 displays the membership function of the relation *x approximately equal to y* on $\boldsymbol{X} = \boldsymbol{Y} = [0, 4]$ assuming that $\alpha = 1$.

Complex concepts can be described by means of fuzzy relations rather than fuzzy sets as they usually involve several spaces.

Example 3 Describe the concept recession as a fuzzy relation.

Recession is described by several key factors (descriptors): Gross National Product (GNP), unemployment and Dow Jones (DJ). The pertinent fuzzy sets contributing to the concept of recession are fuzzy sets

A: *decreasing* GNP, *B- increasing* unemployment, and *C-decreasing* DJ.

The fuzzy relation of recession can be expressed as $R = A \times B \times C$.

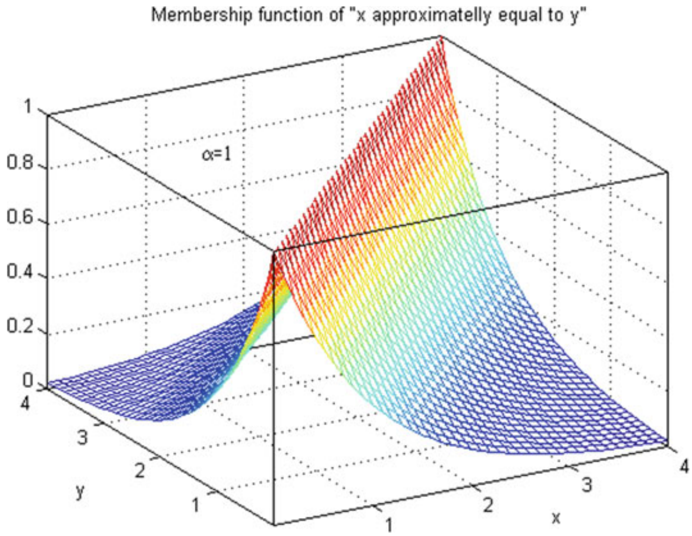


Fig. 4.2 Membership functions of the fuzzy relation *x approximately equal to y*

4.2 Properties of Fuzzy Relations

Fuzzy relations come with a number of properties, which capture the nature of the relationships conveyed by relations.

Domain and codomain of fuzzy relations

The domain, dom *R*, of a fuzzy relation *R* defined in $\boldsymbol{X} \times \boldsymbol{Y}$ is a fuzzy set whose membership function is equal to

$$\text{dom } R(x) = \sup_{y \in Y} R(x,\ y) \tag{4.5}$$

while its codomain, cod *R*, is a fuzzy set whose membership function is given as

$$\text{cod } R(y) = \sup_{x \in X} R(x, y) \tag{4.6}$$

Considering finite universes of discourse, domain and codomain can be viewed as the height of the rows and columns of the fuzzy relation matrix [8].

Representation of fuzzy relations

Similarly as in the case of fuzzy sets, fuzzy relations can be represented by their α-cuts, that is

$$R = \bigcup_{\alpha \in [0,1]} (\alpha\, R_\alpha) \tag{4.7}$$

or in terms of the membership function $R(x, y)$ of *R*

$$R(x, y) = \sup_{\alpha \in [0,1]}[\min(\alpha, R_\alpha(x, y))] \tag{4.8}$$

Equality of fuzzy relations

We say that two fuzzy relations P and Q defined in the same Cartesian product of spaces $\boldsymbol{X} \times \boldsymbol{Y}$ are equal if and only if their membership functions are identical, that is,

$$P(x, y) = Q(x, y) \quad \forall (x, y) \in \boldsymbol{X} \times \boldsymbol{Y} \tag{4.9}$$

Inclusion of fuzzy relations

A fuzzy relation P is included in Q, denoted by $P \subseteq Q$, if and only if

$$P(x, y) \le Q(x, y) \quad \forall (x, y) \in \mathbf{X} \times \mathbf{Y} \tag{4.10}$$

Similarly as it was presented in the case of relations, given n-fold Cartesian product of these universes we define the fuzzy relation in the form

$$R : X_1 \times X_2 \times \cdots \times X_n \to [0, 1] \tag{4.11}$$

If the spaces $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ are finite with $\text{card}(\boldsymbol{X}_1) = n_1 \ldots \text{card}(\boldsymbol{X}_n) = n_p$, then R can be expressed as an n-fold $(n_1 \times \cdots \times n_p)$ matrix $R = [r_{ij..k}]$ with $r_{ij..k} \in [0,1]$ being the degree of association assigned to the n-tuple $(x_i, x_j, \ldots, x_k) \in \boldsymbol{X}_1 \times \boldsymbol{X}_2 \times \cdots \times \boldsymbol{X}_n$. If $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ are infinite, then the membership function of R is a certain function of many variables. The concepts and definitions of equality and inclusion of fuzzy relations are immediately extended for relations defined in multidimensional spaces.

4.3 Operations on Fuzzy Relations

The basic operations on fuzzy relations, say union, intersection, and complement, conceptually follow the corresponding operations on fuzzy sets once fuzzy relations are fuzzy sets formed on multidimensional spaces. For illustrative purposes the definitions of union, intersection and complement below involve two-argument fuzzy relations. Without any loss of generality, we can focus on binary fuzzy relations P, Q, R defined in $\boldsymbol{X} \times \boldsymbol{Y}$. As in the case of fuzzy sets, all definitions are defined pointwise.

Union of fuzzy relations

The union R of two fuzzy relations P and Q defined in $\boldsymbol{X} \times \boldsymbol{Y}$, $R = P \cup Q$ is defined with the use of the following membership function

$$R(x, y) = P(x, y) s Q(x, y) \quad \forall (x, y) \in \boldsymbol{X} \times \boldsymbol{Y} \tag{4.12}$$

Intersection of fuzzy relations

The intersection R of fuzzy relations P and Q defined in $\boldsymbol{X} \times \boldsymbol{Y}$, $R = P \cap Q$ is defined in the following form,

$$R(x, y) = P(x, y)tQ(x, y) \quad \forall (x, y) \in \boldsymbol{X} \times \boldsymbol{Y} \tag{4.13}$$

Complement of fuzzy relations

The complement $\overline{R}$ of the fuzzy relation R is defined by the membership function

$$\overline{R}(x, y) = 1 - R(x, y)\forall (x, y) \in \boldsymbol{X} \times \boldsymbol{Y} \tag{4.14}$$

Transposition of fuzzy relations

Given a fuzzy relation R, its transpose, denoted by R^{T}, is a fuzzy relation on $\boldsymbol{Y} \times \boldsymbol{X}$ such that the following relationship holds

$$R^{\mathrm{T}}(y, x) = P(x, y) \quad \forall (x, y) \in \boldsymbol{X} \times \boldsymbol{Y} \tag{4.15}$$

If R is a relation defined in some finite space, then R^{T} is the transpose of the corresponding $n \times m$ matrix representation of R. Therefore the form of R^{T} is a $m \times n$ matrix whose columns are now the rows of R.

The following properties are direct consequences of the definitions provided above

$$\begin{aligned} \left(R^{\mathrm{T}}\right)^{\mathrm{T}} &= R \\ \overline{R}^{\mathrm{T}} &= \overline{R}^{\mathrm{T}} \end{aligned} \tag{4.16}$$

Cartesian product of fuzzy relations

Given fuzzy sets $A_1, A_2, \ldots, A_n$ defined in universes $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$, respectively, their Cartesian product $A_1 \times A_2 \times \cdots \times A_n$ is a fuzzy relation $\boldsymbol{R}$ on $\boldsymbol{X}_1 \times \boldsymbol{X}_2 \times \cdots \times \boldsymbol{X}_n$ with the following membership function

$$\begin{aligned} &R(x_1, x_2, \ldots, x_n) = \min\{A_1(x_1), A_2(x_2), \ldots, A_n(x_n)\} \\ &\forall x_1 \in \mathbf{X}_1, \forall x_2 \in \mathbf{X}_2, \ldots, \forall x_n \in \mathbf{X}_n \end{aligned} \tag{4.17}$$

In general, we can generalize the concept of this Cartesian product by using some t-norms

$$\begin{aligned} &\boldsymbol{R}(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{n}) = A_1(x_1)tA_2(x_2)t \ldots tA_n(x_n) \\ &\forall x_1 \in X_1, \forall x_2 \in X_2, \ldots, \forall x_n \in X_n \end{aligned} \tag{4.18}$$

In contrast to the concept of the Cartesian product, the idea of projection is to construct fuzzy relations on some subspaces of the original relation. Projection reduces the dimensionality of the original space over which the original fuzzy relation as defined.

Given R being a fuzzy relation defined in $\boldsymbol{X}_1 \times \boldsymbol{X}_2 \times \cdots \times \boldsymbol{X}_n$, its projection on $\boldsymbol{X} = \boldsymbol{X}_i \times \boldsymbol{X}_j \times \cdots \times \boldsymbol{X}_k$, where $I = \{i, j, \ldots, k\}$ is a subsequence of the set of indices $N = \{1, 2, \ldots, n\}$, is a fuzzy relation R_X with the membership function [8].

$$R_x\left(x_i, x_j, \ldots, x_k\right) = \mathrm{Proj}_x R(x_1, x_2, \ldots, x_n) = \sup_{x_t, x_u, \ldots x_v} R(x_1, x_2, \ldots, x_n) \tag{4.19}$$

where $J = \{t, u, \ldots, v\}$ is a subsequence of N such that $I \cup J = \mathrm{N}$ and $I \cap J = \emptyset$. In other words, J is the complement of I with respect to N. Notice that the above expression is computed for all values of $(x_1, x_2, \ldots, x_n) \in \boldsymbol{X}_i \times \boldsymbol{X}_j \times \cdots \times \boldsymbol{X}_k$.

For instance, Fig. 4.3 shows the projections R_X and R_Y of a certain Gaussian binary fuzzy relation R defined in $\boldsymbol{X} \times \boldsymbol{Y}$ with $\boldsymbol{X} = [0, 8]$ and $\boldsymbol{Y} = [0, 10]$, whose membership function is equal to $R(x, y) = \exp\{-\alpha[(x-4)^2 + (y-5)^2]\}$. In this case the projections are formed as

$$R_X(x) = \mathrm{Proj}_X R(x, y) = \sup_{y \in \boldsymbol{Y}} R(x, y) \tag{4.20}$$

$$R_y(y) = \mathrm{Proj}_x R(x, y) = \sup_{x \in \boldsymbol{Y}} R(x, y) \tag{4.21}$$

To find projections of the fuzzy relations defined in some finite spaces, the maximum operation replaces the sup operation occurring in the definition provided above. For example, for the fuzzy relation R: $\boldsymbol{X} \times \boldsymbol{Y} \rightarrow [0, 1]$ with $\boldsymbol{X} = \{1, 2, 3\}$ and $\boldsymbol{Y} = \{1, 2, 3, 4, 5\}$.

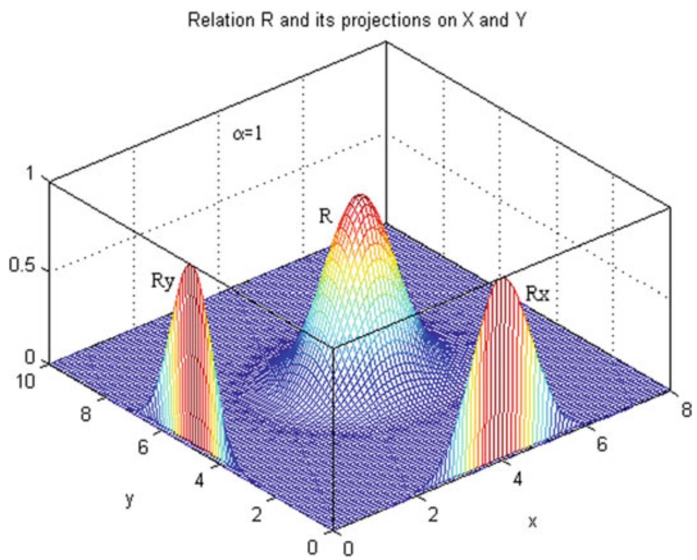


Fig. 4.3 Fuzzy relation R along with its projections on $\boldsymbol{X}$ and $\boldsymbol{Y}$

$$R(x, y) = \begin{bmatrix} 1.0 & 0.6 & 0.8 & 0.5 & 0.2 \\ 0.6 & 0.8 & 1.0 & 0.2 & 0.9 \\ 0.8 & 0.6 & 0.8 & 0.3 & 0.9 \end{bmatrix}$$

the three elements of the projection $R_{\mathbf{X}}$ are determined by taking the maximum computed for each of the three rows of R

$$\begin{aligned} R_{\mathbf{X}} &= [\max(1.0, 0.6, 0.8, 0.5, 0.2)\max(0.6, 0.8.1.0, 0.2, 0.9)\max(0.8, 0.6, 0.8, 0.3, 0.9)] \\ &= [1.01.00.09] \end{aligned}$$

Similarly, the five elements of R_{Y} are taken as the maximum among the entries of the five columns of R. Figure 4.4 shows R and its projections $R_{\mathbf{X}}$ and $R_{\mathbf{Y}}$.

$$R_X = \begin{bmatrix} 1.0 & 0.8 & 1.0 & 0.5 & 0.9 \end{bmatrix}$$

Note that domain and codomain of the fuzzy relation are examples of its projections.

We say that a fuzzy relation is decomposable if $R = \mathrm{Proj}_X R \times \mathrm{Proj}_Y R$.

Cylindrical extension

The cylindrical extension increases the number of coordinates of the Cartesian product over which the fuzzy relation is formed. In this sense, cylindrical extension is an operation that is complementary to the already discussed projection operation [8].

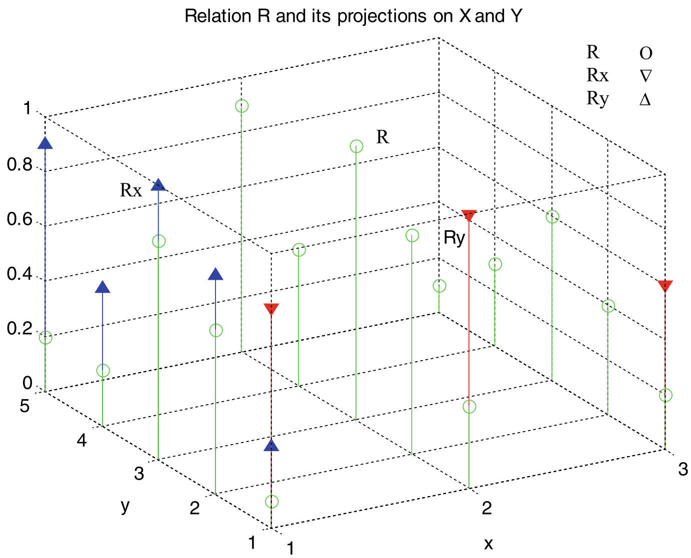


Fig. 4.4 Fuzzy relation R and its projections on $\boldsymbol{X}$ and $\boldsymbol{Y}$

The cylindrical extension on $\boldsymbol{X} \times \boldsymbol{Y}$ of a fuzzy set defined over $\boldsymbol{X}$ is a fuzzy relation cyl(A) whose membership function is equal to

$$\mathrm{cylA(x,y) = A(x)}, \ \forall \mathrm{x} \in \mathbf{X}, \ \forall \mathrm{y} \in \mathbf{Y}. \tag{4.22}$$

If the fuzzy relation is viewed as a two-dimensional matrix, the operation of cylindrical extension forms identical columns indexed by the successive values of $y \in \boldsymbol{Y}$. The main intent of cylindrical extensions is to achieve compatibility of spaces over which fuzzy sets and fuzzy relations are formed. For instance, let A be a fuzzy set of $\boldsymbol{X}$ and R a fuzzy relation on $\boldsymbol{X} \times \boldsymbol{Y}$. Suppose we attempt to compute union and intersection of A and R. Because the universes over which A and R are defined are different, we cannot carry out any set-based operations on A and R. The cylindrical extension of A, denoted by cyl(A) provides the compatibility required. Then the operations such as (cyl(A)) $\cup$ $\boldsymbol{R}$ and (cyl(A)) $\cap$ $\boldsymbol{R}$ make sense. The concept of cylindrical extension can be easily generalized to multidimensional cases.

Reconstruction of fuzzy relations

Projections do not retain complete information conveyed by the original fuzzy relation. This means that in general one cannot faithfully reconstruct a relation from its projections. In other words, projections $\mathrm{Proj}_{\boldsymbol{X}}R$ and $\mathrm{Proj}_{\boldsymbol{Y}}R$ of some fuzzy relation R, do not necessarily lead to the original fuzzy relation R. In general, the reconstruction of a relation via the Cartesian product of its projections is a relation that includes the original relation, that is

$$\mathrm{Proj}_x R \times \mathrm{Proj}_x R \supseteq R \tag{4.23}$$

If, however, the relationship above the equality $\mathrm{Proj}_{\boldsymbol{X}}R \times \mathrm{Proj}_{\boldsymbol{Y}}R = R$ holds, then we call the relation R noninteractive.

Binary fuzzy relations

A binary fuzzy relation R on $\boldsymbol{X} \times \boldsymbol{X}$ is defined as follows:

$$R : X \times X \to [0, 1] \tag{4.24}$$

There are several important features of binary fuzzy relations:

Reflexivity: $R(x, x) = 1 \ \forall \ x \in \boldsymbol{X}$, Fig. 4.5a. When $\boldsymbol{X}$ is finite $R \supseteq I$ where I is an identity matrix, $I(x, y) = 1$ if $x = y$ and $I(x, y) = 0$ otherwise. Reflexivity can be relaxed by admitting a concept of so-called ε-reflexivity, $\varepsilon \in [0, 1]$. This means $R(x, x) \geq \varepsilon$. When $R(x, x) = 0$ the fuzzy relation is irreflexive. A fuzzy relation is locally reflexive if, for any $x, y \in \boldsymbol{X}$, $\max\{R(x, y), R(y, x)\} \leq R(x, x)$.

Symmetry: $R(x, y) = R(y, x) \ \forall \ (x, y) \in \boldsymbol{X} \times \boldsymbol{X}$, Fig. 4.5b. For finite $\boldsymbol{X}$, the matrix representing R has entries distributed symmetrically along the main diagonal. Clearly, if R is symmetric, then $R^{\mathrm{T}} = R$.

Transitivity:. $\sup_{z \varepsilon X}[R(x, z) t R(z, y)] \leq R(x, y)$ for all x, y, z in $\boldsymbol{X}$. In particular, if this relationship holds for $t = \min$, then the relation is called sup-min transitive. Looking at the levels of associations

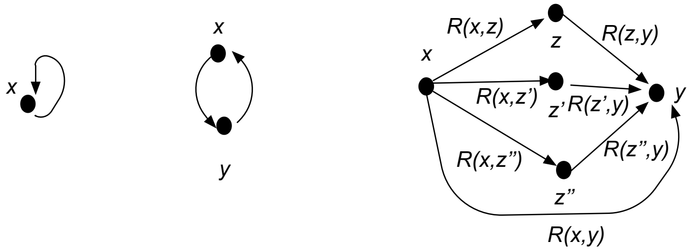


Fig. 4.5 Main characteristics of binary fuzzy relations; see the details in the text

$R(x, z)$ and $R(z, y)$ occurring between x, and z, and z and y, the property of transitivity reflects the maximal strength among all possible links arranged in series (such as ($R(x, z)$ and $R(z, y)$)) that does not exceed the strength of the direct link $R(x, z)$, Fig. 4.5c.

Transitive closure

Given a binary fuzzy relation in a finite universe $\boldsymbol{X}$, there exists a unique fuzzy relation $\overset{\leftrightarrow}{R}$ on **X**, called transitive closure of R, that contains R and itself is included in any transitive fuzzy relation on $\boldsymbol{X}$ that contains R. Therefore, if R is defined on a finite universe of cardinality n, the transitive closure is given by

$$\mathrm{trans}(R) = R \cup R^2 \cup \ldots \cup R^n \tag{4.25}$$

where, by definition,

$$R^2 = R^\circ R \quad R^p = R^\circ R^{p-1} \tag{4.26}$$

$$(R^\circ R)(x, y) = \max_z [R(x, z) t R(z, y)] \tag{4.27}$$

Notice that the composition $R \circ R$ can be computed similarly as encountered in matrix algebra by replacing the ordinary multiplication by some t-norm and the sum by the max operations.

If R is reflexive, then

$$I \subseteq R \subseteq R^2 \subseteq \ldots \subseteq R^{n-1} = \subseteq R^n \tag{4.28}$$

The transitive closure of the fuzzy relation R can be found by computing the successive k max-t products of R until $R^k = R^{k-1}$, a procedure whose complexity is $O(n^3 \log_2 n)$ in time and $O(n^2)$ in space.

4.4 Equivalence and Similarity Relations

Equivalence relations are relations that are reflexive, symmetric and transitive [8]. Suppose that one of the arguments of $R(x, y)$, x for example, has been fixed. Thus, all elements related to x constitute a set called as an equivalence class of R with respect to x, denoted by

$$\mathrm{A}_x = \{y \in \boldsymbol{Y} | R(x, y) = 1\} \tag{4.29}$$

The family of all equivalence classes of R denoted $\mathbf{X}/R$, forms a partition of $\boldsymbol{X}$. In other words, $\boldsymbol{X}/R$ is a family of pairwise disjoint nonempty subsets of $\boldsymbol{X}$ whose union is $\boldsymbol{X}$. Equivalence relations can be viewed as a generalization of the equality relations in the sense that members of an equivalence class can be considered equivalent to each other under the relation R.

Similarity relations are fuzzy relations that are reflexive, symmetric and transitive. Like any fuzzy relation, a similarity relation can be represented by a nested family of its α-cuts, R_α. Each α-cut constitutes a equivalence relation and forms a partition of $\boldsymbol{X}$. Therefore, each similarity relation is associated with a set $P(R)$ of partitions of $\boldsymbol{X}$,

$$P(R) = \{X/R_\alpha |\ \alpha \in [0, 1]\} \tag{4.30}$$

Partitions are nested in the sense that, if $\alpha > \beta$, then the partition $\mathbf{X}/R_\alpha$ is finer than the partition $\mathbf{X}/R_\beta$. For example, consider the relation defined on $\boldsymbol{X} = \{a, b, c, d, e\}$ in the following way

$$R = \begin{bmatrix} 1.0 & 0.8 & 0 & 0 & 0 \\ 0.8 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0.9 & 0.5 \\ 0 & 0 & 0.9 & 1.0 & 0.5 \\ 0 & 0 & 0.5 & 0.5 & 1.0 \end{bmatrix}$$

One can easily verify that R is a symmetric matrix, has values of 1 at its main diagonal, and is max-min transitive. Therefore R is a similarity relation. The levels of refinement of the similarity relation R can be represented in the form of partition tree in which each node corresponds to a fuzzy relation on $\boldsymbol{X}$ whose degrees of association between the elements are greater than or equal to the threshold value α. For instance, we have the following fuzzy relations for $\alpha = 0.5$, 0.8 and 0.9, respectively:

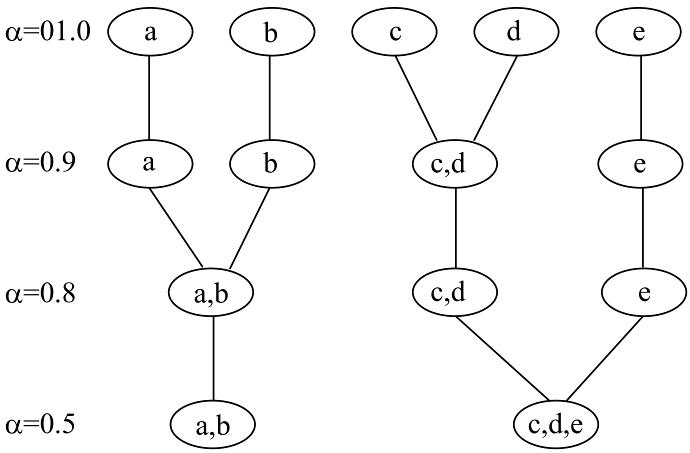


Fig. 4.6 Partition tree induced by binary fuzzy relation R

$$R_{0.5} = \begin{bmatrix} 1\ 1\ 0\ 0\ 0 \\ 1\ 1\ 0\ 0\ 0 \\ 0\ 0\ 1\ 1\ 1 \\ 0\ 0\ 1\ 1\ 1 \\ 0\ 0\ 1\ 1\ 1 \end{bmatrix}, \quad R_{0.8} = \begin{bmatrix} 1\ 1\ 0\ 0\ 0 \\ 1\ 1\ 0\ 0\ 0 \\ 0\ 0\ 1\ 1\ 0 \\ 0\ 0\ 1\ 1\ 0 \\ 0\ 0\ 0\ 0\ 1 \end{bmatrix}, \quad R_{0.9} = \begin{bmatrix} 1\ 0 & 0 & 0 & 0 \\ 0\ 1 & 0 & 0 & 0 \\ 0\ 0 & 1.0 & 1 & 0 \\ 0\ 0 & 1 & 1.0 & 0 \\ 0\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The union realized as the maximum operation and $\Lambda = \{0.5, 0.8, 0.9, 1.0\}$ is the level set of R. Also, notice that the higher the value of α, the finer the classes are, as shown in Fig. 4.6.

4.5 Fuzzy Relational Equations

The expression

$$Y = X^{\circ} R \tag{4.31}$$

with X and Y being fuzzy sets defined in $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively and R denoting a fuzzy relation defined in $\boldsymbol{X}\ \boldsymbol{Y}$ can be sought as a fuzzy relational equation. Two problems are formulated [1–3, 5, 6]:

- given X and Y, determine the fuzzy relation R
- given Y and R, determine X

The first problem is referred to as an estimation problem while the second one is an inverse problem. One regard X as a fuzzy set of causes and Y as a fuzzy set of effects. Hence one can regard the estimation problem as the one where we determine

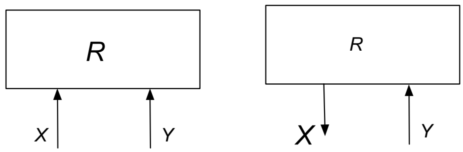


Fig. 4.7 Estimation and inverse problem in fuzzy relational equations

the relationships among causes and effects. The inverse problem is about determining possible causes when given is a collection of effects (Fig. 4.7).

We proceed with a systematic way of the development of the solutions to these two problems. The ϕ-operator (implication) plays a fundamental role here [3]

$$a\phi b = \sup\{c \in [0, 1] | atc \leq b\} \tag{4.32}$$

The ϕ operator satisfies the following properties [they could be easily proved based on the properties of (4.32)]

$$\begin{aligned} a\phi\max(b, c) &\geq \max(a\phi b, a\phi c) \\ at(a\phi b) &\leq b \\ a\phi(atb) &\geq b \end{aligned} \tag{4.33}$$

Based on this operator, we introduce the following two composition fuzzy set-fuzzy set and fuzzy relation- fuzzy set operators

$X\ \phi\ Y$ gives rise to the fuzzy relation with the membership function

$$(X\phi Y)(x, y) = X(x)\phi Y(y) \tag{4.34}$$

$R\ \phi\ Y$ forms a fuzzy set with the membership function

$$(R\phi Y)(x) = \inf_{y\in\mathrm{Y}}[R(x, y)\phi Y(y)] \tag{4.35}$$

The following lemmas are useful in further considerations

Lemma 1

$$R \subset X\phi(X^\circ R) \tag{4.36}$$

Lemma 2

$$X^\circ(X\phi Y) \subset Y \tag{4.37}$$

The proofs of these two lemmas are based directly on the properties of the ϕ operator and omitted here.

Define a family of fuzzy relations satisfying the equation $X^\circ R = Y$

$$\Re = \{R | X^\circ R = Y\} \tag{4.38}$$

The lemmas shown above give rise to the fundamental theorem.

Theorem 1 *If* $\Re \neq \emptyset$ *then the maximal element of the family* $\Re$ *denoted by* R *is expressed as*

$$\widehat{R} = X\phi Y \tag{4.39}$$

Proof In proving the theorem, we take advantage of the two lemmas shown above. Lemma 1 produces the following relationship $X \phi (X \circ R) = X \phi Y = \widehat{R} \supset R$. Due to the monotonicity of the sup-t composition, one has $X \circ \widehat{R} \supset X \circ R = Y$. Next, using Lemma 2, $X \circ (X \phi Y) \subset$ Y so X $\circ\widehat{R} \subset Y$. Because of the simultaneous satisfaction of inclusions $X \circ \widehat{R} \subset Y$ and $X \circ \widehat{R} \supset Y$, one concludes that $X \circ \widehat{R} = Y$.

The solution to the inverse problem is constructed with the aid of the two lemmas

Lemma 3

$$(R\phi Y)^{\circ} R \subset Y \tag{4.40}$$

Lemma 4

$$X \subset R\phi(X^{\circ} R) \tag{4.41}$$

As before denote by $\mathfrak{X}$ a family of fuzzy sets

$$\mathfrak{X} = \{X|X^{\circ} R = Y\} \tag{4.42}$$

Using the above lemmas we obtain the following theorem

Theorem 2 *If* $\mathfrak{X} \neq \emptyset$ *then the maxima element of* $\mathfrak{X}$ (*in terms of fuzzy set inclusion*) *is given in the form*

$$\begin{gathered}\widehat{X} = R\phi Y \\ \widehat{X}(x) = \min_y[R(x, y)\phi Y(y)]\end{gathered} \tag{4.43}$$

Several examples illustrate a way in which solutions are formed; here t-norm is specified as the minimum operator.

Example 4 Solve the following relational equation (estimation problem) $X = [1.0$ $0.5\ 0.2\ 0.7]$ $Y = [0.3\ 0.6\ 0.8]$. Assume that the t-norm is the minimum. Using (4.32) the ϕ operator reads as $a\phi b = \begin{cases} b if a > b \\ 1 if a \leq b \end{cases}$. The composition operator yields the following fuzzy relation

$$\widehat{R} = \begin{bmatrix} 0.3 & 0.6 & 0.8 \\ 0.3 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \\ 0.3 & 0.6 & 1.0 \end{bmatrix}$$

The theorems shown above deliver in an important analytical way the maximal solutions however the results are valid under the assumption that the solution set (either $\Re$ or $\mathfrak{X}$) are non-empty. If this assumption is not valid, then we can resort to optimization methods that produce some approximate solutions.

A generalization of the estimation problem can be formulated by considering a pair of input-output fuzzy sets X_k, Y_k, $k = 1, 2,\ldots, N$. This gives rise to a system of fuzzy relational equations

$X_k \circ R = Y_k$

Making use of Theorem 1 we solve each equation in the system producing $\widehat{R}_k = X_\mathrm{k}\ \phi\ Y_k$. Let us emphasize that the solution is the largest in the family of solutions to the k-th equation $\Re_k$. If these families produce a nonempty intersection, the solution to the system of equations is taken as an intersection of the largest solutions to each equation

$$\hat{R}(x, y) = \min_{k=1,2,\ldots,N} \widehat{R}_k(x,\ y) \tag{4.44}$$

The relational equations can be generalized by considering a number of input fuzzy sets

$$Y = \left(X_1 \times X_2 \times \ldots \times X_p\right)^{\circ} R \tag{4.45}$$

and a system of equations

$$Y_k = \left(X_{1k} \times X_{2k} \times \ldots \times X_{pk}\right)^{\circ} R \tag{4.46}$$

The solutions are obtained using the method developed so far. Depending upon the formulation of the problem and the architecture of the equations. Several examples shown below show the some illustrative situations.

Example 5 Determine the fuzzy sets X and Z defined in $\boldsymbol{X}$ and $\boldsymbol{Z}$, respectively for R and Y given the form. $(X \times Z)\circ R = Y$

The solution is obtained in the two-step process

by solving the inverse problem

$$\widehat{\mathrm{X} \times \mathrm{Z}} = \mathrm{R}\phi\mathrm{Y}$$

and then taking the projections of the result on the corresponding spaces $\widehat{X} = \mathrm{proj}_\mathrm{z}\widehat{\mathrm{X} \times \mathrm{Z}}$ and $\widehat{Y} = \mathrm{proj}_\mathrm{x}\widehat{\mathrm{X} \times \mathrm{Z}}$

Example 6 The input-output data are given in the following form of input-output pairs

input: X_1, output: Y_1
input: X_2, Z_2, output: Y_2
input: X_3, output: Y_3

Fuzzy sets X_1, X_2, and X_3 are defined in $\boldsymbol{X}$, Z_1 and Z_2 are defined in **Z**, and Y_1, Y_2, and Y_3 are given in $\boldsymbol{Y}$. Determine the fuzzy relation R.

We start by forming the cylindrical extensions of the fuzzy sets, $\mathrm{cyl}_{y,z}\,(X_1)$, $\mathrm{cyl}_y(X_2 \times Z_2)$, and $\mathrm{cyl}_{y,z}(X_3)$. In this way, we form a set of relational equations

$\mathrm{cyl}_{y,z}\,(X_1)\, R = Y_1$
$\mathrm{cyl}_y(X_2 \times Z_2)\, R = Y_2$
$\mathrm{cyl}_{y,z}(X_3)\, R = Y_3$

and the solution to the estimation problem is expressed as $[\mathrm{cyl}_{y,z}\,(X_1)\ \phi\ Y_1] \cap [\mathrm{cyl}_y(X_2 \times Z_2)\ \phi\ Y_2] \cap [\mathrm{cyl}_{y,z}(X_3)\ \phi\ Y_3]$

4.6 Conclusions

Fuzzy relations form a conceptual vehicle to capture associations (linkages) among elements. The properties of fuzzy relations quantify such linkages and deliver a complete picture of interactions among elements and spaces in which they are defined. Fuzzy relational equations are examples of processing relational dependencies and, as discussed later, help formulate a number of problems as estimation or inverse tasks.

Problems

1. Show an example fuzzy relation describing linkages among
   (i) movie goers $\{a, b, c, d\}$ and movies $\{ m1, m2, \ldots, m7\}$
   (ii) movie goers $\{a, b, c, d\}$, movies $\{ m1, m2, \ldots, m7\}$, and locations of the movie theaters $\{A, B, C, D, E\}$.

2. Show whether the following relation is decomposable

$$\begin{bmatrix} 0.7 & 0.2 & 0.6 \\ 0.2 & 0.0 & 0.9 \\ 0.7 & 0.4 & 1.0 \\ 0.0 & 0.1 & 0.5 \end{bmatrix}$$

3. Is the following relation T max-min transitive?

$$\begin{bmatrix} 1.0 & 0.5 & 0.7 \\ 0.5 & 1.0 & 0.4 \\ 0.7 & 0.4 & 1.0 \end{bmatrix}$$

4. Show that for a t-norm defined as the product, the ϕ-operator is expressed as follows

$$a\phi b = \begin{cases} b/a \text{ if} a > b \\ 1 \text{ if } a \le b \end{cases}.$$

5. Prove the relationships (4.33).
6. Given is the following similarity relation

$$\begin{bmatrix} 1.0 & 0.2 & 0.1 \\ 0.2 & 1.0 & 0.7 \\ 0.1 & 0.7 & 1.0 \end{bmatrix}$$

Show the resulting partition.
7. Given are pairs of input-output fuzzy sets (X_k, Y_k), $k = 1, 2, \ldots, n$ where X_k is defined in an n-element space, card $(\boldsymbol{X}) = n$. Consider a system of fuzzy relational equations $X_k \circ R = Y_k$. Show that if X_k has a 0-1 membership function in the form [0 …0 1 0 ..0] then the determined fuzzy relation comes in the form

$$R = \begin{bmatrix} Y_1^{\mathrm{T}} \\ Y_2^{\mathrm{T}} \\ \ldots \\ Y_k^{\mathrm{T}} \\ \ldots \\ Y_n^{\mathrm{T}} \end{bmatrix}$$

where the k-th row of R is the fuzzy set Y_k.

References

1. Bartl, E., Belohlavek, R.: Sup-t-norm and inf-residuum are a single type of relational equations. Int. J. Gen Syst. **40**, 599–609 (2011)
2. Bartl, E., Klir, G.J.: Fuzzy relational equations in general framework. Int. J. Gen. Syst. **43**, 1–18 (2014)
3. Di Nola, A., Sanchez, E., Pedrycz, W., Sessa, S.: Fuzzy Relation Equations and Their Applications to Knowledge Engineering. Kluwer, Dordrecht (1989)
4. Grabisch, M., Marichal, J.-L., Mesiar, R., Pap, E.: Aggregation Functions. Cambridge University Press, Cambridge (2009)
5. Sanchez, E.: Resolution of composite fuzzy relation equations. Inf. Control **34**, 38–48 (1976)
6. Shieh, B.-S.: Infinite fuzzy relation equations with continuous t-norms. Inf. Sci. **178**, 1961–1967 (2008)
7. Zadeh, L.A., Aliev, R.A.: Fuzzy Logic Theory and Applications. World Scientific, New Jersey (2019)
8. Zadeh, L.: Similarity relations and fuzzy ordering. Inf. Sci. **3**, 177–200 (1971)

Chapter 5
Fuzzy Set Transformation and Fuzzy Arithmetic

Abstract The chapter elaborates on the ways of transforming fuzzy sets through functions: one of the fundamental concepts of processing fuzzy sets. The extension principle is discussed in detail. Two key ways of mapping fuzzy sets are discussed: the one based on the representation theorem (which directly links to the mapping realized in interval analysis) and another one based on the use of the extension principle). The detailed formulas are developed for the arithmetic of fuzzy numbers.

5.1 The Extension Principle

Given a certain function f: $\boldsymbol{R} \rightarrow \boldsymbol{R}$ and its argument coming in the form of some fuzzy set A, the result of the mapping is a fuzzy set $B = f(A)$. The question is how to determine the membership function of B. The extension principle discussed for interval computing is extended to fuzzy sets (Zadeh [5]; Jardon et al. [2]) by replacing the characteristic function by the membership function

$$f(A) = \sup_{x f(x)=y} A(x) \tag{5.1}$$

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_5

As stated earlier, the principle produces an "optimistic" solution in the sense that one determines the highest value of the membership grades of $A(x_k)$ where x_ks are the solutions to the equation $f(x_k) = y$ for some fixed y. If there is no solution for this y, the solution set is empty and $A(y) = 0$. In this way, we can write down the extension principle as follows

$$f(A)(y) = \begin{cases} \sup_{xf(x)=y} A(x), & \text{if } \{x|f(x) = y\} \neq \varnothing \\ 0, & \text{if } \{x|f(x) = y\} = \varnothing \end{cases} \tag{5.2}$$

The determination of the membership function B completed through the extension principle is in general a nonlinear optimization problem with the constraint specified by f

$$\begin{aligned} &B(y) = \sup_x A(x) \\ &\text{Subject to} \\ &f(x) = y \end{aligned} \tag{5.3}$$

The generalization to a multivariable function f is straightforward. For n-variable function, we compute $Z = f(A_1, A_2,\ldots, A_n)$ in the form

$$\begin{aligned} &Z(y) = \sup_{x_1,x_2\ldots x_n}[\min(A_1(x_1), A_2(x_2), \ldots, A_n(x_n))] \\ &\text{Subject to the constraint} \\ &y = f(x_1, x_2, \ldots, x_n) \end{aligned} \tag{5.4}$$

The augmentation of the extension principle comes with the use of any t-norm instead of the minimum operation. In this way we obtain

$$\begin{aligned} &Z(y) = Z(y) = \sup_{x_1,x_2\ldots x_n}[A_1(x_1)tA_2(x_2)t\ldots tA_n(x_n)] \\ &\text{Subject to the constraint} \\ &y = f(x_1, x_2, \ldots, x_n) \end{aligned} \tag{5.5}$$

Example 1 Consider a problem of determining a distance traveled for *about* 3 h at the speed of *about* 100 km/h. The travel time and the speed are described by fuzzy sets T and V described by the corresponding membership function. The distance is computed in an obvious way as $d = vt$. Using the extension principle, we obtain $D = VT$ where the membership of D is computed as follows

$$\begin{aligned} D(d) &= \sup_{v,t:d=vt}[\min(V(v), T(t))] \\ &= \sup_{v}[\min(V(v), T(d/v))] \end{aligned} \tag{5.6}$$

As the function vt is invertible with respect to both arguments, the constraint can be dropped in the maximization problem.

Example 2 There are three operations carried out on assembly line in a series whose completion times are described by fuzzy sets T_1, T_2, and T_3. Determine the overall completion time of the entire process.

The total time is the sum of the times of the successive operations, $T = T_1 + T_2 + T_3$. Making use of the extension principle one has $T(t) = \sup_{t=t_1+t_2+t_3}(\min(T_1(t_1), T_2(t_2), T_3(t_3)))$. By noting the invertibility of the constraint, we can reformulate the extension principle where the constraint involves less variables, say $T(t) = \sup_{t=t_1+t_2}(\min(T_1(t_1), T_2(t_2), T_3(t - t_1 - t_2)))$.

The extension principle forms a cornerstone of algebra of fuzzy numbers.

5.2 Fuzzy Numbers and Their Arithmetic

Fuzzy numbers are fuzzy sets defined in the space of real numbers $\boldsymbol{R}$ which exhibit the property of normality and unimodality (Dubois and Prade [1]). Given this, any fuzzy number A can be decomposed into its increasing and decreasing portion, which can be written down as (f, g) with f and g being the increasing and decreasing parts of the membership function. More specifically, $A(x) = (f(x), g(x))$. The normality entails that there is some element of A belonging to the fuzzy set to the highest degree. They are used as commonly encountered models of concepts that capture essential aspects of information granularity. Adding and multiplying fuzzy numbers are often present in system modeling. We discuss the algebraic operations on fuzzy numbers. The computing is realized in two different ways (Pedrycz and Gomide [3]; Ross [4]: (i) by involving the representation theorem, and (ii) by using the extension principle.

5.2.1 Computing with Fuzzy Numbers: The Representation Theorem Approach

The computing process is realized by determining α-cuts of the arguments: use the interval calculus to obtain the corresponding α-cuts of the result (depending upon the formulas of interval calculus) and reconstruct the fuzzy set based on the calculated α-cuts. For the algebraic operations we have

$$(A \boxplus B)(x) = \sup_{\alpha \in [0,1]} (A \boxplus B)_\alpha(x) \tag{5.7}$$

where $\boxplus$ stands for the operation of addition, subtraction, multiplication, and division.

We determine the α-cut of the fuzzy sets, namely $a_\alpha = f^{-1}(\alpha)$ and $b_\alpha = g^{-1}(\alpha)$, Fig. 5.1, where f and g are the increasing and decreasing functions corresponding to the parts of the membership function of A.

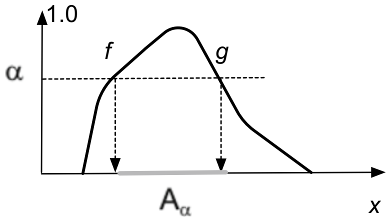


Fig. 5.1 Determining the α-cut of the membership function (*f, g*)

Recall that the corresponding formulas of interval computing (Chap. 3) entail the following formulas

$$\begin{aligned}
&A + B = [a + c, b + d] \\
&A - B = [a - d, b - c] \\
&A * B = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)] \\
&A/B = [a, b] * [1/d, 1/c] = [\min(a/d, a/c, b/d, b/c), \max(a/d, a/c, b/d, b/c)] \\
&\text{(it is assumed that 0 is not included in the interval}[c, d])
\end{aligned} \tag{5.8}$$

By running computations for successive values of α, the fuzzy set of result is easily reconstructed. From a practical perspective, we usually consider a finite number of threshold values α.

Example 3 We complete lgebraic operations on triangular fuzzy numbers $A(x;\, 2, 3, 6)$ and $B(x;\, 0, 1, 2)$.

We start with the determination of the α-cuts of A and B as illustrated in Fig. 5.2. We consider a general situation. In virtue of the form of the membership functions, one has the following α-cuts.

$$\begin{aligned}
A_\alpha &= [(m - a)\alpha + a,\ (m - b)\alpha + b] \\
B_\alpha &= [(n - c)\alpha + c,\ (n - d)\alpha + d]
\end{aligned} \tag{5.9}$$

For this example, in these expressions, we $a = 0$, $m = 1$, $b = 2$, and $c = 2$, $n = 3$, $d = 6$. Using the above formulas of interval calculus, we obtain

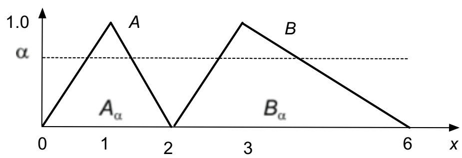


Fig. 5.2 α-cuts of fuzzy sets $A(x;\, 0, 1, 2)$ and $B(x;\, 2,3, 6)$

$$
\begin{aligned}
&(A+B)_\alpha = [(m-a)\alpha + a + (n-c)\alpha + c,\, (m-b)\alpha + b + (n-d)\alpha + d] \\
&(A-B)_\alpha = [(m-a)\alpha + a - (n-d)\alpha - d,\, (m-b)\alpha + b - (n-c)\alpha - c] \\
&(AB)_\alpha = [\min((m-a)\alpha + a,\, (m-b)\alpha + b,\, (n-c)\alpha + c,\, (n-d)\alpha + d), \\
&\max((m-a)\alpha + a,\, (m-b)\alpha + b,\, (n-c)\alpha + c,\, (n-d)\alpha + d)], \\
&(A/B)_\alpha = [((m-a)\alpha + a)/((n-d)\alpha + d),\;\; ((m-b)\alpha + b)/((n-c)\alpha + c)]
\end{aligned}
\tag{5.10}
$$

For the particular numeric values of the parameters, one obtains the detailed results

$$
\begin{aligned}
&(A+B)_\alpha = [2\alpha + 2, \alpha + 7] \\
&(A-B)_\alpha = [4\alpha - 6, \alpha - 1] \\
&(AB)_\alpha = [\alpha(\alpha + 2), -\alpha + 6] \\
&(A/B)_\alpha = [\alpha/(6-\alpha), 1/(\alpha + 2)]
\end{aligned}
$$

It is a noticeable observation that the result of addition and subtraction is a fuzzy set with a triangular membership function (the location of the bounds is a linear function of α) whereas for the product and division the resulting relationship is nonlinear (quadratic function).

Example 4 Determine a product of the fuzzy set A and some constant k.

Recall that A is described by increasing and decreasing parts of its membership function f and g. Using some α-cut one has the interval $[f^{-1}(\alpha), g^{-1}(\alpha)]$. If $k > 0$ we have

$$(kA)_a = \left[kf^{-1}(\alpha), kg^{-1}(\alpha)\right]. \text{ For } k < 0 \text{ we obtain } \left[kg^{-1}(\alpha), kf^{-1}(\alpha)\right].$$

By combining these two results

$$(kA)(x) = \begin{cases} (kf(x), kg(x)), \text{ if } k > 0 \\ (kg(x), kf(x)), \text{ if } k < 0 \end{cases} \tag{5.11}$$

We use the same approach to find the result of min(A, B) where A and B are defined with the corresponding increasing and decreasing functions $A(f_1, g_1)$ and $B(f_2, g_2)$. The α-cuts of A and B are determined as $A_\alpha = [f_1^{-1}(\alpha), g_1^{-1}(\alpha)]$ and $B_\alpha = [f_2^{-1}(\alpha), g_2^{-1}(\alpha)]$. Then the α-cut of the result is

$$C = \left[\min\left(f_1^{-1}(\alpha), f_2^{-1}(\alpha)\right), \min\left(g_1^{-1}(\alpha), g_2^{-1}(\alpha)\right)\right] \tag{5.12}$$

Example 5 Two fuzzy sets A and B with triangular membership functions are shown in Fig. 5.3. Determine the minimum of A and B.

Using formula (5.12) for the linear membership functions (note that the thresholds α^* and α^{**} are shown in Fig. 5.4), we obtain the membership function that is piecewise linear.

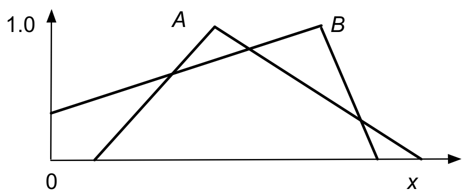


Fig. 5.3 Triangular fuzzy sets *A* and *B*

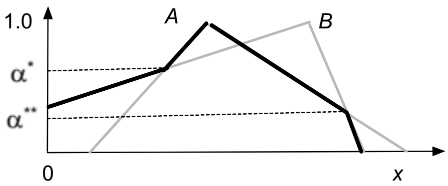


Fig. 5.4 Detailed computing of *C;* α* and α** are the arguments of *x* for which the corresponding membership values are equal, say $f_1 = f_2$ and $g_1 = g_2$

Example 6 Determine the minimum of fuzzy sets *A* and *B* illustrated in Fig. 5.5.

Using the same approach as before the resulting fuzzy set is *A*, which is an intuitively appealing result.

Example 7 Complete addition of fuzzy set *A* and interval *B* shown in Fig. 5.6.

Using α-cuts we obtain a trapezoidal fuzzy set illustrated in Fig. 5.7.

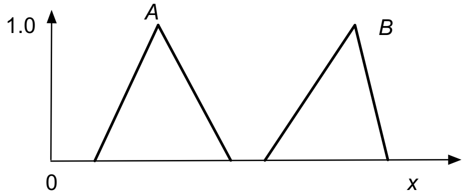


Fig. 5.5 Computing min(*A*, *B*)

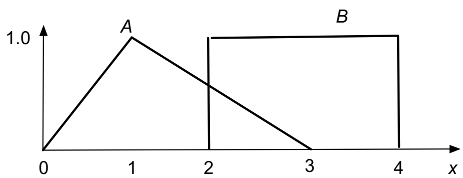


Fig. 5.6 Addition of *A* and *B*

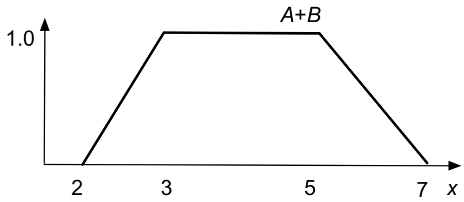


Fig. 5.7 Trapezoidal fuzzy set as a result of addition

5.2.2 *Computing with Fuzzy Numbers: The Extension Principle Approach*

The arithmetic operations are realized with the use of the extension principle. In light of the principle we have

$$C(z) = \sup_{x,y:z=f(x,y)} [\min(A(x), B(y))] \tag{5.13}$$

where the function f is the addition, subtraction, multiplication, and division. Note that in all these four situations, f is invertible so that f^{-1} does exist. For instance, from $z = a + b$, one obtains $b = z - a$, etc. As A and B are normal, there are arguments a and b such that $A(a) = 1$ and $B(b) = 1$. the result C is a normal fuzzy set,

$$C(f(a, b)) = \min(A(a), B(b)) = 1 \tag{5.14}$$

In the following discussion we concentrate on the arithmetic operations completed for triangular fuzzy numbers. Consider two fuzzy numbers $A(x; a, m, b)$ and $B(x; c, n, d)$ with the triples of parameters describing the lower bound, modal values, and the upper bound of the membership functions.

$$A(x;\, a, m, b) = \begin{cases} \frac{x-a}{m-a}, & \text{if } x \in [a, m] \\ \frac{b-x}{b-m}, & \text{if } x \in [m, b] \\ 0, & \text{otherwise} \end{cases}$$

$$B(x;\, c, n, d) = \begin{cases} \frac{x-c}{n-c}, & \text{if } x \in [c, n] \\ \frac{d-x}{d-n}, & \text{if } x \in [n, d] \\ 0, & \text{otherwise} \end{cases} \tag{5.15}$$

In the calculations, as noted before, we concentrate on (i) determining the modal value of the resulting membership function, (ii) working with the increasing parts of the membership function, and (iii) working with the decreasing parts of the membership functions.

As shown in Fig. 5.8, for the increasing portions of the membership functions, we obtain

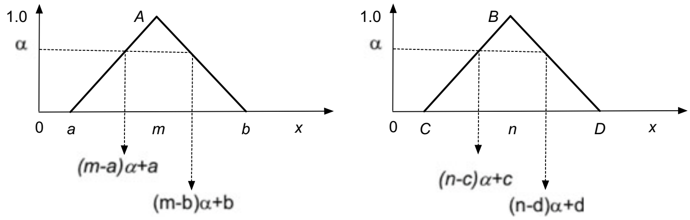


Fig. 5.8 Arguments of the linear membership functions obtained for specified value of α

$$
\begin{aligned}
x &= (m-a)a+a \\
y &= (n-c)a+c
\end{aligned}
\tag{5.16}
$$

The expressions for the decreasing parts are given in the form

$$
\begin{aligned}
x &= (m-b)a+b \\
y &= (n-d)a+d
\end{aligned}
\tag{5.17}
$$

Before proceeding with the details, we note that in case of multiplication and division, the procedure described above applies to so-called positive triangular fuzzy numbers viz. those where the lower bounds a and c are nonnegative.

5.2.3 Addition

The modal value of the result is attained at $z = m + n$.

If $x < m$ and y $< n$ one has $z = x{+}y$ and we involve the increasing parts of the membership function

$$
\begin{aligned}
z &= x+y = x = (m-a)\alpha + a + (n-c)\alpha + c \\
&= \alpha + c + \alpha(m-a+n-c)\alpha = a+c+(m+n-a-c)\alpha
\end{aligned}
$$

In this way for the corresponding increasing parts of the membership functions we obtain a linear membership function in the form $(z-(a+c))/((m+n)-(a+c))$.

If $x > m$ and $y > n$ we compute $z = x + y = (m-b)\alpha + b{+}(n-d)\alpha + d$.

Arranging the above results we have the triangular fuzzy set

$$
C(z) = \frac{z-(a+c)}{(m+n)-(a+c)} \tag{5.18}
$$

If $x > m$ and $y > n$ we compute $z = x + y = (m - b)\alpha + b + (n - d)\alpha + d = b + d + (m - b + n - d)\alpha$. We have the linear membership function given as $(z - (b + d))/((m + n) - (b + d))$.

In sum the result of addition of two triangular fuzzy numbers is a triangular fuzzy number with the membership function increasing linearly over $[a + c, m + n]$ and decreasing linearly over $[m + n, b + d]$. It is zero outside the range $[a + c, b + d]$.

The result has a straightforward interpretation where the values of the parameters of C are just the added values of the parameters of the corresponding arguments of the two fuzzy numbers, namely $a + c$, $m + n$, and $b + d$. In general, we have

$$\begin{aligned} &A_1(x; a_1, m_1, b_1) + A_2(x; a_2, m_2, b_2) + \cdots + A_p(x; a_p, m_p, b_p) \\ &= A\left(x; \sum_{i=1}^{p} a_i, \sum_{i=1}^{p} m_i, \sum_{i=1}^{p} b_i\right) \end{aligned} \tag{5.19}$$

where A is a triangular fuzzy set. The result for the subtraction is determined in an analogous way which leads to the result in the form of the triangular fuzzy number $D = A - B$, $D(x; a - c, m - n, b - d)$.

Example 8 Complete the addition of three triangular fuzzy numbers $A(x; 1, 2, 6)$, $B(x; 0, 5, 6)$ and $C(x; 4, 5, 7)$.

In virtue of the above formulas one obtains the following result $D = A + B + C$ with $D(x; 5, 12, 19)$.

5.2.4 Multiplication

As before, we elaborate on the three cases

$z = mn$. Here the membership function of the result yields $C(z) = 1$.

If $z < m$ and $z < n$, we consider the product of the increasing portions of the linear segments of the membership functions

$$\begin{aligned} z = xy &= [(m - a)\alpha + a][(n - c)\alpha + c] \\ &= (m - a)(n - c)\alpha^2 + (m - a)a\alpha + a(n - c)\alpha + ac = f(\alpha) \end{aligned}$$

Apparently, z being treated as a function of α is a second-order polynomial of α, $z = f(\alpha)$.

Thus the increasing part of the membership function of the result is $f^{-1}(z)$.

If $z > m$ and $z > n$ one has

$$z = xy = [(m - b)\alpha + b][(n - d)\alpha + d] = g(\alpha)$$

and then the decreasing portion of z is $g^{-1}(z)$. By combining the two results we have

$$C(z) = \begin{cases} f^{-1}(z), & \text{if } z < mn \\ 1, & \text{if } z = mn \\ g^{-1}(z), & \text{if } z > mn \end{cases} \tag{5.20}$$

5.2.5 *Division*

The calculations proceed in the same manner as before. For $z = m/n$ the membership grade achieves its maximal value $C(z) = 1$.

For the increasing parts of membership functions A and B, we obtain

$$z = \frac{x}{y} = \frac{(m-a)\alpha + a}{(n-c)\alpha + c} = f(\alpha) \tag{5.21}$$

This membership function $f^{-1}(\alpha)$ applies to all arguments of z within the range $[a/c, m/n]$. For the decreasing parts of A and B, one has

$$z = \frac{x}{y} = \frac{(m-b)\alpha + b}{(n-d)\alpha + d} = g(\alpha) \tag{5.22}$$

which holds for the interval $[m/n, b/d]$. Overall the resulting membership function is

$$C(z) = \begin{cases} f^{-1}(z), & \text{if } z < m/n \\ 1, & \text{if } z = m/n \\ g^{-1}(z), & \text{if } z > m/n \end{cases} \tag{5.23}$$

The generalized version of the extension principle in which the minimum operator is replaced by any t-norm offers more flexibility and leads to interactive algebraic operations on fuzzy numbers.

5.3 Conclusions

The fundamentals of transformation of fuzzy sets are presented here. With this regard the extension principle and the representation theorem play a central role. Through the representation theorem, the calculus of fuzzy sets (fuzzy numbers) is realized in the framework of interval calculus- the approach is simple however more computing overhead is encountered to come up with the fuzzy set of result generated by a series of α-cuts. The arithmetic operations are in common use in modeling with fuzzy sets as discussed in successive chapters.

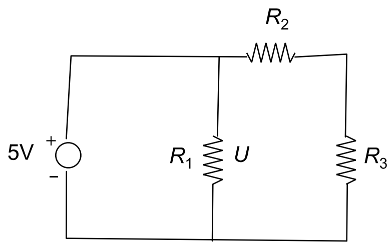


Fig. 5.9 Electric circuit

Problems

1. (i) Compute the sum of two triangular fuzzy numbers $A(x; -1, 0, 3)$ and $B(x; 0, 4, 7)$
   (ii) Determine the product of fuzzy numbers given above.
   Hint: the split into the problem into the three subproblems as outlined above does not work as the assumptions about zero membership grades for negative values of the arguments does not hold here
2. Contrast the two ways of mapping a fuzzy set through some function. Identify some advantages and limitations of these approaches. Offer your explanation.
3. A parabolic fuzzy number is described as follows

$$P(x;\ m,\ a) = \begin{cases} 1 - \left(\frac{x-m}{a}\right)^2, \ if\ x \in [m-a, m+a] \\ 0, \qquad \text{otherwise} \end{cases}$$

   i. Determine the sum of N fuzzy numbers $P(x; m_1, a_1) + P(x; m_2, a_2) + \cdots + P(x; m_N, a_N)$.
   ii. Determine the product of $P(x; m_1, a_1)$ and $P(x; m_2, a_2)$.
   iii. Determine the sum of $P(x; 2, 1)$ and an interval [5, 8].

4. For the circuit shown in Fig. 5.9 determine the voltage drop U considering that $R_1 = 3\ \Omega$, $R_2 =$ *about* $5\ \Omega$ with the triangular membership function $R(r; 2, 5, 6)$ and $R_3 = 2\ \Omega$.
5. Given two fuzzy sets with triangular membership functions, $A(x; -2, 1, 3)$ and $B(x; 0, 2,4)$. Determine the maximum of A and B, max (A, B).
6. Determine the fuzzy set abs(A), A^2, and sqrt(A) for A defined in $\boldsymbol{R}$

x	−5	−3	−1	0	2	3	6
$A(x)$	0.6	0.7	1.0	0.5	0.4	0.7	0.5

7. Given is a fuzzy number $A = (f, g)$. Approximate it with the use of a triangular fuzzy number.
   Hint: approximate f and g by linear functions by minimizing a sum of squared errors between $f(g)$ and the optimized function.

References

1. Dubois, D., Prade, H.: Operations on fuzzy numbers. Int. J. Syst. Sci. **9**, 613–626 (1978)
2. Jardón, D., Sánchez, I., Sanchis, M.: Some questions about Zadeh's extension on metric spaces. Fuzzy Sets Syst. (2019)
3. Pedrycz, W., Gomide, F.: Fuzzy Systems Engineering: Toward Human-Centric Computing. Wiley, New Jersey (2007)
4. Ross, T.J.: Fuzzy Logic with Engineering Applications. Wiley, West Sussex (2009)
5. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning, Parts 1, 2, 3. Inform. Sci. **8**, 199–249, 301–357, 43–80 (1975)

Chapter 6
Operations on Fuzzy Sets

Abstract In this chapter, we discuss set operations on fuzzy sets by beginning with early fuzzy set operations and continuing with their generalization, interpretations, formal requirements and realizations. We emphasize that complement and negations, triangular norms and triangular co-norms are unifying, general realizations of the set-theoretic operations. Combinations of fuzzy sets through aggregation processes are also essential by delivering an additional level of flexibility of processing fuzzy sets.

6.1 Generic Operations on Fuzzy Sets

Analogously, as encountered in set theory, we operate with fuzzy sets to obtain new fuzzy sets. The operations must exhibit properties that match intuition, comply with the semantics of the intended operation, and are also flexible enough to fit application requirements. Furthermore when the developed fuzzy set operators are applied to sets, they should return the same results as encountered when operating on sets [4].

It is instructive to start with the familiar operations of intersection, union and complement of sets.

Let us recall that operations on sets can be regarded as operations on their characteristic functions. Given are two sets A and B defined in the same space $\boldsymbol{X}$ along with

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_6

their characteristic functions, A(x) and $B(x)$. The operations of intersection and union are described by the minimum and maximum operations applied to the characteristic functions of A and B

$$(A \cap B)(x) = \min(A(x), B(x)) \tag{6.1}$$

$$(A \cup B)(x) = \max(A(x), B(x)) \tag{6.2}$$

For $\boldsymbol{X}$ regarded as a subset of real numbers, the plots of the union and intersection of two sets (intervals) are displayed in Fig. 6.1.

Likewise, the complement $\overline{A}$ of set A, expressed in terms of its characteristic function, is the complement of the characteristic function of A, $1 - A(x)$; refer to Fig. 6.2.

Table 6.1 summarizes the main properties of set operations [2, 6].

For fuzzy sets, the minimum, maximum, and complement operations on membership functions are considered and the plots of the obtained membership functions are covered in Fig. 6.3.

While A and B are normal unimodal fuzzy sets, their intersection is commonly a subnormal fuzzy set while the membership function of the union is multimodal. It is interesting to note that the operations on fuzzy sets (6.1)–(6.2) satisfy all the properties outlined in Table 6.1 with an exception of excluded middle and non-contradiction. An illustration of this departure is clearly presented in Fig. 6.3. Note that a so-called overlap and underlap effect comes as a result of membership grades

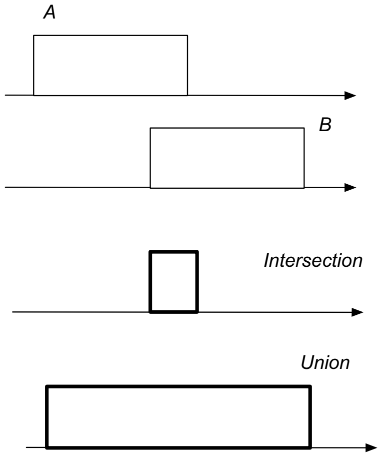


Fig. 6.1 Characteristic functions of A and B and their intersection and union

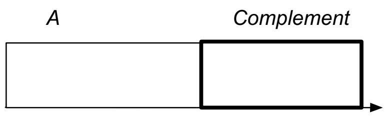


Fig. 6.2 Characteristic function of complement of A

Table 6.1 Main properties of operations on sets

Commutativity	$A \cup B = B \cup A$ $A \cap B = B \cap A$
Associativity	$A \cup (B \cup c) = (A \cup B) \cup C = A \cup B \cup C$ $A \cap (B \cap c) = (A \cap B) \cap C = A \cap B \cap C$
Idempotency	$A \cup A = A$ $A \cap A = A$
Distributivity	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Identity	$A \cup = \emptyset \quad A \cap \boldsymbol{X} = A$
Special absorption	$A \cup \boldsymbol{X} = \boldsymbol{X} \quad A \cap \emptyset = \emptyset$
Involution	$\overline{\overline{A}} = A$
Excluded middle	$A \cup \overline{A} = \boldsymbol{X}$
Non-contradiction	$A \cap \overline{A} = \emptyset$
Ttransitivity	$A \subset B$ and $B \subset C$ then $A \subset C$

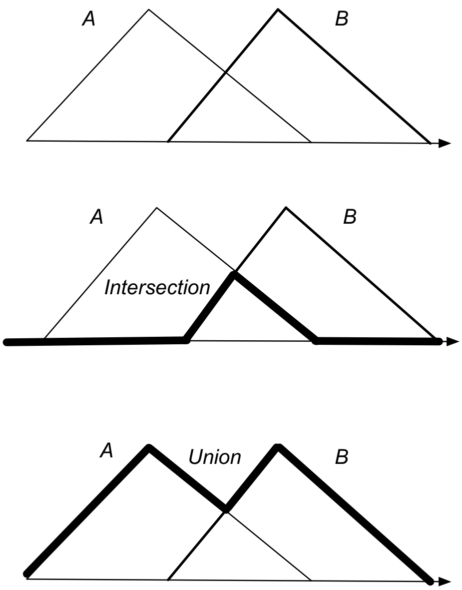


Fig. 6.3 Membership functions of intersection and union of fuzzy sets

which assume values located in-between 0 and 1. The plot of the membership of complement computed as $1 - A(x)$ is shown in Fig. 6.4.

Example 1 Considering the geometric representation of sets and fuzzy sets in the form of points in the unit hypercube, we display the results of intersection, union, and complement of sets described by their characteristic functions $A = [1\ 0]$ and $B = [0\ 1]$. The results are shown in Fig. 6.5a.

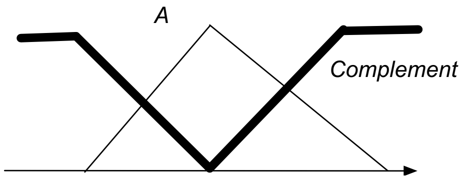


Fig. 6.4 Membership function of complement of A

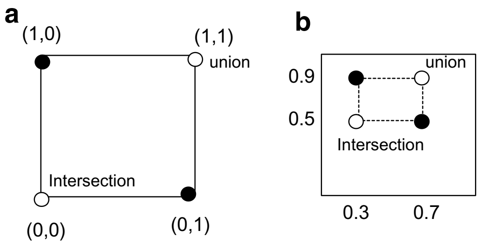


Fig. 6.5 Geometric illustration of intersection and union operation on **a** sets, and **b** fuzzy sets

As it is visible, the results of the operations are located at the corners of the {0,1} square.

Example 2 Let us develop a geometric representation of fuzzy sets $A = [0.7\ 0.5]$ $B = [0.3\ 0.9]$

Computing the minimum and maximum of the minimum of the membership grades, the results are [0.3 0.5] for the intersection and [0.7 0.9] for the union; see Fig. 6.5b.

Example 3 We determine and plot the complements of the fuzzy sets $A = [0.4\ 0.1]$ and $B = [0.5.\ 0.5]$. Again by calculating the values of the expression 6.1 $- A(x)$ the obtained results are shown in Fig. 6.6. It is noticeable that for B, its complement is equal to the same fuzzy set.

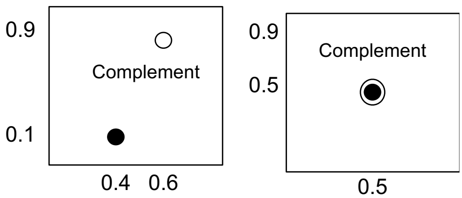


Fig. 6.6 Display of the complement of fuzzy set A and B

6.2 Triangular Norms and Triangular Co-norms as Models of Operations on Fuzzy Sets

Operations on fuzzy sets concern processing of their membership functions. Therefore, they are domain dependent and different contexts may require different realizations of intersection and union. For instance, since operations provide ways to combine pieces of information, they can be performed differently in image processing, control, and diagnostic systems, for example. When contemplating the realization of operations of intersection of fuzzy sets, we require a satisfaction of the following intuitively appealing properties:

$$\begin{aligned}&\text{commutativity} \quad atb = bta\\&\text{associativity} \quad at(btc) = (atb)tc = atbtc\\&\text{monotonicity} \quad \text{if } b \le c \text{ then } atb \le atc\\&\text{boundary conditions} \quad at1 = at0 = 0\end{aligned} \tag{6.3}$$

Here a and b are two membership degrees, $a = A(x)$, $b = B(x)$. Note that in terms of fuzzy sets, those requirements correspond with the properties of set operations shown in Table 6.1. The boundary conditions state that an intersection of any fuzzy set A with the universe of discourse $\boldsymbol{X}$ (whose membership is identically equal to 1) should return this fuzzy set. The intersection of any fuzzy set A and an empty fuzzy set (with the membership function equal identically zero) returns the empty fuzzy set.

The operations $[0,1] \times [0,1] \to [0,1]$ with the properties (6.3) are known in the literature as t-norms and have been introduced long time before the inception of fuzzy sets in the context of studies on probabilistic metric spaces [3, 5, 7].

Any t-norm can be regarded as a useful potential candidate to realize the operation of intersection of fuzzy sets. Note also that in virtue of the boundary conditions when confining to sets, the above stated operations return the same results as encountered in set theory: different t-norms are non-distinguishable when applied to sets. In general, idempotency is not required, however the realizations of union and intersection could be idempotent as this happens for the operations of minimum and maximum where $\min(a, a) = a$ and $\max(a, a) = a$.

Boundary conditions assure that all t-norms attain the same values at boundaries of the unit square $[0,1] \times [0,1]$.

The family of t-norms is infinite. Below we list some commonly encountered examples of t-norms

$$atb = \min(a, b) = a \wedge b \tag{6.4}$$

$$atb = ab \tag{6.5}$$

$$atb = \max(a + b - 1,\ 0) \tag{6.6}$$

$$atb = \begin{cases} \text{a, if } b = 1 \\ b, \text{ if } a = 1 \\ 0, \text{ otherwise} \end{cases} \tag{6.7}$$

$$atb = \frac{ab}{p + (1-p)(a + b - ab)}, \quad p \geq 0 \tag{6.8}$$

$$atb = \max(0,\ (1+p)(a+b-1) - pab), \quad p \geq -1 \tag{6.9}$$

$$atb = \sqrt[p]{\max(0, a^p + b^p - 1)} \quad p > 0 \tag{6.10}$$

Equation (6.6) is referred to as Lukasiewicz *and* operator whereas (6.7) is a so-called drastic product. In general, *t*-norms are not linearly ordered. One can demonstrate that the minimum *t*-norm is the largest among all *t*-norms while the drastic product is the smallest one. They form the lower and upper bounds of all *t*-norms in the following sense

$$\text{drastic product} \leq t\text{-norm} \leq \text{minimum} \tag{6.11}$$

The union operation on fuzzy sets is realized in terms *t*-conorms.

A *t*-conorm is a binary operation s: $[0,1] \times [0,1] \rightarrow [0,1]$ that satisfies the following requirements

$$\begin{aligned} &\text{commutativity} \quad asb = bsa \\ &\text{associativity} \quad as(bsc) = (asb)sc \\ &\text{monotonicity} \quad \text{if } b < c \text{ then } asb < asc \\ &\text{boundary conditions} \quad as\,0 = a \quad as\,1 = 1 \\ &a,\ b,\ c \in [0, 1]. \end{aligned} \tag{6.12}$$

The boundary conditions state that all *t*-conorms behave in the same way at the corners of the unit square $[0,1] \times [0,1]$. Thus, for sets, any *t*-conorm returns the same result as encountered in set theory.

Example 4 If A and B are convex fuzzy sets defined in the space of real numbers, show that their intersection realized by the minimum operation is a convex fuzzy set.

Let us recall that the convexity of A and B means that for any x_1 and x_2 the following inequalities are satisfied

$$\begin{aligned} A(\lambda x_1 + (1-\lambda)x_2) &\geq \min(A(x_1), A(x_2)) \\ B(\lambda x_1 + (1-\lambda)x_2) &\geq \min(B(x_1), B(x_2)) \end{aligned}$$

The intersection of A and B, $C = A \cap B$ comes with the membership function $C(x) = \min(A(x), B(x))$ thus

$$\begin{aligned} C(\lambda x_1 + (1-\lambda)x_2) &= \min(A(\lambda x_1 + (1-\lambda)x_2),\ B(\lambda x_1 + (1-\lambda)x_2)) \\ &\geq \min[\min(A(x_1), A(x_2)),\ \min(B(x_1), B(x_2))] \\ &\geq \min[\min(A(x_1), B(x_1)),\ \min(B(x_2), A(x_2))] \\ &= min(C(x_1), C(x_2)) \end{aligned}$$

viz. $C(\lambda x_1 + (1-\lambda)x_2) \geq \min(C(x_1), C(x_2))$.

One can show that s: $[0,1] \times [0,1] \rightarrow [0,1]$ is a t-conorm if and only if (iff) there exists a t-norm (so-called dual t-norm) such that for any a and b, we have

$$asb = 1 - (1-a)t(1-b) \tag{6.13}$$

For the corresponding dual t-norm we have

$$atb = 1 - (1-a)s(1-b) \tag{6.14}$$

The duality expressed by (6.13) and (6.14) can be viewed as an alternative definition of t-conorms. This duality allows us to deduce the properties of t-conorms on the basis of the analogous properties of t-norms. Notice that after a slight rewrite of (6.13)–(6.14), we obtain

$$(1-a)t(1-b) = 1 - asb \tag{6.15}$$

$$(1-a)s(1-b) = 1 - atb \tag{6.16}$$

These two relationships can be expressed symbolically as

$$\overline{A} \cap \overline{B} = \overline{A \cup B} \tag{6.17}$$

$$\overline{A} \cup \overline{B} = \overline{A \cap B} \tag{6.18}$$

We recognize that they are just the De Morgan laws.

A list of commonly used t-conorms includes the following examples

$$asb = \max(a, b) = a \vee b \tag{6.19}$$

$$asb = a + b - ab \tag{6.20}$$

$$asb = \min(a + b, 1) \tag{6.21}$$

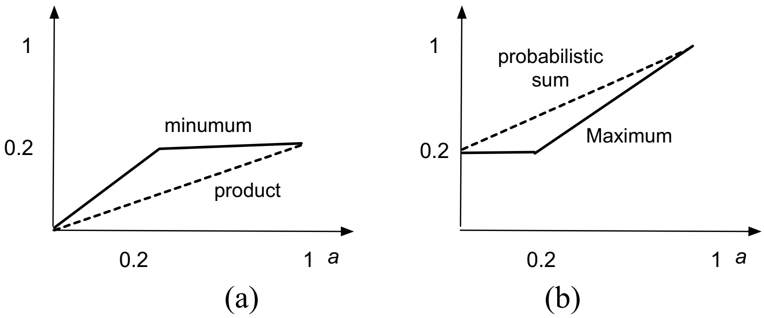


Fig. 6.7 Plots of the minimum (solid line) and product (dotted line) (**a**), the dual co-norms (maximum and probabilistic sum) (**b**); $b = 0.2$

$$asb = \begin{cases} a. & \text{if } b = 0 \\ b, & \text{if } a = 0 \\ 1, & \text{otherwise} \end{cases} \tag{6.22}$$

$$asb = \frac{a + b - ab - (1-p)ab}{1 - (1-p)ab}, \quad p \geq 0 \tag{6.23}$$

$$asb = 1 - \max(0, \sqrt[p]{(1-a)^p + (1-b)^p - 1}), \, p > 0 \tag{6.24}$$

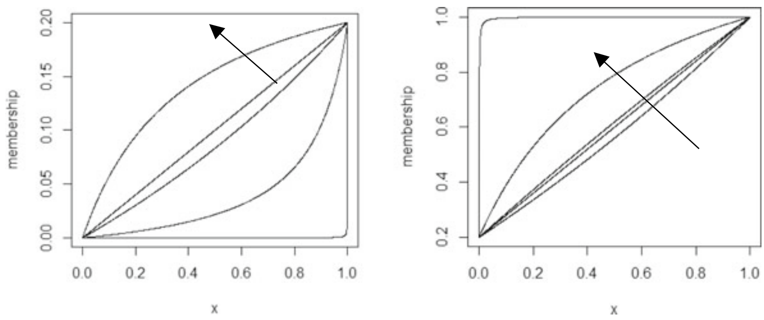
$$asb = min(1, a + b + pab), \quad p \geq 0 \tag{6.25}$$

The Formula (6.20) is the probabilistic sum, (6.21) is the Lukasiewicz *or* operator, and (6.22) is referred to as a drastic sum. t-conorms are not linearly ordered and the bounds are

$$\max \leq t\text{-conorm} \leq \text{drastic sum} \tag{6.26}$$

The plots of the two commonly used t-norms (minimum and the product) and the dual t-conorm (maximum and the probabilistic sum) are presented in Fig. 6.7. Here the value of one argument is fixed ($b = 0.2$) while the second one varies over the unit interval.

The plots of $a\ t\ b = \frac{ab}{p+(1-p)(a+b-ab)}$ and $a\ s\ b = \frac{a+b-ab-(1-p)ab}{1-(1-p)ab}$, for $b = 0.2$ and selected values of p, $p = 0.1$, 1, 1.5, 10, and 10,000 are included in Fig. 6.8.



(a) (b)

Fig. 6.8 Plots of membership grades *atb* (**a**) and *asb* (**b**). The arrows points at the direction of the increasing values of the parameter (p)

6.3 Negations

Negation is a single-argument operation, which is a generalization of the complement operation encountered in set theory. More formally, by a negation we mean a function N: [0,1] $\to$ [0,1] satisfying the following conditions:

$$\begin{array}{ll} \text{monotonicity} & N \text{ is nonincreasing} \\ \text{boundary conditions} & N(0) = 1,\ N(1) = 0 \end{array} \tag{6.27}$$

If the function N is continuous and decreasing, the negation is called strict [4]. If, in addition, a strict negation is involutive, that is $N(N(a)) = a$ for all a in the unit interval, it is called *strong*.

Two realizations of the negation operator include the following

$$N(a) = \sqrt[w]{1 - a^w}, \quad w > 0 \tag{6.28}$$

$$N(a) = \frac{1 - a}{1 + \lambda a} \tag{6.29}$$

Interestingly, if in the above expressions we set $w = 1$ or $\lambda = 0$, these realizations of the negation return the standard complement function, that is $N(a) = 1 - a$.

6.4 Aggregation Operators

Fuzzy sets could be aggregated in many different ways. Formally, an aggregation operation agg: $[0,1]^n \to [0,1]$ is an n-argument mapping satisfying the following requirements

(i) boundary condition $\text{agg}(0, 0, \ldots 0) = 0$ $\text{agg}(1, 1, \ldots, 1) = 1$

(ii) monotonicity $\text{agg}(a_1, a_2, \ldots, a_n) \geq \text{agg}(b_1, b_2, \ldots, b_n)$

for $a_i \geq b_i, i = 1, 2, \ldots, n$ (6.30)

Triangular norms are examples of aggregation operations however there are a number of other interesting alternatives. Below we list some of them.

6.4.1 Ordered Weighted Average

The ordered weighted average (*OWA*) [8] is an interesting example of an aggregation operator which exhibits a significant level of flexibility embracing the number of aggregation operators. Given a vector of membership grades $\boldsymbol{a} = [a_1\ a_2 \ldots a_n]$ such that $a_1 \leq a_2 \leq \cdots \leq a_n$ and weight vector $\boldsymbol{w} = [w_1\ w_2 \ldots w_n]$, where $w_i \in [0,1]$ and $\sum_{i=1}^{n} w_i = 1$, the OWA ($\boldsymbol{w}$; $\boldsymbol{a}$) is expressed as

$$\text{OWA}(\boldsymbol{w};\ \boldsymbol{a}) = \sum_{i=1}^{n} a_i w_i \tag{6.31}$$

The flexibility of the construct associates with the adjustable weight vector. Depending on the entries of $\boldsymbol{w}$ one has the minimum, maximum and the average operations as some special cases of this aggregation

$$\boldsymbol{w} = [1\ 0 \ldots 0] \text{OWA}(\boldsymbol{w};\ \boldsymbol{a}) = a_1 = \min(a_1\ a_2 \ldots a_n)$$
$$\boldsymbol{w} = [0\ 0 \ldots 0\ 1] \text{OWA}(\boldsymbol{w};\ \boldsymbol{a}) = a_n = \max(a_1\ a_2 \ldots a_n)$$
$$\boldsymbol{w} = [1/n\ 1/n \ldots 1/n]\ \text{OWA}(\boldsymbol{w};\ \boldsymbol{a}) = \frac{1}{n}\sum_{i=1}^{n} a_i$$

Both minimum and maximum are examples of the OWA operator. Note that the ordering of the values of $\boldsymbol{a}$ is essential to the calculations completed above.

Example 5 Let us determine the weight vector so that the *OWA* operator returns the median of its arguments.

We consider two situations; (i) the number of elements is odd. The weights are as follows [0..0 1 0…0] where the only nonzero entry of the weight vector is at position $(n + 1)/2$. For instance, if $n = 7$, $\boldsymbol{w} = [0\ 0\ 0\ 1\ 0\ 0\ 0]$. If n is an even number then $\boldsymbol{w}$ has two nonzero entries located at $n/2$ and $n/2 + 1$ entries $\boldsymbol{w} = [0 \ldots 0\ ½\ ½\ 0 \ldots 0]$. For instance, if $n = 6$ $\boldsymbol{w} = [0\ 0\ ½\ ½\ 0\ 0]$.

6.4.2 Compensatory Operators

An interesting situation occurs when an aggregation of membership values is neither of disjunctive nor conjunctive nature but falls somewhere in-between exhibiting a certain degree of *orness* (*andness*). A so-called compensative operator was proposed in (Zimmerman and Zysno [9]

$$(A\blacksquare B)(x) = [(A \cap B)(x)]^{1-\gamma}[(A \cup B)(x)]^{\gamma} \tag{6.32}$$

A and B are two fuzzy sets defined in the same space $\boldsymbol{X}$ and γ stands for the compensation factor assuming values in [0,1]. Noticeably, if γ is equal 0, one has the union operator while for $\gamma = 1$, the aggregation operation reduces to the intersection operator. There is a spectrum of possibilities in-between which are generated by changing the values of γ. Another realization of the compensatory operator comes in the form of the convex combination

$$(A\blacksquare B)(x) = (1-\gamma)[(A \cap B)(x)] + \gamma[(A \cup B)(x)] \tag{6.33}$$

As illustrated in Fig. 6.9, changing the values of γ produce a smooth transition from the or like nature of aggregation to the and form of the connective; the compensation effect becomes clearly visible.

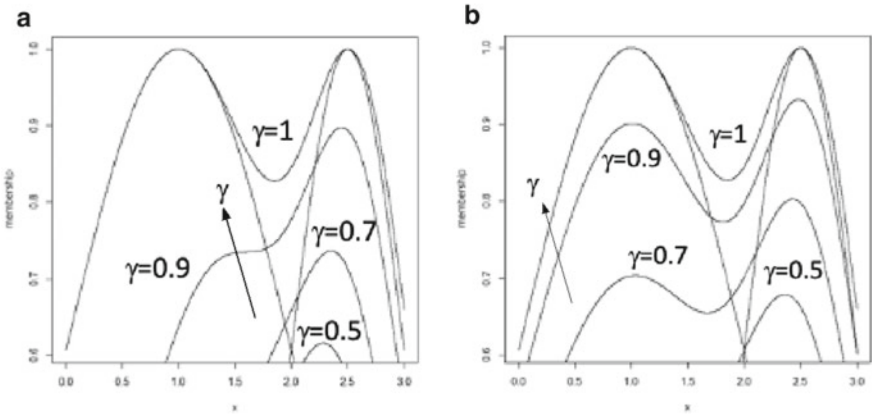


Fig. 6.9 Plots of the compensatory operators for selected values of γ: **a** formula (6.32), **b** formula (6.33)

6.4.3 Averaging Operator

An averaging operator (generalized mean) is expressed in the following parameterized form [1]

$$agg(a_1, a_2, \ldots, a_n) = \sqrt[p]{\frac{1}{n}\sum_{i=1}^{n}(a_i)^p} \tag{6.34}$$

When p is a certain parameter making the class of generalized mean a generalized class of operators. The averaging operator is idempotent, commutative, monotonic and satisfies the boundary conditions agg(0,0,…0) = 0 agg(1, 1, …,1) = 1

Depending on the values of the parameter p, there are several interesting cases

$p = 1$ arithmetic mean $agg(a_1, a_2, \ldots, a_n) = \frac{1}{n}\sum_{i=1}^{n}(a_i)$
$p \to 0$ geometric mean $\text{agg}(a_1, a_2, \ldots, a_n) = (a_1 a_2 \ldots a_n)^{1/n}$
$p = -1$ harmonic mean $agg(a_1, a_2, \ldots, a_n) = \frac{n}{\sum_{i=1}^{n}(1/a_i)}$
$p \to -\infty$ maximum $\text{agg}(a_1, a_2, \ldots, a_n) = \max(a_1, a_2, \ldots, a_n)$
$p \to\to \infty$ minimum $\text{agg}(a_1, a_2, \ldots, a_n) = \min(a_1, a_2, \ldots, a_n)$.

6.4.4 Sensitivity Analysis of Aggregation Operators

From the point of view of applications, aggregation operators are impacted by the changes of the arguments. This impact is quantified with the concept of sensitivity. We are interested: how much changes in the results of aggregation operator are affected by the changes in its values? Two sensitivity measures are considered [4, 6].

6.4.5 Extreme Sensitivity

This measure is expressed in the following form

$$\begin{aligned} &s(d) = \text{Max}_{a,b}|\text{agg}(\boldsymbol{a}) - \text{agg}(\boldsymbol{b})| \\ &\text{subject to } |a_1 - b_1| \le \delta, |a_2 - b_2| \le \delta \ldots |a_n - b_n| \le \delta \end{aligned} \tag{6.35}$$

where $\boldsymbol{a} = [a_1\ a_2 \ldots a_n]$ and $\boldsymbol{b} = [b_1\ b_2 \ldots b_n]$.

For given δ in [0,1] we determine the maximum of the absolute difference between the agg($\boldsymbol{a}$) and agg($\boldsymbol{b}$).

Example 6 Determine the extreme sensitivity of the product for $n = 2$; agg(a_1, a_2) $= a_1 a_2$. The optimization problem comes in the form

$$s(d) = \text{Max}_{a1,a2,b1,b2}|a_1 a_2 - b_1 b_2|$$

$$\text{subject to } |a_1 - b_1| \leq \delta \text{ and } |a_2 - b_2| \leq \delta$$

We rewrite the above constraints as follows $-\delta \leq a_1 - b_1 \leq \delta$ and $-\delta \leq a_2 - b_2 \leq \delta$. Then we compute

$$\begin{aligned} s(\delta) &= \text{Max}_{b1,b2}|(b_1 - \delta)(b_2 - \delta) - b_1 b_2| \\ &= \text{Max}_{b1,b2}\left|b_1 b_2 - \delta b_2 - \delta b_1 + \delta^2 - b_1 b_2\right| \\ &= \text{Max}_{b1,b2}\left|\delta^2 - (\delta b_2 + \delta b_1)\right| = \left|\delta^2 - 2\delta\right| = 2\delta - \delta^2 \end{aligned}$$

The extreme sensitivity of the product is a quadratic function of δ; it is an increasing function of its argument.

6.4.6 Average Sensitivity

As the name stipulates, the measure determines the following expression which realizes averaging over the space of all arguments involved in the aggregation operation

$$s = \int\limits_{[0,1]^n} \left[\sum_{i=1}^{n} \left(\frac{\partial agg(a_1, a_2, \ldots, a_i, \ldots, a_n)}{\partial a_i} \right)^2 \right] da_i \tag{6.36}$$

Example 7 Let us determine the average sensitivity of the product (t-norm) with two arguments. The formula (6.36) reads now as

$$s = \int\limits_{[0,1]} \int\limits_{[0,1]} \left[\left(\left(\frac{\partial \text{agg}(a_1, a_2)}{\partial a_1} \right)^2 + \frac{\partial \text{agg}(a_1, a_2)}{\partial a_2} \right)^2 \right] da_1 da_2$$

$$s = \iint_0^1 \left[a_2^2 + a_1^2\right] da_1 da_2 = 1/3 + 1/3 = 2/3$$

6.5 Conclusions

Aggregation of fuzzy sets comes with a plethora of different operators. Union and intersection of fuzzy sets is commonly implemented with the use triangular norms and conforms with intent to capture the semantics of logic connectives. Their usefulness becomes apparent when coping with real-world problems where there arises a need to carefully address aspects of compensation, interactivity and sensitivity of operations on information granules.

Problems

1. Consider the Zimmermann–Zysno data (Zimmermann and Zysno [9] coming as triples of (a_k, b_k, c_k) and experiment with various t-norms and t-conorms applied to pairs (a_k, b_k) and producing result $c_k{}' = a_k t b_k$ or $a_k s b_k$. Calculate the sum of distances $V = \sum_k \left|c_k' - c_k\right|$ and discuss the usefulness of various logic connectives in case of these data in producing the lowest value of this sum. Report your results in a tabular form; discuss which logic operator led to the minimal value of V.

t-norm/t-conorm									
V									

2. Demonstrate that the drastic sum and drastic product satisfy the law of excluded middle and the law of contradiction.
3. Show that Lukasiewicz *and* and *or* operators satisfy the law of excluded middle and the law of contradiction.
4. Consider two fuzzy sets with triangular membership functions $A(x; 5, 6, 10)$ and $B(x; 1,3,4)$. Determine the intersection and union using product and probabilistic sum.
5. Determine the membership function $C = ((\overline{A} \cup B) \cap A$ using the maximum and minimum operations for A and B defined as follow

x	x_1	x_2	x_3	x_4	x_5
A	0.7	0.2	0.1	0.9	1.0
B	0.4	0.6	1.0	0.5	0.1

6. The modification operators applied to the membership function A are described as follows

$$\mathrm{mod}_1(A)(x) = \begin{cases} 2A(x)^2 & \text{if } A(x) \leq 0.5 \\ 1 - 2(1 - A(x))^2 & \text{if } A(x) > 0.5 \end{cases}$$

and

$$\mathrm{mod}_2(A)(x) = \begin{cases} \sqrt{\mathrm{A(x)/2}} & \text{if } A(x) \leq 0.5 \\ 1 - \sqrt{(1 - A(\mathrm{x}))/2} & \text{if } A(\mathrm{x}) > 0.5 \end{cases}$$

Plot the membership functions of $\mathrm{mod}_1(A)$ and $\mathrm{mod}_2(A)$ considering A to be a Gaussian fuzzy set $A(x) = \exp(-(x - 1)^2/2)$. Discuss how the original membership function has been modified by using these modifiers.

7. Show that the maximum and minimum operations can be described as follows

$$\min(x,\ y) = (|x + y| - |x - y\})/2 \quad \text{and} \quad \max(x, y) = (|x + y| + |x - y\})/2.$$

8. Plot the graph of the negation operator (28) for selected values of w. How does this parameter impact the shape of the relationship?
9. You are assessing a potential of real estate by considering several features as closeness to school, low pollution level, low property taxes, safe neighborhood. What logic operator would you choose to describe a concept of desirable real estate.
10. Determine result of OWA operator applied to $\boldsymbol{a} = [0.4\ 0.7\ 0.9\ 0.1\ 0.6]$ with $w_i = 1/5$, $i = 1, 2\ldots, 5$.

References

1. Dyckhoff, H., Pedrycz, W.: Generalized means as a model of compensative connectives. Fuzzy Sets Syst. **14**, 143–154 (1984)
2. Hanss, M.: Applied Fuzzy Arithmetic. Springer, Berlin (2005)
3. Menger, K.: Statistical metric spaces. Proc. Nat. Acad. Sci. (USA) **28**, 535–537 (1942)
4. Pedrycz, W., Gomide, F.: An Introduction to Fuzzy Sets. Analysis and Design. MIT Press, Cambridge (1998)
5. Schweizer, B., Sklar, A.: Probability Metric Spaces. North Holland, New York (1983)
6. Walker, E.A., Nguyen, H.T., Walker, C.L.: A First Course in Fuzzy Logic, 4th edn. CRC Press, Boca Raton, Fl (2019)
7. Yager, R.R.: On a general class of fuzzy connectives. Fuzzy Sets Syst. **4**, 235–242 (1980)
8. Yager, R.R.: On ordered weighted averaging aggregation operations in multicriteria decision-making. IEEE Trans. Syst. Man Cybernet. **18**, 183–190 (1988)
9. Zimmermann, H.J., Zysno, P.: Latent connectives in human decision making. Fuzzy Sets Syst. **43**, 37–51 (1980)

Chapter 7
Higher Type, Higher Order Fuzzy Sets and Hybrid Fuzzy Sets

Abstract In this chapter, we provide an introduction to more advanced, hierarchically structured information granules such as those of higher type and higher order. In general, when talking about information granules of higher type, say type-2, we mean information granules whose elements are characterized by membership grades, which themselves are information granules (instead of being plain numeric values). For instance, in type-2 fuzzy sets, membership grades are quantified as fuzzy sets in [0,1] or intervals in the unit interval. Of course, one could envision a plethora of the constructs along this line; for instance, the membership grades could be rough sets or probability density functions (as this is the case in probabilistic sets). When talking about higher order information granules, we mean constructs for which the universe of discourse comprises a family of information granules instead of single elements. Hybridization, on the other hand, is about bringing several formalisms of information granules and using them in an orthogonal setting. This situation is visible in fuzzy probabilities.

7.1 Fuzzy Sets of Higher Order

There is an apparent distinction between *explicit* and *implicit* manifestations of fuzzy sets. This observation triggers further conceptual investigations and leads to the concept of fuzzy sets of higher order. Let us recall that a fuzzy set is defined in a certain universe of discourse $\boldsymbol{X}$ so that for each element of it we come up with the

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_7

corresponding membership degree which is interpreted as a degree of compatibility of this specific element with the concept conveyed by the fuzzy set under discussion. The essence of a fuzzy set of 2nd order is that it is defined over a collection of some generic fuzzy sets. As an illustration, let us consider a concept of a *comfortable* temperature which we define over a finite collection of some generic fuzzy sets, say *around* 10 °C, *warm*, *hot*, *cold*, *around* 20 °C, … etc. We could easily come with a quick conclusion that the term *comfortable* sounds more "descriptive" and hence becomes semantically more advanced being in rapport with real-world concepts in comparison to the generic terms using which we describe it. An illustration of this 2nd order fuzzy set is illustrated in Fig. 7.1. To make a clear distinction and come up with a coherent terminology, fuzzy sets studied so far can be referred to as fuzzy sets of the 1st order.

Using the membership degrees as portrayed in Fig. 7.1, we can write down the membership of *comfortable* temperature in the vector form as [0.7 0.1 0.9 0.8 0.3]. It is understood that the corresponding entries of this vector pertain to the generic fuzzy sets we started with when forming the fuzzy set. Figure 7.2 graphically emphasizes the difference between fuzzy sets (fuzzy sets of the first order) and fuzzy sets of the second order. For the order 2 fuzzy set, we can use the notation $B = [\lambda_1, \lambda_2, \lambda_3]$

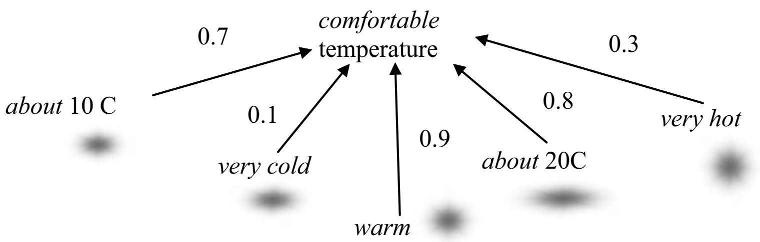


Fig. 7.1 An example of 2nd order fuzzy set of *comfortable* temperature defined over a collection of basic terms—generic fuzzy sets (graphically displayed as small clouds); shown are also corresponding membership degrees

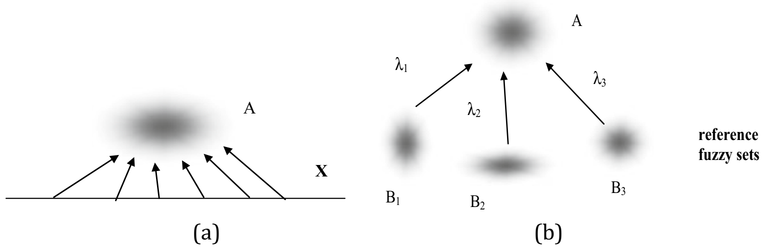


Fig. 7.2 Contrasting fuzzy sets of order 1 (**a**) and order 2 (**b**). Note a role of reference fuzzy sets played in the development of order 2 fuzzy sets

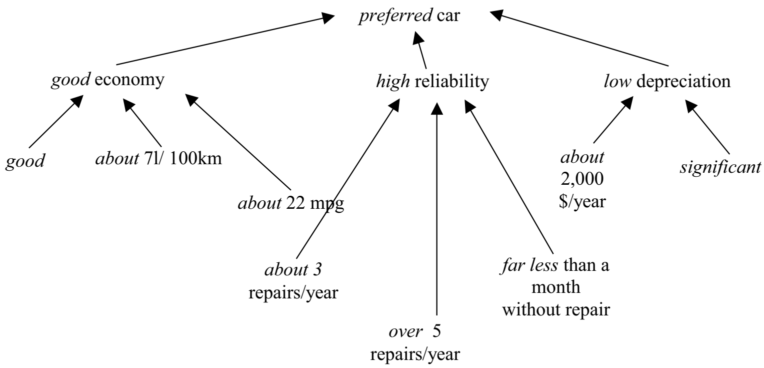


Fig. 7.3 Fuzzy set of order 2 of *preferred* cars; note a number of descriptors quantified in terms of fuzzy sets and contributing directly to its formation

given that the reference fuzzy sets are A_1, A_2, and A_3 and the corresponding entries of B are the numeric membership grades specifying to which extent the reference fuzzy sets form the higher order concept.

Fuzzy sets of order 2 could be also formed on a Cartesian product of some families of generic fuzzy sets. Consider, for instance, a concept of a *preferred* car. To everybody this term could mean something else yet all of us agree that the concept itself is quite complex and definitely multidimensional. We easily include several aspects such as economy, reliability, depreciation, acceleration, and others. For each of these aspects we might have a finite family of fuzzy sets, say when talking about economy, we may use descriptors such as *about* 10 l/100 km (alternatively expressed in mpg), *high* fuel consumption, *about* 30 mpg, etc. For the given families of generic fuzzy sets in the vocabulary of generic descriptors we combine them in a hierarchical manner as illustrated in Fig. 7.3.

In a similar way, we can propose fuzzy sets of higher order, say 3rd order or higher. They are formed in a recursive manner. While conceptually appealing and straightforward, its applicability could become an open issue. One may not wish to venture in allocating more effort into their design unless there is a legitimate reason behind the further usage of fuzzy sets of higher order.

Nothing prevents us from building fuzzy sets of 2nd order on a family of generic terms that are not only fuzzy sets. One might consider a family of information granules such as sets over which a certain fuzzy set is being formed.

7.2 Rough Fuzzy Sets and Fuzzy Rough Sets

It is interesting to note that the vocabulary used in the above construct could comprise information granules being expressed in terms of any other formalism, say fuzzy sets.

Quite often we can encounter constructs like rough fuzzy sets and fuzzy rough sets in which both fuzzy sets and rough sets are organized together.

These constructs rely on the interaction between fuzzy sets and sets being used in the construct. Let us consider a finite collection of sets $\{A_i\}$ and use them to describe some fuzzy set X. In this scheme, we arrive at the concept of a certain fuzzy rough set, refer to Fig. 7.4. The upper bound of this fuzzy rough set is computed as in the previous case yet given the membership function of X the detailed calculations return membership degrees rather than 0-1 values. Given the binary character of A_i's the above expression for the upper bound comes in the form

$$X_+(A_i) = \sup_x[\min(A_i(x), X(x))] = \sup_{x \in \sup p(A_i)} X(x) \tag{7.1}$$

The lower bound of the resulting fuzzy rough set is taken in the form

$$X_-(A_i) = \inf_x[\max(1 - A_i(x), X(x))] \tag{7.2}$$

Example 1 Let us consider a universe of discourse $\boldsymbol{X} = [-3, 3]$ and a collection of intervals regarded as a family of basic granular descriptors, see Fig. 7.5.

The fuzzy set A with a triangular membership function distributed between −2 and 2 gives rise to some rough set with the lower and upper approximation in the following form

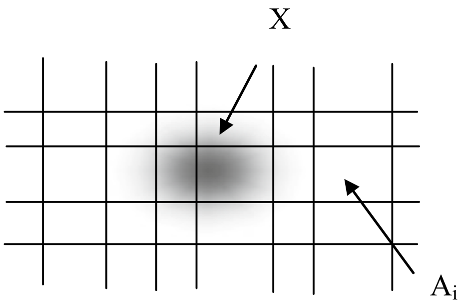


Fig. 7.4 The development of the fuzzy rough set

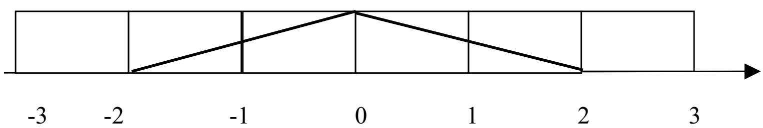


Fig. 7.5 Family of generic descriptors, fuzzy set and its representation in the form of some rough set

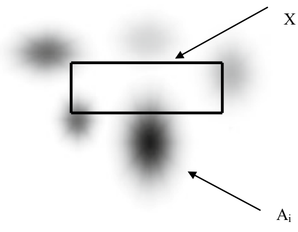


Fig. 7.6 A concept of a rough fuzzy set

$$X_+ = [0.0\,0.5\,1.0\,1.0\ \ 0.5\,0.0] \text{ and } X_- = [0.0\,0.0\,0.5\,0.5\,0.0\,0.0].$$

We can also consider another combination of information granules in which $\{A_i\}$ is a family of fuzzy sets and X is a set, see Fig. 7.6. This leads us to the concept of rough fuzzy sets.

Alternatively, we can envision a situation in which both $\{A_i\}$ and X are fuzzy sets. The result comes with the lower and upper bound whose computing follows the formulas presented above.

7.3 Type-2 Fuzzy Sets

Type-2 fuzzy sets form an intuitively appealing generalization of interval-valued fuzzy sets. Instead of intervals of numeric values of membership degrees, we allow for the characterization of membership by fuzzy sets themselves. Consider a certain element of the universe of discourse, say x. The membership of x to A is captured by a certain fuzzy set formed over the unit interval. This construct generalizes the fundamental idea of a fuzzy set and helps us relieve from the restriction of having single numeric values describing a given fuzzy set [7, 8]. An example of type-2 fuzzy set is illustrated in Fig. 7.7.

With regard to these forms of generalizations of fuzzy sets, there are two important facets that should be taken into consideration. First, there should be a clear motivation and a straightforward need to develop and use them. Second, it is imperative that

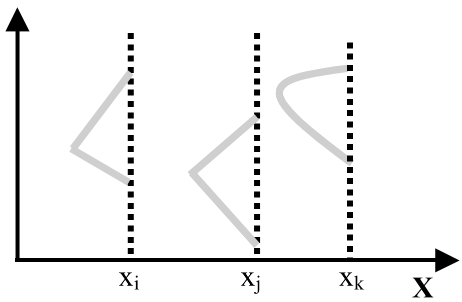


Fig. 7.7 An illustration of type-2 fuzzy set; for each element of ***X*** there is a corresponding fuzzy set of membership grades

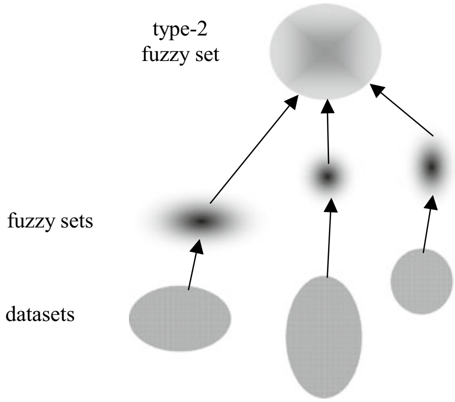


Fig. 7.8 A scheme of aggregation of fuzzy sets induced by P datasets

there is sound membership determination procedure in place using which we can construct the pertinent fuzzy set.

To elaborate on these two issues, let us discuss a situation in which we deal with several databases populated by data coming from different regions of the same country. Using them we build a fuzzy set describing a concept of *high* income where the descriptor "*high*" is modeled as a certain fuzzy set. Given the experimental evidence, the principle of justifiable granularity could be a viable development alternative to pursue. By being induced by some locally available data, the concept could exhibit some level of variability yet we may anticipate that all membership functions might be quite similar as being reflective of some general commonalities. Given that the individual estimated membership functions are trapezoidal (or triangular), we can consider two alternatives to come up with some aggregation of the individual fuzzy sets, see Fig. 7.8.

To fix the notation, for P databases, the corresponding estimated trapezoidal membership functions are denoted by $A_1(x;\, a_1,\, m_1,\, n_1,\, b_1)$, $A_2(x;\, a_2,\, m_2,\, n_2,\, b_2)$, …, $A_P(x;\, a_P,\, m_P,\, n_P,\, b_P)$, respectively. The first aggregation alternative leads to the emergence of an interval-valued fuzzy set $A(x)$. Its membership function assumes interval values where for each x the interval of possible values of the membership grades is given in the form $[\min_i A_i(x;\, a_i,\, m_i,\, n_i,\, b_i),\ \max_i A_i(x;\, a_i,\, m_i,\, n_i,\, b_i)]$. Alternatively, one can form the interval-valued fuzzy set by considering the principle of justifiable granularity; refer to Chap. 10. In this case the result is a trapezoidal fuzzy set.

Example 2 For a collection of trapezoidal fuzzy sets in Fig. 7.9a, by taking the minimal and maximal values of the membership grades for each x, we obtain a fuzzy set with interval values of membership grades. The result is shown in Fig. 7.9b where the interval-valued membership grades are shown.

The form of the interval valued fuzzy set may be advantages in further computing yet the estimation process could be very conservative leading to very broad ranges of membership grades (which is particularly visible when dealing with different data and fuzzy sets induced on their basis).

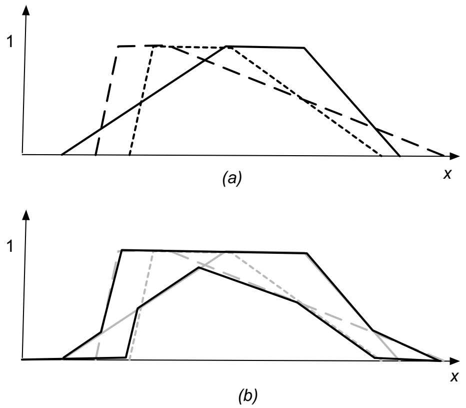


Fig. 7.9 A family of trapezoidal fuzzy sets (**a**) and their interval-valued fuzzy set (**b**)

7.4 Interval-Valued Fuzzy Sets

When defining or estimating membership functions or membership degrees, one may argue that characterizing membership degrees as single numeric values could be counterproductive and even somewhat counterintuitive given the nature of fuzzy sets themselves. Some remedy could be sought along the line of capturing the semantics of fuzzy sets through intervals of possible membership grades rather than single numeric entities [3, 6]. This gives rise to the concept of so-called interval-valued fuzzy sets. Formally, an interval-valued fuzzy set A is defined by two mappings from $\boldsymbol{X}$ to the unit interval $A = (A_-, A_+)$ where A_- and A_+ are the lower and upper bound of membership grades, $A_-(x)$ $A_+(x)$ for all $x \in \boldsymbol{X}$ where $A_-(x) \leq A_+(x)$. The bounds are used to capture an effect of a lack of uniqueness of numeric membership – not knowing the detailed numeric values we admit that there are some bounds of possible membership grades. Hence emerges the name of the interval-valued fuzzy sets, which becomes very much descriptive of the essence of the construct. The broader the range of the membership values, the less specific we are about membership degree of the element to the information granule. An illustration of the interval-valued fuzzy set is included in Fig. 7.10. Again the principle of justifiable granularity can serve here as a viable design vehicle.

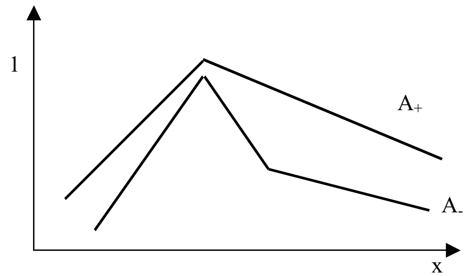


Fig. 7.10 An illustration of an interval-valued fuzzy set; note that the lower and upper bound of possible membership grades could differ quite substantially across the universe of discourse

In particular, when $A_{-}(x) = A_{+}(x)$, we end up with a "standard" (type-1) fuzzy set. The operations on interval-valued fuzzy sets are defined by considering separately the lower and upper bounds describing ranges of membership degrees. Given two interval-valued fuzzy sets $A = (A_{-}, A_{+})$ and $B = (B_{-}, B_{+})$ their union, intersection, and complement are introduced as follows,

$$\begin{aligned} \text{union } (\cup) \quad & \left(\max\left(A^{-}(x), B^{-}(x)\right), \max\left(A^{+}(x), B^{+}(x)\right)\right) \\ \text{intersection } (\cap) \quad & \left(\min\left(A^{-}(x), B^{-}(x)\right), \min\left(A^{+}(x), B^{+}(x)\right)\right) \\ \text{complement} \quad & \left(1 - A^{+}(x),\ 1 - A^{-}(x)\right) \end{aligned} \tag{7.3}$$

Example 3 A fuzzy set of preference over a finite set of alternatives $a_1, a_2, \ldots, a_n$ can be estimated using the method of pairwise comparison. In group decision-making. There is a panel of experts and each of them develops his/her own fuzzy set of preferences. It is likely that the individual results could vary. With the group of N experts, we end up having N fuzzy sets $A_1, A_2, \ldots, A_N$. They naturally give rise to an interval-valued fuzzy set whose membership function is constructed with the aid of the principle of justifiable granularity. For instance, for the ith alternative we have the numeric membership grades $A_1(a_i)$, $A_2(a_i)$, …, $A_N(a_i)$. The principle produces an interval information granule in the form $[a_i-, a_i+]$.

7.5 Probabilistic Sets

The idea of probabilistic sets introduced by Hirota [4]; see also Hirota and Pedrycz [5] has emerged in the context of pattern recognition where an existence of various sources of uncertainty is recognized including ambiguity of objects and their properties and subjectivity of observers.

From the general conceptual perspective, probabilistic sets build upon fuzzy sets in which instead of numeric membership grades of the membership function, we consider associated probabilistic characteristics. Formally, a probabilistic set A on $\boldsymbol{X}$ is described by a defining function

$$A : \boldsymbol{X} \times \Omega \rightarrow \Omega_c \tag{7.4}$$

where A is the $(\boldsymbol{B}, \boldsymbol{B}_c)$—measurable function for any $x \in \boldsymbol{X}$. Here $(\Omega, \boldsymbol{B}, P)$ is a parameter space, $(\Omega_c, \boldsymbol{B}_c) =$ ([0,1], Borel sets) is a characteristic space and $\{m| \ m: \Omega \rightarrow \Omega_c \ (\boldsymbol{B}, \boldsymbol{B}_c)$ measurable function$\}$ is a family of characteristic variables.

From a formal point of view, a probabilistic set can be treated as a random field; to emphasize this fact, we use a notation A(x, ω).

With each $x \in \boldsymbol{X}$ one can associate a probability density function $p_x(u)$ defined over [0,1]. Alternatively, other probabilistic descriptors such as e.g., probability cumulative function $F_x(u)$ can be used.

Probabilistic sets come with an interesting moment analysis. For any fixed x, the mean value $E(A(x))$, vagueness(variance) $V(A(x))$, and the nth moment denoted as $M^n(A(x))$ are considered

$$
\begin{aligned}
E(A(x)) &= \int_0^1 p_x(u)du = M^1(A(x)) \\
V(A(x)) &= \int_0^1 (u - E(A(x)))^2 p_x(u)du \\
M^n(A(x)) &= \int_0^1 p_x^n(u)du
\end{aligned}
\tag{7.5}
$$

where $p_x(u)$ is a probability density function for a given element of the universe (x) of discourse. An interesting relationship holds for the moments

$$
M^1(A(x)) \geq M^2(A(x)) \geq M^3(A(x)) \geq \ldots \tag{7.6}
$$

which stresses that most of information (sound description) about A is confined only to a few lowest moments. This also implies that higher type information granules are not always justifiable and their practical relevance could be limited. Noticeable is fact that the mean $E(A)$ can be sought as a membership function used in fuzzy sets.

Assuming that the corresponding probabilistic characteristics are available, we can determine results for logic operations on probabilistic sets [1, 2]. For instance, let us determine the result for the union and intersection of two probabilistic sets A and B. For the cumulative probability functions of A and B, $F_{A(x)}$ and $F_{B(x)}$ we obtain

$$
F_{\max(A,B)(x)}(u) = F_{A(x)}(u)F_{B(x)}(u) \tag{7.7}
$$

and

$$
F_{\min(A,B)(x)}(u) = F_{A(x)}(u) + F_{B(x)}(u) - F_{A(x)}(u)F_{B(x)}(u) \tag{7.8}
$$

where $u \in [0,1]$.

7.6 Hybrid Models of Information Granules: Probabilistic and Fuzzy Set Information Granules

In a number of real-world situations, we describe concepts and form models using several formalisms of information granularity. These constructs become of particular interest when information granules have to capture a multifaceted nature of the problem. There are a number of interesting application-driven scenarios. The one that deserves attention here concerns a model involving probability and fuzzy sets. While from the very inception of fuzzy sets, there were quite vivid debates about linkages between these two formal vehicles of information granulation. While there was a claim that these two formalisms are the same, a prevailing and justifiable position is that probability and fuzzy sets are orthogonal by capturing two very distinct facets of reality. Let us stress that when dealing wit probabilities, we are concerned with an occurrence (or non-occurrence) of a certain event (typically described by some set). Fuzzy sets are not about occurrence but about a degree of membership to a certain concept. In the early studies by Zadeh [9], the aspect of orthogonality has been raised very clearly with the notions such as fuzzy events, probability of fuzzy events and alike. It is instructive to recall the main definitions introduced in the literature [9]. As usual, the probability space is viewed as a triple ($\boldsymbol{R}^n$, $\boldsymbol{B}$, P) where $\boldsymbol{B}$ is a the σ-field of Borel sets in $\boldsymbol{R}^n$ and P is a probability measure over $\boldsymbol{R}^n$. If $A \in \boldsymbol{B}$ then the probability of A, $P(A)$ can be expressed as $P(A) = \int_{\mathbf{R}^n} A(x)dP$ or using the characteristic function of A, one has $P(A) = \int_{\mathbf{R}^n} A(x)dP = E(A)$. In other words $P(A)$ is an expected value of the characteristic function of A.

The generalization comes in the form of a so-called fuzzy event A where A is a fuzzy set in $\boldsymbol{R}^n$ with a Borel measurable membership function.

The probability of a fuzzy event A is the Lebesgue-Stielties integral

$$P(A) = \int_{\boldsymbol{R}^n} A(x)dP = E(A) \tag{7.9}$$

The mean (expected value) of the fuzzy event A is

$$m(A) = \int_{\boldsymbol{R}^n} \boldsymbol{x}A(\boldsymbol{x})dP \tag{7.10}$$

The variance of A defined in $\boldsymbol{R}$ is given in the form

$$\mathrm{var}(A) = \int_{\boldsymbol{R}} (x - A(x))^2 dP \tag{7.11}$$

The generalization of the entropy of A defined in the finite space $\{x_1, x_2, \ldots, x_n\}$ with probabilities $p_1, p_2, \ldots, p_n$ comes in the form [9]

$$H(A) = -\sum_{i=1}^{n} A(x_i)p_i \log(p_i) \tag{7.12}$$

It is quite intuitive that probability of a fuzzy event and its specificity are interrelated. Considering that the specificity measure characterizes how detailed (specific) a certain piece of knowledge A is, it can be regarded as a decreasing function of a sigma count of the fuzzy event (assuming that A is a normal fuzzy set) such that $\mathrm{sp}(A) = 1$ for $A = \{x_0\}$ and $\mathrm{sp}(\boldsymbol{X}) = 0$ where $\boldsymbol{X}$ is the universe of discourse (space) over which information granules are formed. Along with the specificity, which we

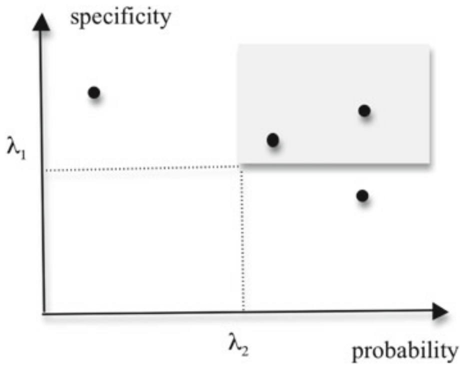


Fig. 7.11 Information granules described in terms of their specificity and probability; shown in a feasible region (shaded area) and visualized are several information granules satisfying and not satisfying the requirements identified by the thresholds

wish to see as high as possible (so that the piece of knowledge conveyed by A is highly meaningful), we also require that the probability of the fuzzy event is high enough. Imposing some threshold values on these two requirements, say λ_1 for the specificity and λ_2 for the probability we express the above requirements as follows: $\mathrm{sp}(A) > \lambda_1$ and $\mathrm{Prob}(A) > \lambda_2$, see Fig. 7.11 showing a region of feasible information granules. In this way, given an information granule A, one can easily check if it is feasible with regard to the two requirements presented above. Furthermore one can think of possible adjustments of information granule by decreasing its specificity in order to increase the probability or moving it around the universe of discourse to increase specificity and probability value.

Example 4 The concept *low* outdoor temperature is described with the use of a fuzzy set A whose membership function is shown in Fig. 7.12.

Considering that the probability of temperature in winter is described by the Gaussian distribution $N(x; -10, 5)$, determine the probability of this fuzzy event.

Following (7.9) the probability of this fuzzy event is computed as $P(A) = \int_{-40}^{5} A(x)p(x)dx = \int_{-40}^{-10} p(x)dx + \int_{-10}^{5} A(x)p(x)dx$

There are also some other interesting and practically viable linguistic descriptions in which we encounter so-called linguistic probabilities [10], say *low* probability, *high* probability, probability *about* 0.7, *very likely*, etc., that are applied to describe events or fuzzy events or events described by any other information granule.

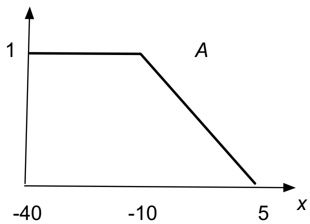


Fig. 7.12 Piecewise linear membership function of low outdoor temperature

The pieces of knowledge such as “it is *very high* probability that inflation rate will increase *sharply*” or “*high* error value occurs with a *very low* probability” are compelling examples of granular descriptors built with the use of probabilistic and fuzzy set information granules.

7.7 Conclusions

A wealth of formal frameworks of information granules is essential in the description and handling a diversity of real-world phenomena. We stressed the need behind the formation of information granules of higher order and higher type. When arranged together they are helpful to cope with the complexity of the descriptors. One has to be aware that although conceptually it is possible to develop a multilevel hierarchy of high order or/and high type of information granules, pursuing an excessive number of levels of the hierarchy could be counterproductive. Of course, one may argue that type-2 fuzzy sets could be extended (generalized) to type-3, type-4, etc. fuzzy sets however one has to be cognizant of the need to provide strong and compelling evidence behind successive levels of the hierarchy and the associated estimation costs of the determination of such hierarchically structured information granules.

The hybrid constructs form an interesting avenue to investigate as they invoke a slew of synergistic constructs, see e.g., Zhang [11]. While we showed the probabilistic—linguistic synergy, some other alternatives might be worth pursuing.

Problems

1. Fuzzy set A and the probability function are defined in some discrete space as shown in the following table

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
A	0.2	0.3	0.4	0.7	1.0	0.8	0.4	0.1	0.1
p	0.3	0.1	0.3	0.05	0.05	0.0	0.0	0.2	0.0

   (i) Determine the probability of the fuzzy event
   (ii) Determine the probability of the complement of this fuzzy event.

2. Provide an example of a type 2 fuzzy set
3. For interval-valued fuzzy sets A and B

$$A = [[0.4 0.7][0.1 0.6][0.0 0.8][0.5 0.7]]$$
$$B = [[0.1 0.5][0.9 1.0][0.1 0.5][0.6 0.6]]$$

   determine their union, intersection, and complement

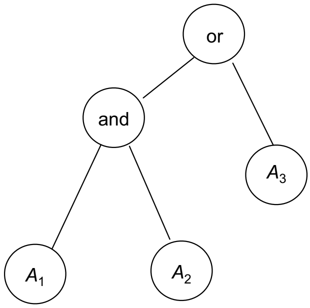


Fig. 7.13 Graph with interval-valued fuzzy sets

4. Compute the first two moments of the probabilistic set defined in $\{x_1, x_2, x_3, x_4\}$ whose membership grades are described by uniform probability density function $U[a, b]$ where the values of the parameters are given as follows

$$x_1 : U(0.4,\ 0.6)\, x_2 : U(0.1,\ 0.2)\, x_3 : U(0.5,\ 0.9)\, x_4 : U(0.4,\ 0.6)$$

5. What is the probability of the fuzzy events A and *very* (A) and more or less (A) where A is a triangular fuzzy set $A(x;\ 0.1,\ 1.0,\ 1.5)$ and probability density function is uniform $U(1.0, 1.5)$.
6. The temperature is described as a triangular fuzzy set $A(x) = 1 - x/20$ for $x \in [0, 20]$ and 0 otherwise.
   Determine the standard deviation of $p(x)$ of Gaussian distribution $N(x;\ 10, \sigma)$ so that $E(A)$ is no less than some threshold γ.
7. Given is a graph (tree), Fig. 7.13, with interval valued fuzzy sets A_1, A_2, and A_3 associated with its input nodes. What is the result of computing produced at the root of the graph. The membership functions are as follows

$$A_1(x) = \left[\exp(-(x-1)^2/3, \exp(-(x-1)^2/2\right]$$
$$A_2(x) = \left[\exp(-(x-2)^2/5, \exp(-(x-2)^2/3\right]$$
$$A_3(x) = \left[\exp(-(x-4)^2/6, \exp(-(x-1)^2/2\right]$$

References

1. Czogala, E., Pedrycz, W.: On the concept of fuzzy probabilistic controllers. Fuzzy Sets Syst. **10**, 109–121 (1983)
2. Czogala, E., Gottwald, S., Pedrycz, W.: Logical connectives of probabilistic sets. Fuzzy Sets Syst. **10**, 299–308 (1983)
3. Gorzałczany, M.B.: An interval-valued fuzzy inference method— Some basic properties. Fuzzy Sets Syst. **31**, 243–251 (1989)
4. Hirota, K.: Concepts of probabilistic sets. Fuzzy Sets Syst. **5**(1), 31–46 (1981)

5. Hirota, K., Pedrycz, W.: Characterization of fuzzy clustering algorithms in terms of entropy of probabilistic sets. Pattern Recogn. Lett. **2**(4), 213–216 (1984)
6. Hisdal, E.: The IF THEN ELSE statement and interval-valued fuzzy sets of higher type. Int. J. Man Mach. Stud. **15**, 385–455 (1981)
7. Karnik, N.M., Mendel, J.M., Liang, Q.: Type-2 fuzzy logic systems. IEEE Trans. Fuzzy Syst. **7**, 643–658 (1999)
8. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice Hall Upper Saddle River, NJ (2001)
9. Zadeh, L.A.: Probability measures of fuzzy events. J. Math. Anal. Appl. **23**, 421–427 (1968)
10. Zadeh, L.A.: Toward a perception-based theory of probabilistic reasoning with imprecise probabilities. J. Stat. Plann. Infer **105**, 233–264 (2002)
11. Zhang, Z.: On characterization of generalized interval type-2 fuzzy rough sets. Inf. Sci. **2013**(219), 124–150 (2013)

Chapter 8
User-Oriented Elicitation of Membership Functions

Abstract The issue of elicitation and interpretation of fuzzy sets (their membership functions) is of significant relevance from the conceptual, algorithmic, and application-oriented standpoints (Dubois and Prade in Theory and Applications. Academic Press, New York, 1980 [4]; Klir and Yuan in Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, Upper Saddle River, 1995 [6]; Nguyen and Walker in A First Course in Fuzzy Logic, CRC Press, Boca Raton, 1999 [11]). In the literature we can encounter a great deal of methods that support the construction of membership functions. In general, we distinguish a variety of methods positioned in-between *user-driven* and *data-driven* approaches with a number of techniques that share some features specific to both data- and user-driven techniques and hence being located somewhere in-between. The determination of membership functions has been a debatable issue for a long time almost since the very inception of fuzzy sets. In contrast to interval analysis and set theory where the estimation of bounds of the interval constructs has not attracted a great deal of attention and seemed to be somewhat taken for granted, an estimation of membership degrees (and membership functions, in general) became essential and over time has led us to sound, well justified and algorithmically appealing estimation techniques (Civanlar and Trussell in Fuzzy Sets Syst 18:1–13, 1986 [1]; Dishkant in Fuzzy Sets Syst 5:141–147, 1981 [2]; Dombi in Fuzzy Sets Syst 35:1–21, 1990 [3]; Hong and Lee in Fuzzy Sets Syst 84:389–404, 1996 [5]; Medaglia et al. in Eur J Oper Res 139:84–95, 2002 [9]; Turksen in Fuzzy Sets Syst 40:5–38, 1991 [15]).

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_8

8.1 Semantics of Fuzzy Sets: Some General Observations

Fuzzy sets are constructs that come with a well-defined meaning. They capture the semantics of the framework and its concepts 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 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 [10] 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 assumption holds. For instance, when describing linguistically quantified variables, say error or change of error, quantify temperature (*warm, hot, cold*, etc.) we may use seven generic concepts (descriptors) labeling them as positive *large*, positive *medium*, positive *small*, *around* zero, negative *small*, negative *medium*, 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. We can stress that the descriptive power of numbers is very much limited and numbers themselves are not used to abstract concepts. 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 such concepts we intend to establish to capture the essence of the domain knowledge. With the increase in the number of fuzzy sets, their semantics and interpretation capabilities become also negatively impacted. Fuzzy sets may be arranged into a hierarchy of terms (descriptors) but at each level of this hierarchy (when moving down towards higher specificity that is an increasing level of detail), the number of fuzzy sets is kept relatively low.

While fuzzy sets capture the semantics of the concepts, they may require some calibration depending upon 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, a term *low* temperature comes with a clear meaning yet it requires a certain calibration depending upon 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 (for instance, Saaty's approach, as discussed later in this chapter) as one of the quite visible and 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, which will help us appreciate the level and flexibility of fuzzy sets.

8.2 Fuzzy Set as a Descriptor of Feasible Solutions

In a commonly accepted sense, the optimization problem (maximization or minimization) of a certain objective function (performance index) f formulated in some space L where $L \subset \boldsymbol{R}^n$ returns a solution as a single element $\boldsymbol{x}_0 = \arg\max_{\mathrm{x} \in \mathrm{L}} f(x)$ or $\boldsymbol{x}_0 = \arg\max_{\mathrm{x} \in \mathrm{L}} f(x)$. From the practical perspective, an equally important is to answer a question: how to choose a suitable performance index reflecting the crux of the problem? How practically relevant is the optimal solution? What happens to the values of the objective function when a slight deviation from x_0 occurs? How could these slight deviations impact the quality of the obtained solution? The robustness of the result along with its quality comes as its comprehensive descriptor of the solution to the optimization problem. As an illustration, consider that the maximized one-dimensional objective function Q formed for the problem of interest which depends upon x as illustrated in Fig. 8.1a. For the same problem, formulated was another performance index $Q^{'}$ and its dependence on x is illustrated in Fig. 8.1b with the maximum observed at $x = x^{'}$.

Evidently the quality of x_0 and $x^{'}$ expressed in terms of Q and $Q^{'}$ is the same (the highest value of the corresponding objective functions). Both x_0 and $x^{'}$ are optimal (in terms of the corresponding criterion). When taking a global look at these solutions, there is a striking difference though: even small changes to x_0 come with a high detrimental behavior (a significant drop in the values of Q) whereas even substantial changes to x' do not result in any significant deterioration manifesting in the visibly lowered values of $Q^{'}$.

We advocate that the comprehensive knowledge about the quality of the solution can be conveniently conveyed in the form of a suitable fuzzy set of optimal

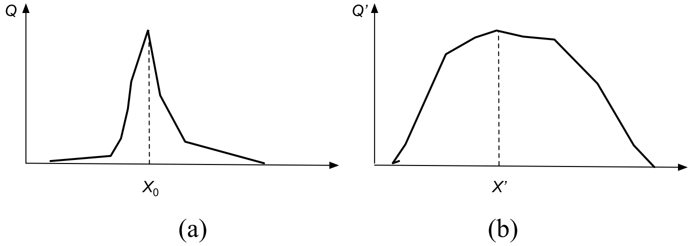


Fig. 8.1 Performance indexes Q (**a**) and $Q^{'}$ (**b**) as functions of some parameter under optimization

(feasible) solutions. The degree of membership is sough here as a degree of optimality (preference) of the solution.

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 L, that is f: $L \rightarrow \boldsymbol{R}^{+}$, where $L \subset \boldsymbol{R}$. Our intent is to determine its maximum, namely $x_0 = \arg\max_{x \in L} f(x)$. On a 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:

$$A(x) = \frac{f(x) - \min_{x \in L} f(x)}{\max_{x \in L} f(x) - \min_{x \in L} f(x)} \tag{8.1}$$

The boundary conditions are associated with values of minimum and maximum of f obtained over L. For other values of x where f attains is maximal value, $A(x)$ is equal 1 and around this point, the membership values become reduced. The shape of the membership function depends upon the nature of the function under consideration and as such it carries an important information about the behavior of the solution.

If the fuzzy set is used to quantify the quality (performance) of the solution to the minimization problem, then the resulting membership function is expressed as follows

$$A(x) = 1 - \frac{f(x) - \min_{x \in L} f(x)}{\max_{x \in L} f(x) - \min_{x \in L} f(x)} \tag{8.2}$$

As $f(\mathrm{x}_0)$ achieves its minimum at this point then $A(x_0)$ is equal to 1 as x_0 is the most preferred as a solution to this minimization problem.

If the function of interest assumes values in **R**, then these two formulas are modified by including the absolute values of the differences, that is $A(x) = \frac{|f(x) - \min_{x \in L} f(x)|}{|\max_{x \in L} f(x) - \min_{x \in L} f(x)|}$ and $A(x) = 1 - \frac{|f(x) - \min_{x \in L} f(x)|}{|\max_{x \in L} f(x) - \min_{x \in L} f(x)|}$.

Linearization is commonly encountered in a wealth of practical problems, say in control. Linearization, its quality, and description of such quality fall under the same banner as the above optimization problem. We show how the membership function could be formed in this case to reflect the quality of the solution. When linearizing a function around some predetermine point, a quality of the linearization scheme can be quantified in a form of some fuzzy set. Its membership function attains one 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). The following examples illustrate this idea.

Example 1 We are interested in the linearization of the function $y = g(x) = x^2$ around $x_0 = 1$ and assessing the quality of this linearization in the range $L = [0, 4]$.

The generic linearization formula reads as

$$y - y_0 = g'(x_0)(x - x_0)$$

where $y_0 = g(x_0)$ and g'(x_0) is the derivative of $g(x)$ at x_0. Given the form of the function under consideration, its linearized version reads as y-1 $= (2x_0)(x - x_0) = 2x - 1$. We define the quality of this linearization by taking the absolute value of the difference between the original function and its linearization, $f(x) = |g(x) + 1 - 2x)| = |x^2 - 2x + 1|$. Refer to Fig. 8.2 for the detailed plot of the function and its linearization.

As the fuzzy set A describes the quality of linearization (and we are concerned about the linearization error), its membership function has to take into consideration the expression

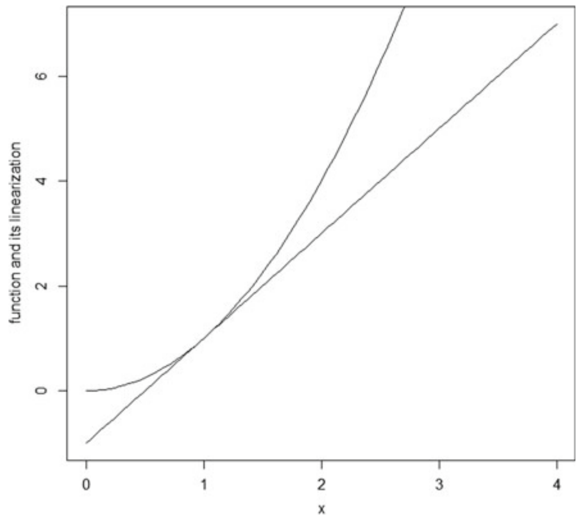


Fig. 8.2 Nonlinear function and its linearization

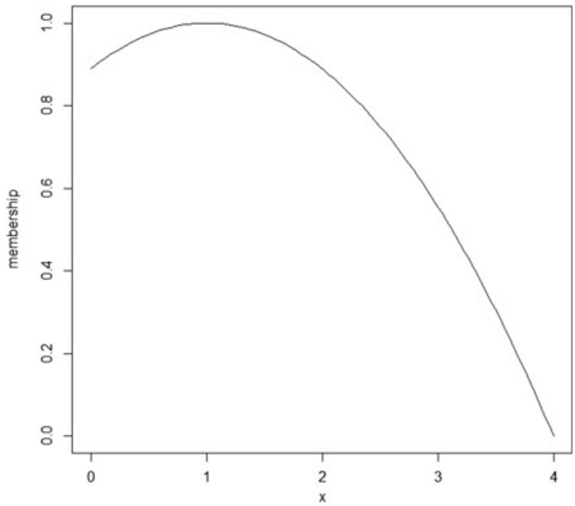


Fig. 8.3 Membership function describing the quality of linearization

$$A(x) = 1 - \frac{f(x) - \min_{x \in L} f(x)}{\max_{x \in L} f(x) - \min_{x \in L} f(x)} \tag{8.3}$$

The plot of the membership function is shown in Fig. 8.3.

The maximum and minimum of f over L is 16 and 0, respectively. Evidently, at $x = x_0$, a zero linearization error has been produced (ideal linearization at the linearization point). We note that the higher quality of approximation is achieved for the arguments positioned to the left from x_0.

Example 2 A simple electric circuit consisting of a voltage source with some internal resistance r and external resistance R assuming values ranging from 0 Ω to 100r Ω. It is well-known that the maximal power dissipated on R attains maximum for $R = r$.

The power dissipated on R is given as $Q = E^2R/(R + r)^2$. As we are concerned with the maximization of Q, the membership function of optimal external resistance A is shown in Fig. 8.4.

Apparently, the membership function of A is highly asymmetric. It is a quantification of an obvious fact: making the resistance a bit lower than the optimal value results in a rapid decline of the membership values, whereas making the resistor larger has a very limited impact on the decline of the membership values.

Example 3 Let us linearize the function $y = \sin(x)$ defined in $[0, \pi]$ around the linearization point $x_0 = \pi/2$.

The linearization formula reads as $A(x) = 1 - \frac{f(x) - \min_{x \in L} f(x)}{\max_{x \in L} f(x) - \min_{x \in L} f(x)}$ with $\min_{x \in L} f(x) = 0$, $\max_{x \in L} f(x) = 1$ with the minimum and maximum determined over $L = [0, \pi]$. The plot of the membership function is displayed in Fig. 8.5.

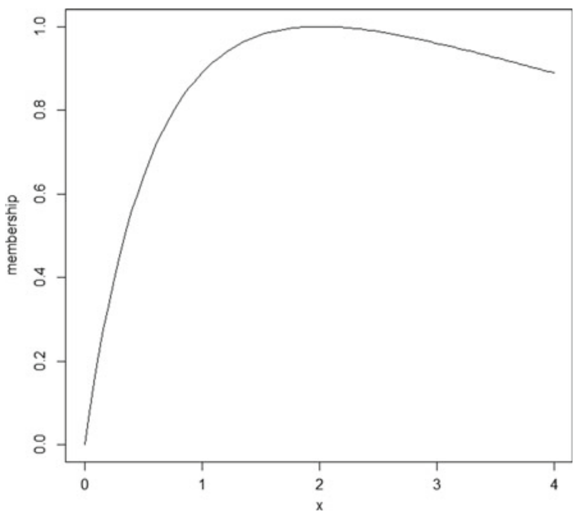


Fig. 8.4 Fuzzy set of optimal resistance R; $r = 2$ Ω, $E = 5$ V

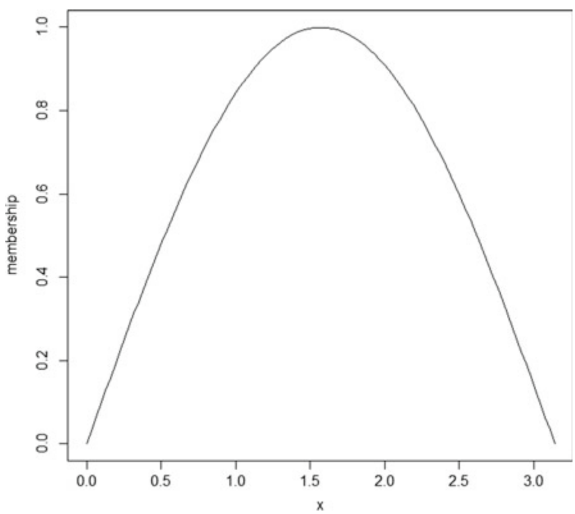


Fig. 8.5 Membership function of the optimal linearization error

8.3 Fuzzy Set as a Descriptor of the Notion of Typicality

Fuzzy sets address and quantify an issue of gradual *typicality* of elements to a given concept whose essence is being captured by the fuzzy set. 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 conveniently captured by adjusting the parameters of the membership function or choosing its form depending upon available experimental data. For instance, consider

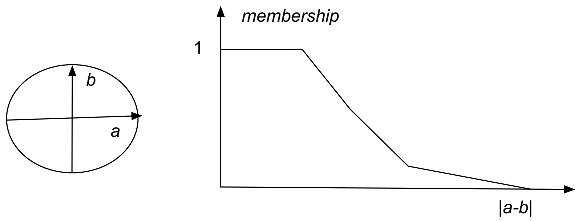


Fig. 8.6 Perception of geometry of ellipsoid and quantification of membership grades to the concept of *fuzzy* circle

a fuzzy set of circles. Formally, an ellipsoid includes a circular shape as its very special example which satisfies the condition of equal axes, that is $a = b$, see Fig. 8.6. What if we have $a = b + \varepsilon$ where ε is a very small positive number? Could this figure be sought as a circle? It is very likely so. Perhaps not a circle in a straight mathematical sense which we may note by assigning the membership grade that is very close to 1, say 0.97. Our perception, which comes with some level of tolerance to imprecision, does not allow us to tell apart this figure from the ideal circle.

It is intuitively appealing to see that higher differences between the values of the axes a and b result in lower values of the membership function. The definition of the fuzzy set of the concept of circle could be formed in a number of ways. Prior to the definition or even before a visualization of the shape of the membership function, it is important to formulate a universe of discourse over which it is to be defined. There are several sound alternatives worth considering:

(a) for each pair of values of the axes (a and b), collect an experimental assessment of membership of the ellipsoids to the category of circles. Here the membership function is defined over a Cartesian space of the spaces of lengths of axes of the ellipsoids. While selecting a form of the membership we require that it assumes values at $a = b$ and becomes gradually reduced when the arguments start getting more different.
(b) we can define an absolute distance between a and b, $|a - b|$ and form a fuzzy set over this space $\boldsymbol{X}$; $\boldsymbol{X} = \{x|x = |a - b|\}$ $\boldsymbol{X} \subset \boldsymbol{R}_+$. This semantic constraints translate into the condition of $A(0) = 1$. For higher values of x we may consider monotonically decreasing values of A.
(c) we can envision ratios of a and b $x = a/b$ and construct a fuzzy set over the space of $\boldsymbol{R}_+$ such that $\boldsymbol{X} = \{x|x = a/b\}$. Here we require that $A(1) = 1$. We also anticipate lower values of membership grades when moving to the left and to the right from $x = 1$. Note that the membership function could be asymmetric so we allow for different membership values for the same length of the sides, say $a = 6$, $b = 5$ and $a = 5$ and $b = 6$ (the effect could be quite apparent due to the occurrence of visual effects when perceiving geometric phenomena). The previous model of $\boldsymbol{X}$ as outlined in (a) cannot capture this effect.

Once the form of the membership function has been defined, it could be further adjusted by modifying the values of its parameters on a basis of some experimental findings. They come in the form of ordered triples or pairs, say (a_k, b_k, μ_k), or $(|a_k - b_k|, \mu_k)$, $k = 1, 2, \ldots, N$ with μ_k being the corresponding membership degree. Once these triples have been collected, one can estimate the parameters of the membership function depending on the previously accepted definition of the universe of discourse. The membership values of A are those available from the expert offering an assessment of the likeness of the corresponding geometric figure. Note that the resulting membership functions become formulated in different universes of discourse.

8.4 Vertical and Horizontal Schemes of Membership Function 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 $\boldsymbol{X}$ and request that an expert answers the following question

$$\text{does x belong to concept } A? \tag{8.4}$$

The answers are expected to come in a binary (yes-no) format. The concept A defined in $\boldsymbol{X}$ could be any linguistic notion, say *high* speed, *low* temperature, etc. Given N experts whose answers for a given point of $\boldsymbol{X}$ form a mixture of yes-no replies, we count the number of *yes* answers and compute the ratio of the positive answers (N') versus the total number of replies(N), that is N'/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 $\boldsymbol{X}$ requires collecting results for some other elements of $\boldsymbol{X}$ and determining the corresponding ratios as illustrated in Fig. 8.7 (observe a series of estimates that are determined for selected elements of $\boldsymbol{X}$; note also that the elements of $\boldsymbol{X}$ need not to be evenly distributed).

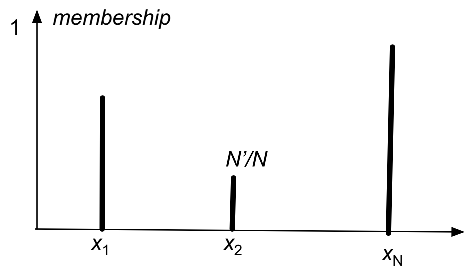


Fig. 8.7 A horizontal method of the estimation of the membership function

The advantage of the method comes with its simplicity as the technique relies explicitly upon a direct counting of responses. The concept is also intuitively appealing. The probabilistic nature of the replies helps us construct confidence intervals that are essential to the assessment of the specificity of the membership quantification. A certain drawback is related with 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 $\boldsymbol{X}$ is a subset of real numbers.

Example 4 The number of yes replies coming from a group of $N = 10$ experts is given in the tabular form below

x	x_1	x_2	x_3	x_4	x_5	x_6
N'	0	10	8	5	3	1

The membership function estimated in this way becomes [0 1 0.8 0.5 0.3 0.1].

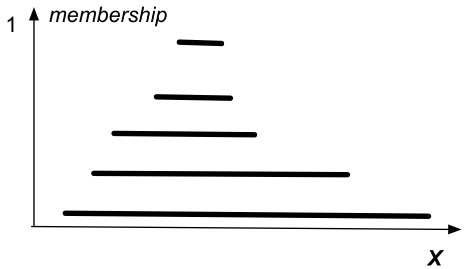
The vertical mode of membership estimation is concerned with the estimation of the membership function by focusing on the determination of the successive α-cuts. The experiment focuses on the unit interval of membership grades. The experts involved in the experiment are asked the questions of the following form

what are the elements of $\boldsymbol{X}$ which belong to fuzzy set A at degree not lower than? (8.5)

where α is a certain level (threshold) of membership grades in [0,1]. The essence of the method is illustrated in Fig. 8.8. 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 $\boldsymbol{X}$; 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.

These elements are identified by the expert as those forming the corresponding α-cuts of A. By repeating the process for several selected values of α we end up with

Fig. 8.8 A vertical approach of membership estimation through the reconstruction of a fuzzy set through its estimated α-cuts



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. 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.

The vertical and horizontal methods of membership estimation can be used together and their mutual usage could help check consistency of the results produced by the both methods. The following example illustrates the main idea.

Example 5 Consider the space $\boldsymbol{X} = \{a, b, c, d\}$. The vertical approach produces the following α-cuts

$$\alpha = 1.0\{c\}\,\alpha = 0.7\{b,\ c\}\,\alpha = 0.5\{b,\ c,\ d\}\alpha = 0.3\{b,\ c,\ d\}\alpha = 0.1\{a, b,\ c,\ d\}$$

What should be the membership grades produced by the horizontal method so that the results produced by the both methods produce consistent results.

To make the results consistent, the membership grades obtained through the horizontal method can be contained in the corresponding intervals. α = 1.0 implies that the membership grade of c determined by the horizontal method has to be 1 to make the result at this element of the universe of discourse consistent. The membership grade of b has to greater than 0.7 and lower than 1.0. Subsequently, for d, the membership grade could assume values positioned in the [0.5, 0.7) interval. For a the membership grade is in the [0.1, 0.3) interval.

8.5 Saaty's Priority Approach of Pairwise Membership Function Estimation

The priority approach introduced by Saaty [12–14] forms another interesting alternative used to estimate the membership function which help alleviate the limitations which are associated with the horizontal and vertical schemes of membership function estimation. To explain the essence of the method, let us consider a collection of elements $x_1, x_2, \ldots, x_n$ (those could be, for instance, some alternatives whose allocation to a certain fuzzy set is to be estimated) for which given are membership grades $A(x_1), A(x_2), \ldots, A(x_n)$. Let us organize them into a so-called reciprocal matrix of the following form

$$R = \left[r_{ij}\right] = \begin{bmatrix} \frac{A(x_1)}{A(x_1)} & \frac{A(x_1)}{A(x_2)} & \cdots & \frac{A(x_1)}{A(x_n)} \\ \frac{A(x_2)}{A(x_1)} & \frac{A(x_2)}{A(x_2)} & & \frac{A(x_2)}{A(x_n)} \\ \frac{A(x_n)}{A(x_1)} & \frac{A(x_n)}{A(x_2)} & & \frac{A(x_n)}{A(x_n)} \end{bmatrix} = \begin{bmatrix} 1 & \frac{A(x_1)}{A(x_2)} & \cdots & \frac{A(x_1)}{A(x_n)} \\ \frac{A(x_2)}{A(x_1)} & 1 & & \frac{A(x_2)}{A(x_n)} \\ \frac{A(x_n)}{A(x_1)} & \frac{A(x_n)}{A(x_2)} & & 1 \end{bmatrix} \tag{8.6}$$

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 $R(x_i, x_j) = 1/R(x_j, x_i)$. We will be referring to this form of reciprocity as *multiplicative reciprocity* as opposed to so-called *additive reciprocity* for which $R(x_i, x_j) + R(x_j, x_i) = 1$. Furthermore an important transitivity property holds that is $R(x_i, x_k)$ $R(x_k, x_{\mathrm{j}}) = R(x_i, x_j)$ for all triples of indexes *i, j*, and *k*. This property holds because of the way in which the matrix has been constructed. By plugging in the corresponding ratios, one obtains $R(x_i, x_k)\, R(x_k, x_j) = \frac{A(x_i)}{A(x_k)} \frac{A(x_k)}{A(x_j)} = \frac{A(x_i)}{A(x_j)}$. Let us now multiply the matrix by the vector of the membership grades $A = [A(x_1)\ \mathrm{A}(x_2) \ldots \mathrm{A}(x_n)]^{\mathrm{T}}$. For the *i*-th row of *R* (that is the *i*-th entry of the resulting vector of results) we obtain

$$[RA]_i = \begin{bmatrix} & \cdots & \\ \frac{A(x_i)}{A(x_1)} & \frac{A(x_i)}{A(x_2)} & \cdots & \frac{A(x_i)}{A(x_n)} \\ & \cdots & \end{bmatrix} \begin{bmatrix} A(x_1) \\ A(x_2) \\ \cdots \\ A(x_n) \end{bmatrix} \tag{8.7}$$

Thus the *i*-th element of the vector is equal to $nA(x_i)$. Overall once completing the calculations for all rows (i) this leads us to the expression $RA = nA$ or equivalently $(R - nI)A = 0$ with *I* being an identity matrix. 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 considerations, we have 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 5 or 9 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. The general quantification of preferences positioned on the scale of 1–7 can be described as in Table 8.1 [13, 14].

On the other hand, if x_j is preferred over x_i, the corresponding entry assumes values below one. Given the reciprocal nature 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 effort to complete pairwise evaluations is far more manageable in comparison to any

Table 8.1 Scale of intensities of relative importance

Intensity of relative importance	Description
1	Equal importance
3	Moderate importance of one element over another
5	Essential or strong importance
7	Demonstrated importance
9	Extreme importance
2, 4, 6, 8	Intermediate values between the two adjacent judgments

experimental overhead we need when assigning membership grades to all elements (alternatives) 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

$$\mathrm{v} = (\lambda_{\max} - n)/(n-1) \tag{8.8}$$

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 the re-examination of the experimental data and a re-run 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

$$D(i,k) = \left(r_{ik} - r_{ij}r_{jk}\right)^2 \tag{8.9}$$

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 $\sum_{i>k}^{n} V(i,k)$ becomes then a global indicator of the lack of transitivity of the experimental pairs of assessments.

Example 6 We consider an estimation of the membership of the fuzzy set $A = high$ temperature considering a space composed of temperatures 5, 10, 15, 20, 25, 30, 35 °C. The results of pairwise comparison are collected in the following reciprocal matrix R,

$$R = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{7} & \frac{1}{9} \\ 2 & 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{6} & \frac{1}{8} \\ 3 & 2 & 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 4 & 3 & 2 & 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{3} \\ 5 & 4 & 3 & 2 & 1 & \frac{1}{2} & \frac{1}{2} \\ 7 & 6 & 4 & 2 & 2 & 1 & \frac{1}{2} \\ & 9 & 8 & 5 & 3 & 2 & 1 \end{bmatrix}$$

The largest eigenvalue $\lambda_{\max}$ is equal 7.1 while the corresponding eigenvector is [0.06 0.09 0.14 0.24 0.36 0.51 0.73]. After its normalization, this yields the membership function of *high* temperature A = [0.08 0.12. 0.19 0.33 0.49 0.70 1.00].

Example 7 Now let us consider the following reciprocal matrix with the entries

$$R = \begin{bmatrix} 1 & 1/2 & 1/4 & 1/5 \\ 2 & 1 & 1/3 & 4 \\ 4 & 3 & 1 & 1/3 \\ 5 & 1/4 & 3 & 1 \end{bmatrix}$$

Now the maximal eigenvalue is far higher than the dimensionality of the problem, $\lambda_{\max} = 5.426$. In this case the resulting value of the inconsistency index is high, $\nu = (5.426 - 4)/3 = 0.475$ and hence there is no point to compute the corresponding eigenvector. To fix the inconsistency problem, we could have determined the lack of transitivity for the triples of indexes (i, j, k) and in this way highlight these assessments that tend to be highly inconsistent. These are the candidates whose evaluation has to be revised. Alternatively, one could reduce the length of the scale, say 1–3.

The method of pairwise comparison has been generalized in many different ways by allowing for estimates being expressed as fuzzy sets [8]. One can refer to a number of applications in which the technique of pairwise comparison has been directly applied [7].

8.6 Conclusions

The acquisition of membership functions forms a central design issue that is central to the analysis and synthesis of fuzzy-set based architectures. In virtue of the nature of fuzzy sets, their elicitation is inherently human-centric. The methods presented in this chapter are reflective of the human-centricity of the evaluation process. Fuzzy sets could be sought as a formalization and visualization of granular concepts implied by the problem at hand. One has to proceed with caution given the subjective facet of the elicitation process. The methods of horizontal and vertical membership estimation are straightforward however the results could easily exhibit some potential inconsistency, in particular in situations when the finite space comes with a large number of elements. The pairwise comparison method helps alleviate inconsistency by focusing the estimation on pairs of elements for which comparison process is carried out. Furthermore, the consistency can be quantified, which makes this approach suitable to build fuzzy sets.

Problems

1. The concept of sport car is conveniently described by a fuzzy set. Given is a collection of car makes presented in Fig. 8.9.
   Using the AHP method, determine the membership function; check the consistency of the completed pairwise comparison process.
2. Provide conditions making the result of horizonal method consistent with the results delivered by the vertical method in the form of α-cuts

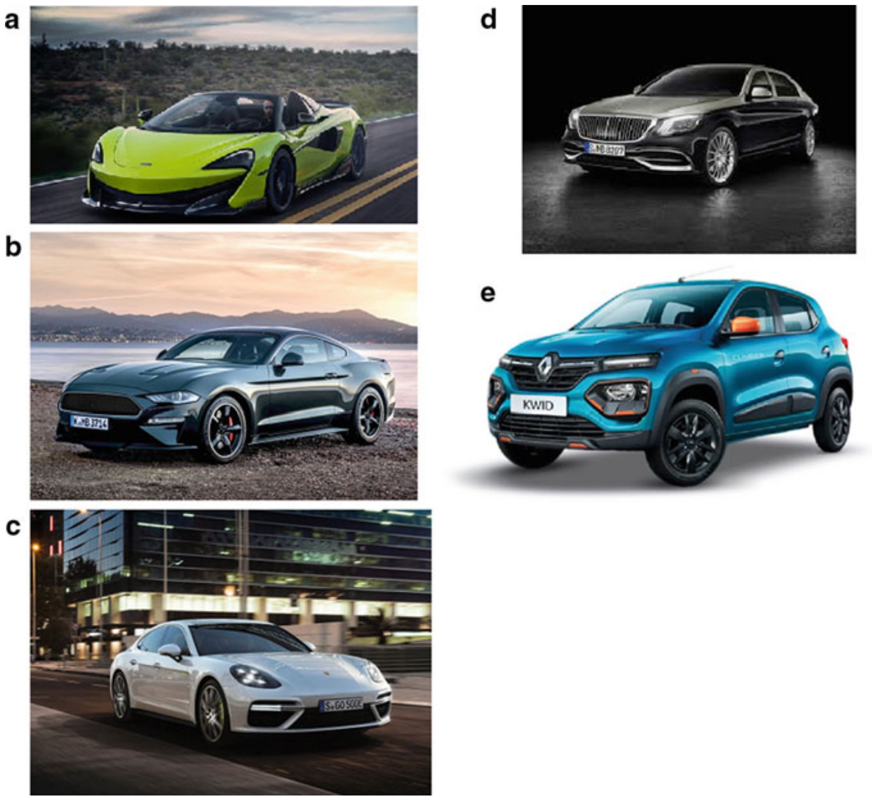


Fig. 8.9 A set of cars for which a fuzzy set of *sport* car is to be determined

$\alpha = 1.0$ $\{a, b\}$ $\alpha = 0.8$ $\{a, b, c\}$ $\alpha = 0.6$ $\{a, b, c, e, f\}$ $\alpha = 0.2$ $\{a, b, c, d, e, f\}$

3. Determine the membership function when using the horizontal method using results provided by 12 experts

x	x_1	x_2	x_3	x_4	x_5	x_6
N'	8	9	12	5	7	8

4. Given is a nonlinear function $f(x) = 4x^3 + 2x^2 - 3x$ for x in $[-10, 10]$. Characterize the quality of linearization procedure carried out at $x_0 = 0$.
5. Define a concept of equilateral triangle as a fuzzy set.

References

1. Civanlar, M., Trussell, H.: Constructing membership functions using statistical data. Fuzzy Sets Syst. **18**, 1–13 (1986)
2. Dishkant, H.: About membership functions estimation. Fuzzy Sets Syst. **5**, 141–147 (1981)
3. Dombi, J.: Membership function as an evaluation. Fuzzy Sets Syst. **35**, 1–21 (1990)
4. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
5. Hong, T., Lee, C.: Induction of fuzzy rules and membership functions from training examples. Fuzzy Sets Syst. **84**, 389–404 (1996)
6. Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice-Hall, Upper Saddle River (1995)
7. Kulak, O., Kahraman, C.: Fuzzy multi-attribute selection among transportation companies using axiomatic design and analytic hierarchy process. Inf. Sci. **170**, 191–210 (2005)
8. van Laarhoven, P., Pedrycz, W.: A fuzzy extension of Saaty's priority theory. Fuzzy Sets Syst. **11**, 199–227 (1983)
9. Medaglia, A., Fang, S., Nuttle, H., Wilson, J.: An efficient and flexible mechanism for constructing membership functions. Eur. J. Oper. Res. **139**, 84–95 (2002)
10. Miller, G.A.: The magical number seven plus or minus two: some limits of our capacity for processing information. Psychol. Rev. **63**, 81–97 (1956)
11. Nguyen, H., Walker, E.: A First Course in Fuzzy Logic. CRC Press, Boca Raton (1999)
12. Saaty, T.: The Analytic Hierarchy Process. McGraw Hill, New York (1980)
13. Saaty, T.: Scaling the membership functions. Eur. J. Oper. Res. **25**, 320–329 (1986)
14. Saaty, T. L.: Axiomatic foundation of the analytic hierarchy process. Manag. Sci. **32**(7) (1986)
15. Turksen, I.: Measurement of membership functions and their acquisition. Fuzzy Sets Syst. **40**, 5–38 (1991)

Chapter 9
Fuzzy Clustering

Abstract Clustering and fuzzy clustering can be analyzed, developed and used by having two perspectives in mind. On the one hand, these techniques create one of the generic tools of data analytics aimed at revealing a structure in data and are regarded as a essential prerequisite for a slew of detailed algorithms (Anderberg in Cluster analysis for applications. Academic Press, New York, 1973 [1]). There are numerous applications of clustering (Bezerra et al. in Inf Sci 518:13–28, 2020 [4]; D'Urso et al. in Spat Stat 30:71–102, 2019 [5]; Tao et al. in Neurocomputing, 2019 [10]) including Big Data (Ianni et al. in Future Gener Comput Syst 102:84–94, 2020 [7; Shukla and Muhuri in Eng Appl Artif Intell 77:268–282, 2019 [9]). On the other hand, in virtue of the underlying principle of clustering (group similar objects together), clusters can be sought as some information granules that offer an abstract, condensed view at the data. From this perspective, clustering can be sought as a conceptually sound and algorithmically appealing vehicle to construct fuzzy sets, viz. determine their membership grades (functions). Because the clustering process revolves about available data, we refer to this method of building fuzzy sets as a data-driven approach.

9.1 From Data to Clusters: Structure Discovery

Before proceeding with the formal formulation of the problem and pursuing algorithmic developments, it could be instructive to look at the rationale behind the clustering algorithms. Consider some examples of clouds of two-dimensional data

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_9

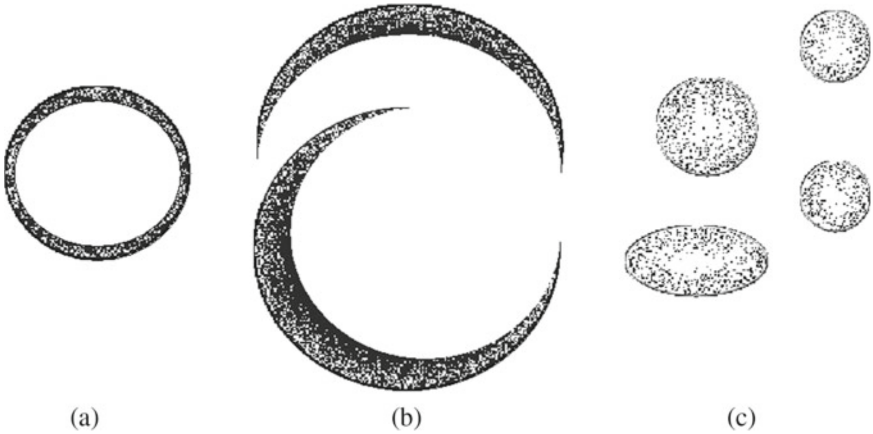


Fig. 9.1 Example data sets and revealed structure (clusters): **a** ring, **b** two moons, **c** spherical shapes

shown in Fig. 9.1. Being asked about finding a structure in these data, our answer is immediate—we effortlessly form groups of data that in our view have to be considered together. If being asked about of the rationale behind our decision, we might not provide an explicit answer but note that the formation of the structure is driven by the concept of closeness (resemblance).

As exemplified in Fig. 9.1, clusters exhibit a genuine diversity in terms of their shapes, for instance, Fig. 9.1a shows data that form a ring; two moon-like data are shown in Fig. 9.1b. Several circular and ellipsoidal shapes are the visible geometry of the data, Fig. 9.1c. Thus instead of dealing with hundreds or thousands of data, we just form and talk about a handful of clusters (groups). This offers an obvious advantage delivered to data analytics.

A central and intriguing question arises: how did we arrive at the discovery of the structure? Apparently, we followed an intuitively appealing rule: data points close to each other are put together. Thus a level of similarity between two data points is the reason why some points are grouped or not. The notion of similarity (closeness) becomes pivotal to all clustering algorithms.

Low resemblance means that the data are regarded as different and not considered as having anything in common. The terms resemblance, closeness, affinity are synonymous.

Formally, the idea of closeness becomes quantified by considering a concept of distance: the smaller the distance between data $\boldsymbol{x}$ and $\boldsymbol{y}$, the more eager we are willing to group these two items. And vice versa: large values of distances make the points unlikely to be placed into the same cluster. Furthermore we have some domain knowledge about the area the data came from and we implicitly engage into the grouping process.

9.2 The Clustering Problem: A Formal Formulation

Before delving into detailed algorithmic developments, it becomes instructive to set up a general framework of clustering algorithms and make it clear what results are going to be produced as the outcome of clustering, Fig. 9.2. Let us also assume that the data are composed of N n-dimensional data points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$, $\boldsymbol{x}_k \in \boldsymbol{R}^n$. The number of clusters is denoted by c and this number is fixed in advance.

Fundamentally, the results of clustering are conveyed in a form of a partition matrix U of dimensionality c by N. Commonly the representation of data and a description of clusters is conveyed through a family of c prototypes (centroids), $\boldsymbol{v}_1$, $\boldsymbol{v}_2, \ldots, \boldsymbol{v}_c$ which can be sought as the sound numeric representatives of the data. In many situations, they can be explicitly derived from the data and the already constructed partition matrix. The partition matrix U is defined as a c by N Boolean array with entries $u_{ik} \in \{0,1\}$ satisfying the following properties

$$0 < \sum_{k=1}^{N} u_{ik} < N \tag{9.1}$$

$$\sum_{i=1}^{c} u_{ik} = 1 \tag{9.2}$$

Let us denote by $\boldsymbol{U}$ a family of matrices satisfying these two requirements (9.1)–(9.2). The first requirement states that each cluster has to be nonempty and different from the entire set. The second requirement states that the sum of the membership grades should be confined to 1.

For example, the partition matrix describing three clusters determined for data $\boldsymbol{x}_1$, $\boldsymbol{x}_2, \ldots, \boldsymbol{x}_6\}$ is given in the following form,

$$U = \begin{bmatrix} 010110 \\ 101000 \\ 000001 \end{bmatrix}$$

The partition matrix conveys a complete structure about the data. Cluster 1 (the first row of the partition matrix) is composed of data $\boldsymbol{x}_2$, $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$. Cluster 2 consists of data $\boldsymbol{x}_1$ and $\boldsymbol{x}_3$ and cluster 3 includes $\boldsymbol{x}_6$. The partition matrix is Boolean: its

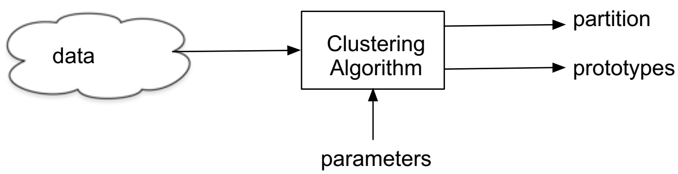


Fig. 9.2 From data to clustering algorithm and its results

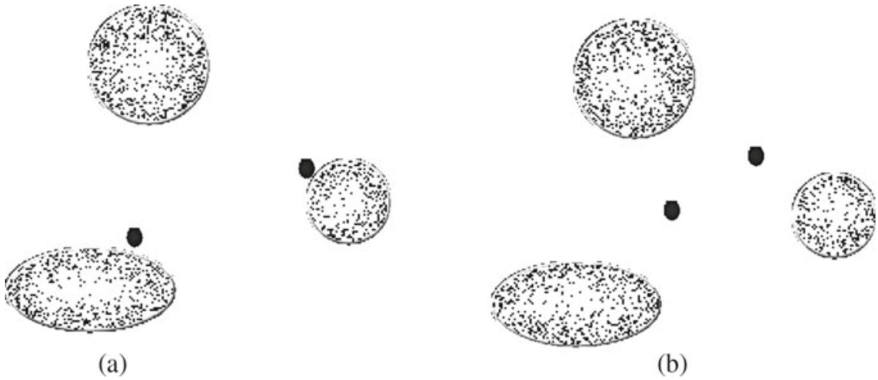


Fig. 9.3 Boolean (**a**) and fuzzy (**b**) nature of two data points

entries are 0 or 1. The value of 1 indicates that $\boldsymbol{x}_k$ belongs to cluster i. We can relax this requirement to cope with commonly encountered real-world scenarios where a concept of partial membership occurs quite commonly. In Fig. 9.3, we contrast the binary nature of clustering, Fig. 9.3a, with a situation illustrated in Fig. 9.3b.

Here the belongingness of the two points (shown as black dots) is obvious as to their full assignment (or exclusion) to each cluster. The location of the two data point in Fig. 9.3b is such that their membership to the clusters is difficult to quantify this observation and gain a certain level of comfort in making this assignment. We would be inclined to associate membership degrees in-between 0 and 1. By doing so, we end up with a fuzzy partition matrix with the entries located in [0,1]. Formally a fuzzy partition matrix satisfies the requirements (9.1)–(9.2) as already stated but $u_{ik} \in [0,1]$.

9.3 The Fuzzy C-Means Algorithm

In what follows, we proceed with the detailed description of the Fuzzy C-Means (FCM) clustering algorithm [2, 8]. It is a typical example of a so-called objective function clustering. The clustering problem boils down to a minimization of a properly formed objective function Q being a function of the parameters of the clusters, viz the prototypes and a partition matrix. The minimization is achieved by modifying the values of these parameters. In a nutshell, the optimization problem is formalized as

$$\mathrm{Min}\, Q(U; \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c) \tag{9.3}$$

where the minimization is completed with respect to the partition matrix $U \in \boldsymbol{U}$ and the prototypes $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c$.

Given a collection of n-dimensional data set $\{\boldsymbol{x}_k\}$, $k = 1,2,\ldots,N$, the objective function (performance index) Q is expressed as a double sum of the squared distances with the summation taken over data and clusters

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 \tag{9.4}$$

u_{ik} is the membership degree of data $\mathbf{x}_\mathrm{k}$ to the i-th cluster. The distance between the data $\boldsymbol{x}_k$ and prototype $\boldsymbol{v}_i$ is denoted by ||.||. The fuzzification coefficient m (>1.0) expresses the impact of the membership grades on the individual clusters and implies a shape of membership functions.

The minimization of Q completed with respect to $U \in \mathbf{U}$ and the collection of prototypes of the clusters $\boldsymbol{v}_i$, $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c\}$. More explicitly, we write it down as follows

$$\begin{array}{cc} \min & Q \\ \text{with respect to } U \in \boldsymbol{U}, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c \in \boldsymbol{R}^n & \end{array} \tag{9.5}$$

From the optimization standpoint, there are two individual optimization tasks to be carried out separately for the partition matrix and the prototypes. The first one concerns the minimization with respect to the constraints given the requirement of the form (9.2), which holds for each data point $\boldsymbol{x}_k$. The use of Lagrange multipliers transforms the problem with constraint into its constraint-free version. We form an augmented objective function formulated for each data point, $k = 1, 2, \ldots, N$, which reads as

$$V = \sum_{i=1}^{c} u_{ik}^{m} d_{ik}^{2} + \lambda \left(\sum_{i=1}^{c} u_{ik} - 1 \right) \tag{9.6}$$

where $d_{ik}^2 = \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2$. The necessary conditions for achieving the minimum of V for $k = 1, 2, \ldots N$, one has

$$\begin{gathered} \frac{\partial V}{\partial u_{st}} = 0 \\ \frac{\partial V}{\partial \lambda} = 0 \end{gathered} \tag{9.7}$$

$s = 1, 2 \ldots c$, $t = 1, 2 \ldots N$. Now we calculate the partial derivative of V with respect to the elements of the partition matrix in the following way

$$\frac{\partial V}{\partial u_{st}} = m u_{st}^{m-1} d_{st}^{2} + \lambda \tag{9.8}$$

Setting $\frac{\partial V}{\partial u_{st}} = 0$, we calculate the membership grade u_{st} to be equal to

$$u_{st} = -\left(\frac{\lambda}{m}\right)^{1/(m-1)} d_{st}^{2/(m-1)} \tag{9.9}$$

Given the requirement $\sum_{j=1}^{c} u_{jt} = 1$ and plugging it into (9.9) one has

$$-\left(\frac{\lambda}{m}\right)^{1/(m-1)} \sum_{j=1}^{c} d_{jt}^{2/(m-1)} = 1 \tag{9.10}$$

We complete some re-arrangements of the above expression by isolating the term including the Lagrange multiplier λ,

$$-\left(\frac{\lambda}{m}\right)^{1/(m-1)} = \frac{1}{\sum_{j=1}^{c} d_{jt}^{2/(m-1)}} \tag{9.11}$$

Inserting this expression into (9.9), we obtain the corresponding entries of the partition matrix

$$u_{st} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{st}^2}{d_{jt}^2}\right)^{1/(m-1)}} \tag{9.12}$$

Going back to the detailed notation, one has the following expression to determine the entries of the partition matrix

$$A_i(\boldsymbol{x}_k) = \frac{1}{\sum_{j=1}^{c} \left(\frac{||\boldsymbol{x}_k - \mathbf{v}_i||^2}{||\boldsymbol{x}_k - \boldsymbol{v}_j||^2}\right)^{1/(m-1)}} \tag{9.13}$$

The second part of the minimization of Q is about determining the prototypes. To carry out this minimization, one has to specify the distance ||.|| and the ensuing calculations are impacted by this selection. For the time being, we consider the Euclidean distance that is $\|\boldsymbol{x}_\mathrm{k} - \boldsymbol{v}_i\|^2 = \sum_{j=1}^{n} \left(x_{kj} - v_{ij}\right)^2$. The objective function reads now as

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m \sum_{j=1}^{n} (x_{kj} - v_{ij})^2 \tag{9.14}$$

and its gradient determined with respect to $\boldsymbol{v}_i$ and made equal to zero yields the following expression

$$\sum_{k=1}^{N} u_{ik}^m (x_{kt} - v_{st}), s = 1,\ 2, \ldots, c; \quad t = 1, 2, \ldots, n \tag{9.15}$$

Thus in a concise form, the prototypes are computed as

$$\boldsymbol{v}_i = \sum\nolimits_{k=1}^{N} u_{ik}^m \boldsymbol{x}_k / \sum\nolimits_{k=1}^{N} u_{ik}^m \tag{9.16}$$

The weighted Euclidean distance can be considered, namely $\sum_{j=1}^{n}(x_{kj} - v_{ij})^2/\sigma_j^2$ where σ_j^2 is a variance of the j-th variable. In this situation, the calculations are carried out in the same way.

One should emphasize that the use of some other distance different from the Euclidean one brings some computational complexity and the formula for the prototype cannot be computed as shown above. We will get to this problem in Sect. 9.8.

Clustering is a representative example of so-called unsupervised learning—given the data, their structure is revealed. There is no mechanism of supervision as typically encountered in supervised learning conveyed by a training data, viz. a pair of input-output pairs and the objective is to learn a mapping between elements of the input-output pairs. The mappings are referred to as classifiers (with a finite, usually small number of labels specified in the output space) or regressors (when the output space is a subset of real numbers $\boldsymbol{R}$).

It becomes evident that the computing of the partition matrix and the prototypes is intertwined: to calculate the partition matrix, we need the prototype and vice versa. Overall, the FCM clustering is completed through a sequence of iterations involving (9.13) and (9.16) where we start from some random allocation of data to clusters (a certain randomly initialized partition matrix) and carry out the following updates by adjusting the values of the partition matrix and the prototypes. In summary, the overall procedure is outlined as follows

- randomly initialize partition matrix U

  repeat

  - update prototypes using (9.16)
  - update partition matrix (9.13)

  until a certain termination criterion has been satisfied

The iterative process is terminated once a certain termination criterion has been satisfied. Typically, the termination condition is formed by looking at the changes in the membership values of the successive partition matrices. Denote by $U(t)$ and $U(t+1)$ the two partition matrices produced in the two consecutive iterations of the algorithm, namely t and $t+1$. If the distance $||U(t+1) - U(t)||$ is les than some small predefined threshold ε, then we terminate the algorithm. Here, one considers the Tchebyschev distance between the partition matrices meaning that the termination criterion reads as follows

$$\max_{i,k} |u_{ik}(t+1) - u_{ik}(t)| \leq \varepsilon \tag{9.17}$$

The membership functions offer an interesting feature of evaluating an extent to which a certain data point is shared between different clusters and in this sense become difficult to allocate to a single cluster (fuzzy set). We can consider an entropy measure h,

$$h(\boldsymbol{x}_k) = \sum_{i=1}^{c} h(u_{ik}) \tag{9.18}$$

which delivers a self-flagging mechanism—the data points of high value of entropy $h(\boldsymbol{x}_k)$ may require a thorough inspection.

It is worth stressing that the membership functions generated by the FCM are indeed described analytically. While the partition matrix U delivers the membership grades for the discrete data points, the analytical formula comes from the derivations completed above. For any $\boldsymbol{x}$ in $\boldsymbol{R}^n$ there are c membership functions of fuzzy sets A_1, A_2, …, A_c described as follows

$$A_i(\boldsymbol{x}) = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|\boldsymbol{x}-\mathbf{v}_i\|^2}{\|\boldsymbol{x}-\boldsymbol{v}_j\|^2} \right)^{1/(m-1)}} \tag{9.19}$$

Example 1 We consider a two-dimensional example of data composed of 10 points, ses Fig. 9.4. Considering $c = 2$ and starting from a random initialization of the partition matrix, the values of the minimized objective function decrease rapidly at some initial iterations of the method, Fig. 9.5.

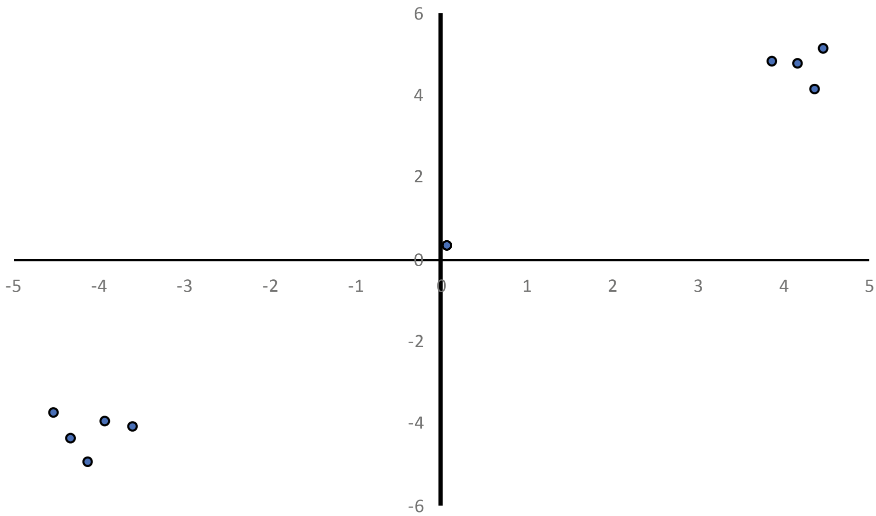


Fig. 9.4 Two-dimensional synthetic data

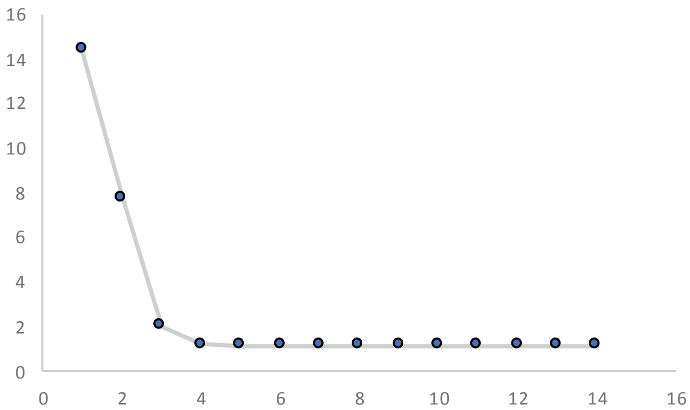


Fig. 9.5 The values of the objective function Q in successive iterations of the algorithm

The obtained partition matrix U has the following entries

$$U = \begin{bmatrix} 0.002 & 0.005 & 0.001 & 0.002 & 0.480 & 0.999 & 0.994 & 0.998 & 0.999 & 0.999 \\ 0.998 & 0.995 & 0.999 & 0.998 & 0.520 & 0.001 & 0.006 & 0.002 & 0.001 & 0.001 \end{bmatrix}$$

It is apparent that while two highly condensed clusters are revealed (the corresponding values of U are close to 1 or 0), the fifth data point is flagged as its membership grades are 0.48 and 0.52—these membership grades identify this point as being localized in-between the two compact clusters; as such it might require more inspection as a potential outlier or might be associated with some unusual behavior of the process governing the data. Both the partition matrix and the prototypes $\boldsymbol{v}_1 = [-3.89\ -4.05]$, $\boldsymbol{v}_2 = [3.98\ 4.39]$ deliver a sound description of the structure present in the data.

9.4 Main Parameters of the FCM Algorithm

The FCM algorithm comes with a collection of parameters whose values need to be specified prior to the execution of the method. We discuss the role and an impact on the results being produced.

Number of clusters (c) is associated with the structure in the data set and the number of fuzzy sets estimated by the method; the increase in the number of clusters produces lower values of the objective function. In limit, when $c = N$, this yields $Q = 0$. However, given the semantics of fuzzy sets one should maintain this number quite low (5–9 information granules).

Distance reflects (or imposes) a geometry of the clusters one is looking for; it is an essential design parameter affecting the shape (geometry) of the produced

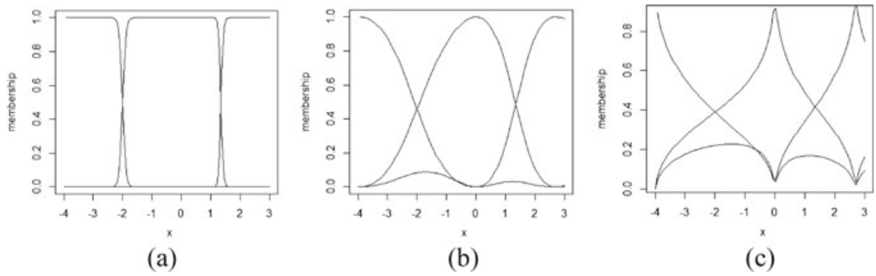


Fig. 9.6 Plots of membership functions for selected values of m: **a** $m = 1.1$, **b** $m = 2.0$, **c** $m = 3.5$. The prototypes are $v_1 = -4$, $v_2 = 0.0$; $v_3 = 2.7$

membership functions. It has to be selected in advance before proceeding with the execution of the algorithm.

Fuzzification coefficient implies a certain shape of membership functions present in the partition matrix; essential design parameter; see Fig. 9.6. Low values of m (being close to 1.0) induce characteristic function. Note that when m tends to 1, the method translates into the K-Means clustering and the produced partition matrix is of a Boolean nature (with 0–1 entries only). The value $m = 2$ is commonly used; the membership functions resemble Gaussian functions. The values of m higher than 2.0 yield spiky membership functions.

The plots of the membership functions obtained for various values of the fuzzification coefficient are shown in Fig. 9.6.

In addition to the varying shape of the membership functions, we observe that the requirement put on the sum of membership grades imposed on the fuzzy sets yields some multimodality. The membership functions are not unimodal but may exhibit some rippling effect whose intensity depends upon the distribution of the prototypes and the values of the fuzzification coefficient. The intensity of the rippling effect is also affected by the values of m and increases with the higher values of this parameter.

Termination criterion is formed by computing the distance between partition matrices obtained in the two successive iterations; the algorithm terminated once the distance below some assumed positive threshold (ε). Alternatively, one can monitor the changes in the values of $Q(iter)$, $Q(iter + 1)$ and terminate computing once there are no visible changes in the values of the objective function.

Example 1 Is A_i a normal fuzzy set? Determine its support.

A_i is a normal fuzzy set, $hgt(A_i) = 1$ and the maximal value of the membership function occurs at $\boldsymbol{x} = \boldsymbol{v}_i$. The support of A_i is $\boldsymbol{R}^n - \{\boldsymbol{v}_1, \ldots \boldsymbol{v}_{i-1}, \boldsymbol{v}_{i+1}, \ldots, \boldsymbol{v}_c\}$

Example 2 Given are three examples of two-dimensional data shown in Fig. 9.7.

When considering FCM with the Euclidean distance, how many clusters could you suggest to use. Justify your choice.

The Euclidean distance favours a spherical geometry of the data. Such spheres can be regarded as patches using which we tend to “cover” the data. Having this in

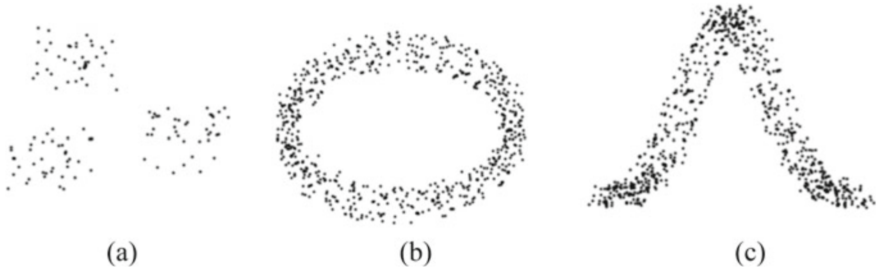


Fig. 9.7 Examples of two-dimensional data

mind, In Fig. 9.7a three clusters could be considered while for Fig. 9.7b, c a larger number of clusters is advisable.

9.5 Construction of One-Dimensional Fuzzy Sets

Fuzzy sets produced by the FCM algorithm are typically defined in the n-dimensional space of real numbers. Our previous investigations were mostly focused on one dimensional space over which fuzzy sets are defined. We can take the same position here by treating the prototypes as elements in $\boldsymbol{R}^n$ and projecting them on the individual coordinates, see Fig. 9.8.

Consider that the projection of the prototypes $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c$ over the jth coordinate results in the one-dimensional prototypes $v_{1j}, v_{2j}, \ldots, v_{cj}$. The corresponding membership functions $A_{1j}, A_{2j}, \ldots, A_{cj}$ are computed on their basis following the formula (9.13) applied to one-dimensional arguments, namely

$$A_{ij}\left(x_j\right) = \frac{1}{\sum_{l=1}^{c}\left(\frac{(x_j - v_{ij})^2}{(x_j - v_{lj})^2}\right)^{1/(m-1)}} \tag{9.20}$$

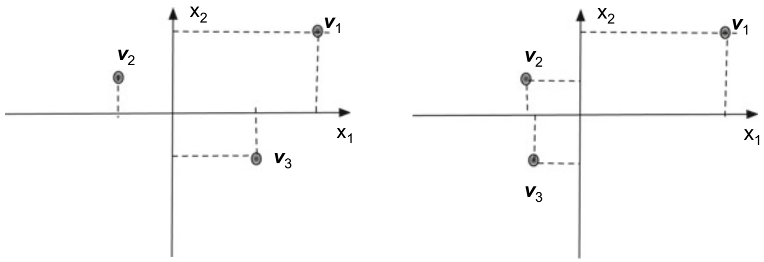


Fig. 9.8 Prototypes in the two-dimensional space $\boldsymbol{R}^2$ along with their and their projections leading to the definition of fuzzy sets over real lines

$i = 1, 2, \ldots, c; j = 1, 2, \ldots, n$. Again, this result clearly emphasis the role of fuzzy clustering—the membership functions are built automatically and their analytical formula is provided.

Recalling the idea of a family of referential fuzzy sets defined over a certain space, in particular, the distinguishability criterion, one can characterize the satisfaction of this criterion.

The projection of the prototypes on the individual variables reveals some interesting dependencies with this regard. Let us consider some illustrative examples as shown in Fig. 9.8. In Fig. 9.8a the projected prototypes are clearly distinguishable both in the x_1 and x_2 variable. In Fig. 9.8b the prototypes projected on x_2 are indistinguishable—the values of the corresponding projected prototypes v_{21} and v_{23} are very close to each other. In other words, the number of semantically sound fuzzy sets is reduced to two; the projection of the first and the second cluster is collapsed.

9.6 Structural View of the Clusters

The structural relationships formed by the FCM algorithm are quantified by taking a data view and a cluster viev.

Data view. the data view is concerned with expressing closeness relationship between any two data points. This dependency is described in the form of a proximity matrix P. The proximity matrix of dimensionality N by N, $P = [p_{kl}]$ is defined as follows

$$p_{kl} = \sum_{i=1}^{c} \min(u_{ik}, u_{il}) \tag{9.21}$$

$k, l = 1, 2, \ldots, N$. In virtue of the definition of the partition matrix, $p_{kk} = 1$ and $p_{kl} = p_{lk}$ (symmetry). The higher the proximity value, the closer the corresponding data (k, l) are.

Cluster view The cluster view is established at the level of clusters and expresses a level of strength of association (linkage) between any two clusters. For clusters i and j, the association is computed as the cosine similarity measure based on the vectors of membership grades of $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ in the following way

$$assoc_{ij} = \cos\left(\boldsymbol{u}_i, \boldsymbol{u}_j\right) = \frac{\sum_{k=1}^{N} u_{ik} u_{jk}}{\sqrt{\sum_{k=1}^{N} u_{ik}^2} \sqrt{\sum_{k=1}^{N} u_{jk}^2}} \tag{9.22}$$

The joint information about the structure conveyed by the prototypes of the clusters and the cluster view is arranged together in the form of an attributed graph, Fig. 9.9. This graph has c nodes associated with the corresponding prototypes. The edges are quantified by the values of the strength of association (9.22).

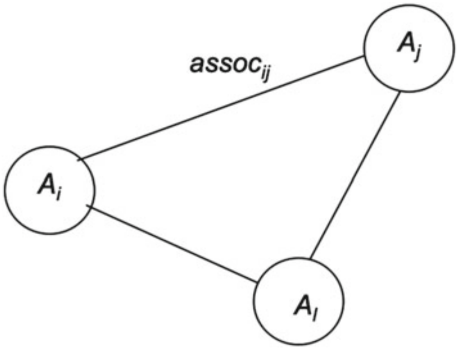


Fig. 9.9 Attributed graph displaying the structural content of data conveyed by clusters

Example 3 An incomplete partition matrix U is given below

$$U = \begin{bmatrix} 0.2\ 0.9\ 0.0\ 0.5 \\ 0.4\ 0.0\ 0.4\ 0.0 \\ ?\ 0.0\ ?\ 0.3 \\ 0.3\ 0.1\ 0.2\ ? \end{bmatrix}$$

Determine the missing entries of the matrix (denoted by ?).

As the sum of entries in each column should be equal to 1, we have $u_{31} = 0.1$. $u_{33} = 0.4$ and $u_{44} = 0.2$.

9.7 Internal and External Measures of Quality of Fuzzy Clusters

Clustering methods are challenging as algorithms of unsupervised learning hence the question about the quality of the clustering results remains open to a significant extent. The topic has been studied and several performance measures have been proposed. The taxonomy of the performance measures is outlined in Fig. 9.10.

The external measures are about quantifying the performance of the clustering results used in the constructs such as classifiers and predictors. The reconstruction error (which will be discussed in Chap. 10 in the context of a so-called granulation-degranulation tandem or fuzzification-defuzzification) is another measure that describes the performance of the constructed clusters.

The internal measures are predominantly concerned with cluster validity indexes They are focused on the determination of the number of clusters. The minimal or maximal value of the index (depending upon the underlying formula) is the one which points at the preferred number of clusters. Two main categories are identified here, namely

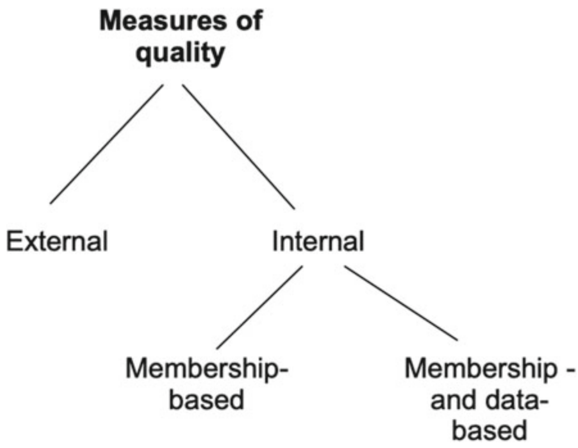


Fig. 9.10 A taxonomy of performance measures of quality of fuzzy clusters

(i) based on membership grades. Some examples in this category include partition coefficient PC [3]

$$PC = \frac{1}{N}\sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2 \tag{9.23}$$

and partition entropy PE [2]

$$PE = \frac{1}{N}\sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}\log_a u_{ik} \tag{9.24}$$

(a-the base of the logarithm).

(ii) based on membership grades and data involved in clustering. Here some representative examples include a Fukuyama and Sugeno index [6]

$$\begin{aligned} FS &= \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2 - \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2 ||\mathbf{v}_i - \overline{\mathbf{v}}||^2 \\ \overline{\mathbf{v}} &= \sum_{i=1}^{c} \mathbf{v}_i \end{aligned} \tag{9.25}$$

and Xie and Beni index [12]

$$XB = \frac{\frac{1}{N}\sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2}{\min_{\substack{i,j \\ i\neq j}} \left\|\mathbf{v}_i - \mathbf{v}_j\right\|^2}. \tag{9.26}$$

In (9.26) the nominator quantifies the compactness of the clusters (note this is the original objective function) whereas the denominator is concerned with the compactness of the clusters (distances between the prototypes).

They have been a number of proposals of cluster validity indexes, refer to the survey paper for a detailed discussion [11]. As each validity index comes with its own rationale, it is not surprising that each index points at the different values of clusters. The experimental studies strongly support this view.

9.8 Distance Functions in the FCM Algorithm

The commonly used distance function is the Euclidean one. This choice is motivated by the two commonly advocated factors: (i) omnipresence of normally distributed data, and (ii) computational convenience.

Distances other than the Euclidean one can be used however this leads to the more demanding computing when it comes to the optimization of the prototypes. The partition matrix U is determined in the same way as before; the selection of the distance does not impact the complexity of calculations of the partition matrix.

The Minkowski distance (L_p distance) forms a general class of distances; its underlying formula reads as follows

$$L_p(\boldsymbol{x}, \boldsymbol{y}) = \sqrt[p]{\sum_{j=1}^{n} \left|x_j - y_j\right|^p} \tag{9.27}$$

$\boldsymbol{x}$ and $\boldsymbol{y}$ are located in an n-dimensional space of real numbers, where $p > 1$ is a parameter bringing a significant level of flexibility. If $p = 1$ we have a so-called Hamming distance, $p = 2$ generates the Euclidean distance, whereas $p = \infty$ entails the Tchebyshev distance.

Each distance comes with a clear geometric interpretation. To get a better insight into the geometry associated with various distances, we determine a position of all points in $\boldsymbol{R}^2$ that are located at the same distance from the origin. The corresponding plots are contained in Fig. 9.11. There is a visible impact of the values of p on the shape of the generated surfaces: for $p = 1$ one has a diamond-like contours, $p = 2$ generates a collection of concentric circles while the Tchebyshev distance results in box-like shapes.

The optimization of the prototypes is straightforward in case of the Euclidean distance; we derived the detailed closed-form formula (9.16). In all remaining cases, one has to engage in the iterative optimization process in which the i-th prototype is adjusted iteratively following the scheme

$$\boldsymbol{v}_i(iter + 1) = \boldsymbol{v}_i(iter) - a\nabla_{\boldsymbol{v}_i} Q \tag{9.28}$$

α is the learning rate, $\alpha > 0$, $i = 1, 2, \ldots, c$. The calculations of the gradient for the Hamming distance (L_1) proceed as follows

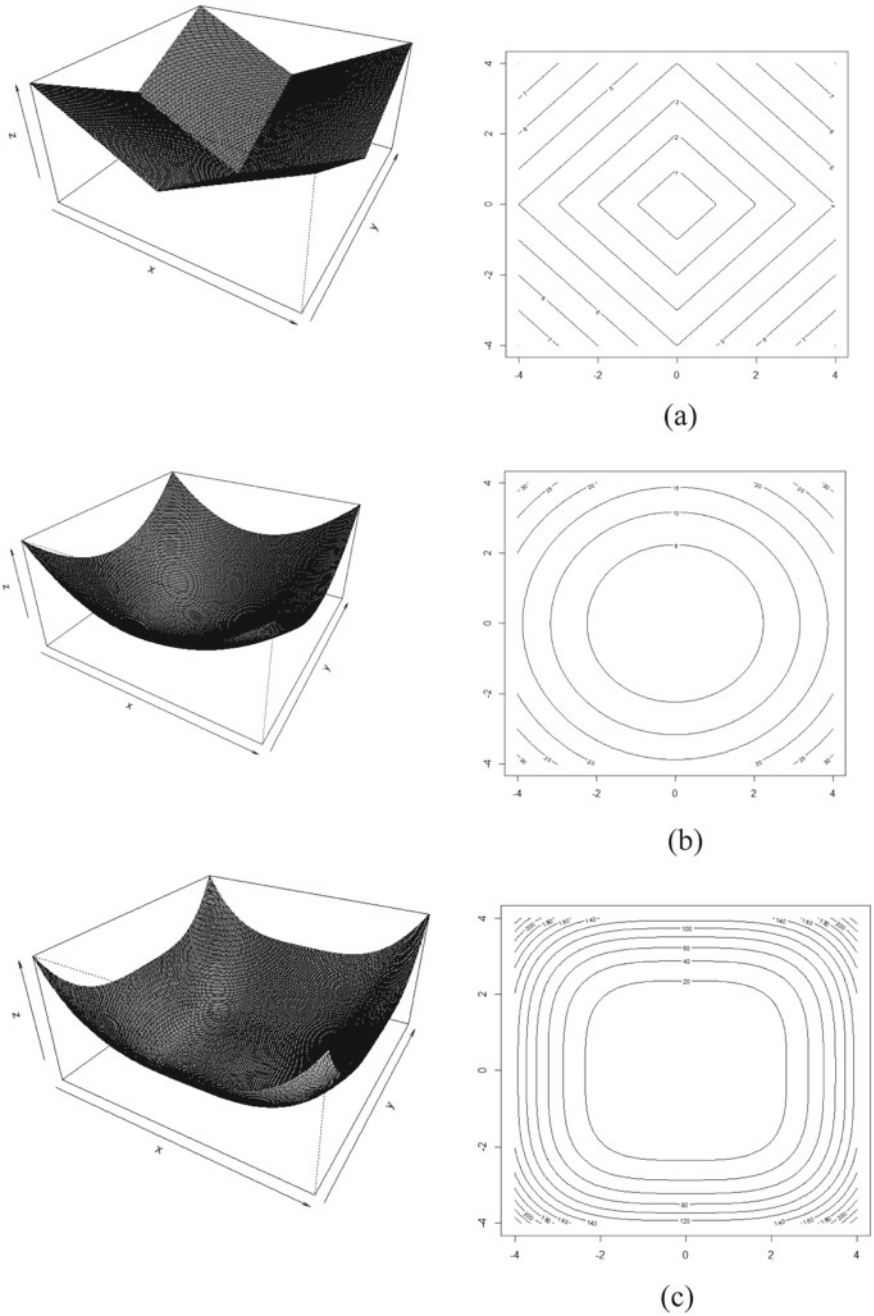
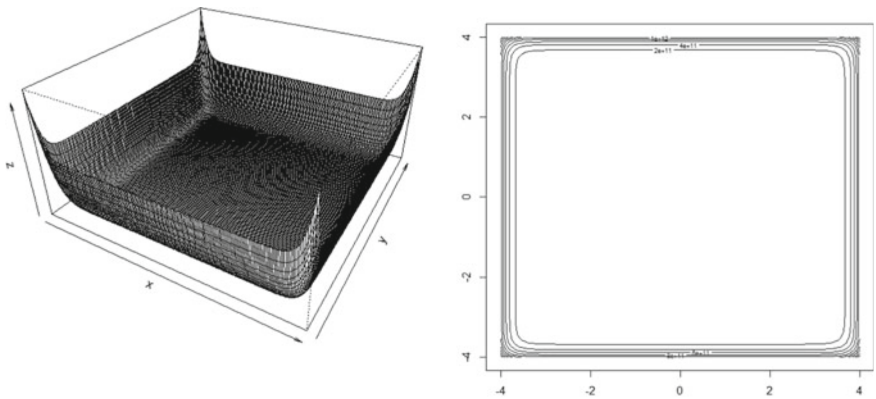


Fig. 9.11 Plots of various distance functions (3D plots and contour plots): **a** Hamming, **b** Euclidean, **c** p = 3.5, **d** Tchebyshev



(d)

Fig. 9.11 (continued)

$$\frac{\partial}{\partial v_{st}}|x_{kt}-v_{st}| = \frac{\partial}{\partial v_{st}}\begin{cases} x_{kt}-v_{st}, & \text{if } x_{kt} \geq v_{st} \\ -x_{kt}+v_{st}, & \text{otherwise} \end{cases}$$
$$= \begin{cases} -1, & \text{if } x_{kt} \geq v_{st} \\ +1, & \text{otherwise} \end{cases} - \operatorname{sgn}(x_{kt}-v_{st}) \quad (9.29)$$

For the Tchebyshev distance (L_∞) the detailed formulas are given below

$$\frac{\partial}{\partial v_{st}} \max_{j=1,2,\ldots,n}\left|x_{kj}-v_{sj}\right| = \frac{\partial}{\partial v_{st}}\begin{cases} \left|x_{kt}-v_{st}\right|, & \text{if } \max\limits_{\substack{j=1,2,\ldots,n,\\ j\neq t}} \left|x_{kj}-v_{sj}\right| < \left|x_{kt}-v_{st}\right| \\ \max\limits_{\substack{j=1,2,\ldots,n,\\ j\neq t}} \left|x_{kj}-v_{sj}\right|, & \text{otherwise} \end{cases}$$
$$= \begin{cases} \frac{\partial}{\partial v_{st}}\left|x_{kt}-v_{st}\right|, & \text{if } \max\limits_{\substack{j=1,2,\ldots,n,\\ j\neq t}} \left|x_{kj}-v_{sj}\right| < \left|x_{kt}-v_{st}\right| \\ 0, & \text{otherwise} \end{cases} \quad (9.30)$$

The FCM algorithm uses two nested iterative loops

- randomly initialize partition matrix U

  repeat

    repeat

    - update prototypes using (9.29) or (9.30)

    until a certain termination criterion has been met

    - update partition matrix (9.13)

  until a certain termination criterion has been satisfied

9.9 Conditional Fuzzy C-Means

The conditional FCM is an example of knowledge-based clustering [8]. The essence of this method is that clustering is carried out in the context of some auxiliary information granule so that the formation of the structure is *conditional* implied by some additional piece of knowledge. Formally speaking, the problem is formulated as follows: given is a finite set of data composed of pairs $(\boldsymbol{x}_k, \boldsymbol{y}_k)$, $k = 1, \ldots, N$ where $\boldsymbol{x}_k \in \boldsymbol{R}^n$ and $\boldsymbol{y}_k \in \boldsymbol{R}^p$. D stands for some predefined information granule (say, a fuzzy set) defined in $\boldsymbol{R}^p$ and serves as a context which navigates (constraints) the process of building clusters in $\boldsymbol{R}^n$. While the aim of the clustering task can be verbally articulated as

- find a structure in data $\boldsymbol{X}$

  the goal of conditional clustering can be posed as

- find a structure in $\boldsymbol{X}$ in the presence of context (condition) D

The ensuing minimized objective function is the same as given by (9.14) however the constraints imposed on the partition matrix involve D and read as follows

$$\sum_{i=1}^{c} u_{ik} = D(y_k) \tag{9.31}$$

In other words, D serves as a selector (logic filter) imposed on original data using which a contribution of some data in $\boldsymbol{R}^n$ is limited ($D(\boldsymbol{y}_k) < 1$) or eliminated completely ($D(\boldsymbol{y}_k) = 0$), see Fig. 9.12.

The solution to the above optimization problem leads to the computing of the partition matrix realized in the following way

$$A_i(\boldsymbol{x}_k) = \frac{D(\boldsymbol{y}_k)}{\sum_{j=1}^{c} \left(\frac{||\boldsymbol{x}_k - v_i||^2}{||\boldsymbol{x}_k - v_j||^2} \right)^{1/(m-1)}} \tag{9.32}$$

The computing of the prototypes is unchanged and follows (9.16) given that the Euclidean distance has been considered.

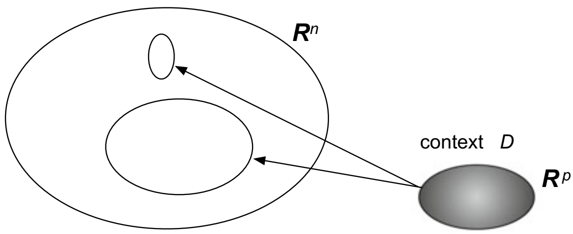


Fig. 9.12 Conditional FCM: only a subset of data is clustered

9.10 Conclusions

The exposure of fuzzy clustering severs two main purposes. First, we emphasized that clustering serves as a conceptual and algorithmic environment for data analytics; here the unsupervised mode of learning and interpretability of results are the evident assets. From the design perspective of fuzzy sets, fuzzy clustering is essential to the construction of membership function based on available data. The generic optimization algorithm is presented. The role of a suite of parameters of the FCM algorithm is discussed, in particular, the fuzzification coefficient. The distance function and the number of clusters. The characterization of the structure with the aid of the proximity matrices and association matrices offers another view at the relationships among fuzzy sets. Clustering serves as a prerequisite for the development of fuzzy models and will be discussed in the consecutive chapters.

Problems

1. Given is the following partition matrix

$$U = \begin{bmatrix} 0.3 & 1.0 & 0.4 & 0.0 & 0.1 \\ 0.0 & 0.0 & 0.3 & 0.9 & 0.5 \\ 0.7 & 0.0 & 0.3 & 0.1 & 0.4 \end{bmatrix}$$

   (i) determine proximity matrix P
   (ii) determine association matrix and identify two entries of the highest entries of this matrix.

2. Run the FCM algorithm for the butterfly data shown in Fig. 9.13. Experiment with different values of c and m. Interpret the obtained results.
3. Could the following matrix be regarded as a partition matrix

$$U = \begin{bmatrix} 1.0 & 0.3 & 0.5 \\ 0.0 & 0.0 & 0.0 \\ 0.1 & 0.7 & 0.3 \end{bmatrix}$$

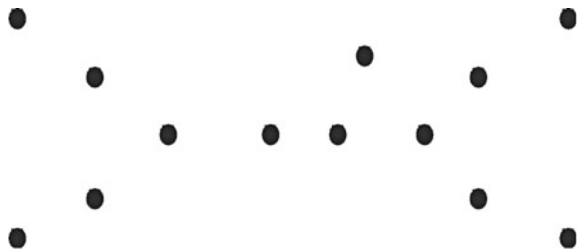


Fig. 9.13 Two-dimensional butterfly data

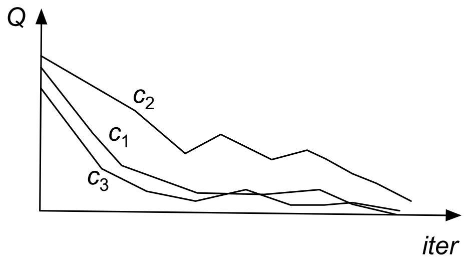


Fig. 9.14 Objective function obtained in successive iterations for selected values of the number of clusters

4. Which data (pattern) is the most unclear and as such might require further investigation

$$U = \begin{bmatrix} 0.4 & 0.3 & 0.2 & 1.0 & 0.1 & 0.2 \\ 0.2 & 0.4 & 0.0 & 0.0 & 0.9 & 0.1 \\ 0.4 & 0.3 & 0.8 & 0.0 & 0.0 & 0.7 \end{bmatrix}$$

5. By inspecting the behavior of the objective function in successive iterations, Fig. 9.14, what would be a suitable number of clusters?
6. If you are to cluster cars shown in digital photos, what would be a data space in which clustering could be carried out.
7. For the following partition matrix, determine cluster validity indexes (partition coefficient, partition entropy, Fukuyama-Sugeno, Xie-Beni)

$$U = \begin{bmatrix} 0.4 & 0.4 & 0.2 & 1.0 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.0 & 0.0 & 0.9 & 0.1 \\ 0.4 & 0.1 & 0.8 & 0.0 & 0.0 & 0.5 \end{bmatrix}$$

8. Discuss how to carry out clustering of interval-valued data as shown in Fig. 9.15.

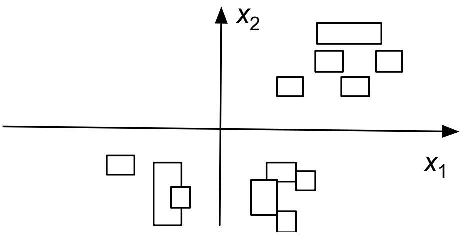


Fig. 9.15 Example of two-dimensional interval data

References

1. Anderberg, M.R.: Cluster Analysis for Applications. Academic Press, New York (1973)
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
3. Bezdek, J.C.: Cluster validity with fuzzy sets. J. Cybern. **3**, 58–73 (1974)
4. Bezerra, C.G., Costa, B.S.J., Guedes, L.A., Angelov, P.P.: An evolving approach to data streams clustering based on typicality and eccentricity data analytics. Inf. Sci. **518**, 13–28 (2020)
5. D'Urso, P., De Giovanni, L., Disegna, M., Massari, R.: Fuzzy clustering with spatial–temporal information. Spat. Stat. **30**, 71–102 (2019)
6. Fukuyama, Y., Sugeno, M.: A new method of choosing the number of clusters for the fuzzy c-means method. In: Proceedings of 5th Fuzzy Systems Symposium, pp. 247–250 (1989)
7. Ianni, M., Masciari, E., Mazzeo, G.M., Mezzanzanica, M., Zaniolo, C.: Fast and effective big data exploration by clustering. Future Gener. Comput. Syst. **102**, 84–94 (2020)
8. Pedrycz, W.: Knowledge-Based clustering. John Wiley & Sons, Hoboken (2005)
9. Shukla, A.K., Muhuri, P.K.: Big-data clustering with interval type-2 fuzzy uncertainty modeling in gene expression datasets. Eng. Appl. Artif. Intell. **77**, 268–282 (2019)
10. Tao, Q., Gu, C., Wang, Z., Jiang, D.: An intelligent clustering algorithm for high-dimensional multiview data in big data applications. Neurocomputing (in press) (2019)
11. Wang, W., Zhang, Y.: On fuzzy cluster validity indices. Fuzzy Sets Syst. **158**, 2095–2117 (2007)
12. Xie, X.L., Beni, G.: A validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell. **13**, 841–847 (1991)

Chapter 10
The Principle of Justifiable Granularity

Abstract The previous chapters focused on the construction of fuzzy sets (or information granules) based either on formalization of expert (user) perception or data (data-driven constructs). Both of them, in spite of compelling arguments, exhibit some limitations and weaknesses. The approach discussed here and referred to as a principle of justifiable granularity, can be regarded as the one that exploits the available experimental data while augmenting the construct by some domain knowledge either in a form of a single component of the general criterion or in the form of more problem-oriented domain knowledge. In this sense, one can think of the principle as taking a position in-between the two discussed previously. We highlight the main features of the construct by pointing out at its generality. Furthermore we show that the principle can be used as a follow up of the numeric constructs produced by fuzzy clustering and results formed in group decision processes.

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_10

10.1 The Main Idea

The principle of justifiable granularity guides a construction of an information granule based on available experimental evidence [4, 5]. For further extensions and applications, the reader can refer to Wang et al. [7], Kosheleva and Kreinovich [2], Zhongjie and Jian [8].

In a nutshell, when using this principle, we emphasize that a resulting information granule becomes a summarization of data (viz. the available experimental evidence). The underlying rationale behind the principle is to deliver a concise and abstract characterization of the data such that (i) the produced granule is *justified* in light of the available experimental data, and (ii) the granule comes with a well-defined *semantics* meaning that it can be easily interpreted and becomes distinguishable from the others.

Formally speaking, these two intuitively appealing criteria are expressed by the criterion of coverage and the criterion of specificity. Coverage states how much data are positioned behind the constructed information granule. Put it differently—coverage quantifies an extent to which information granule is supported by available experimental evidence. Specificity, on the other hand, is concerned with the semantics of information granule stressing the semantics (meaning) of the granule. We start with a one-dimensional case of data for which we design information granule.

One-dimensional case

The definition of coverage and specificity requires formalization and this depends upon the formal nature of information granule to be formed. As an illustration, consider an interval form of information granule A. In case of intervals built on a basis of one-dimensional numeric data (evidence) $x_1, x_2, \ldots, x_N$, the coverage measure is associated with a count of the number of data embraced by A, namely

$$\mathrm{cov}(A) = \frac{1}{N} card\{x_k | x_k \in A\} \tag{10.1}$$

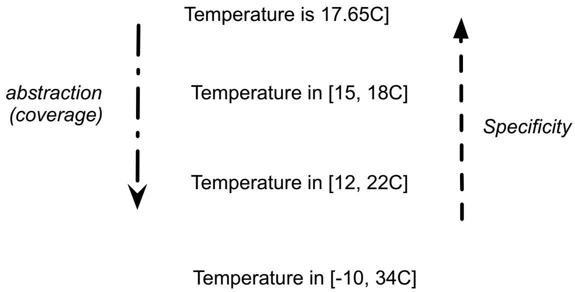
card (.) denotes the cardinality of A, viz. the number (count) of elements x_k belonging to A. In essence, coverage has a visible probabilistic flavor. Let us recall that the specificity of A, sp(A) is regarded as some decreasing function g of the size (length, in particular) of information granule. If the granule is composed of a single element, sp(A) attains the highest value and returns 1. If A is included in some other information granule B, then $sp(A) > sp(B)$. In a limit case if A is an entire space of interest $sp(A)$ returns zero. For an interval-valued information granule $A = [a, b]$, a simple implementation of specificity with g being a linearly decreasing function comes as

$$sp(A) = g(length(A)) = 1 - \frac{|b - a|}{range} \tag{10.2}$$

where *range* stands for an entire space over which intervals are defined.

The criteria of coverage and specificity are in an obvious relationship, Fig. 10.1.

Fig. 10.1 Relationships between abstraction (coverage) and specificity of information granules of temperature



We are interested in forecasting temperature: the more specific the statement about this prediction becomes, the lower the likelihood of its satisfaction is.

From the practical perspective, we require that an information granule describing a piece of knowledge has to be meaningful in terms of its existence in light of the experimental evidence and at the same time, it is specific enough. For instance, when making a prediction about temperature, the statement about the predicted temperature 17.65 is highly specific but the likelihood of its being true is practically zero. On the other hand, the piece of knowledge (information granule) describing temperature as an interval $[-10, 34]$ lacks specificity (albeit is heavily supported by experimental evidence) and its usefulness is highly questionable—as such this information granule is very likely regarded as non-actionable. No doubt, some sound compromise is needed. It is behind the principle of justifiable granularity.

Witnessing the conflicting nature of the two criteria, we introduce the following product of coverage and specificity

$$V = cov(A)sp(A) \tag{10.3}$$

The desired solution (viz. the developed information granule) is the one where the value of V attains its maximum. Formally speaking, consider that an information granule is described by the vector of parameters $\boldsymbol{p}$, $V(\boldsymbol{p})$. The principle of justifiable granularity applied to experimental evidence returns to an information granule that maximizes $V, \boldsymbol{p}_{\text{opt}} = \arg_{\boldsymbol{p}} V(\boldsymbol{p})$.

To maximize the index V through the adjusting the parameters of the information granule, two different strategies are encountered

(i) a two-phase development is considered. First a numeric representative (mean, median, modal value, etc.) is determined. It can be sought as an initial representation of the data. Next the parameters of the information granule are optimized by maximizing V. For instance, in case of an interval $[a, b]$, one has the bounds (a and b) to be determined. These two parameters are determined separately, viz. the values of a and b are determined by maximizing $V(a)$ and $V(b)$. The data used in the maximization of $V(b)$ involves these data larger than the numeric representative. Likewise $V(a)$ is optimized on a basis of the data lower than this representative.

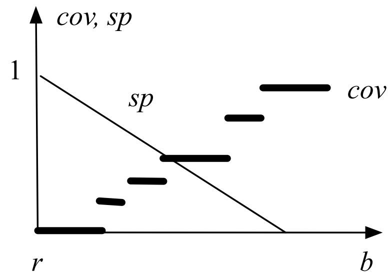


Fig. 10.2 Example plots of coverage and specificity (linear model) regarded as a function of *b*

(ii) a single-phase procedure in which all parameters of information granule are determined at the same time.

The two-phase algorithm works as follows. Having a certain numeric representative of $\boldsymbol{X}$, say the mean, it can be regarded as a rough initial representative of the data. In the second phase, we separately determine the lower bound (a) and the upper bound (b) of the interval by maximizing the product of the coverage and specificity as formulated by the optimization criterion. This simplifies the process of building the granule as we encounter two separate optimization tasks

$$a_{\text{opt}} = \arg Max_a V(a) \quad V(a) = \text{cov}([a, r])^* \text{sp}([a, r])$$
$$b_{\text{opt}} == \arg Max_b V(b) \quad V(b) = \text{cov}([r, b])^* \text{sp}([r, b]) \tag{10.4}$$

We calculate $\text{cov}([r, b]) = \text{card}\ \{x_k | x_k \in [r, b]\}/N$. The specificity model has to be provided in advance. Its simplest linear version is expressed as $\text{sp}([r, b]) = 1 - |b - r|/(x_{\max} - r)$. By sweeping through possible values of b positioned within the range $[r, x_{\max}]$, we observe that the coverage is a stair-wise increasing function whereas the specificity decreases linearly, see Fig. 10.1. The maximum of the product can be easily determined (Fig. 10.2).

The determination of the optimal value of the lower bound of the interval a is completed in the same way as above. We determine the coverage by counting the data located to the left from the numeric representative r, namely $\text{cov}([a, r]) = \text{card}\ \{x_k | x_k \in [a, r]\}/N$ and compute the specificity as $\text{sp}([a, r]) = 1 - |a - r|/(\text{r} - x_{\min})$.

Example 1 Let us consider a collection of one-dimensional data

$$\{-5.2\ \ -4.5\ \ -2.0\ \ -1.9\ \ -0.5\ 0.1\ 0.7\ 0.8\ 1.7\ 1.9\ 2.2\ 2.5\ 3.1\ 3.5\ 4.4\ 5.3\}$$

We construct an interval information granule $[a, b]$ by optimizing the values of a and b separately. We start with the numeric representative in the form of the mean value, $r =$ The mean is 0.756. The range $x_{\max}$-r is 4.54

The detailed results of calculations of coverage and specificity for the values of b coinciding with the values of the data are reported below.

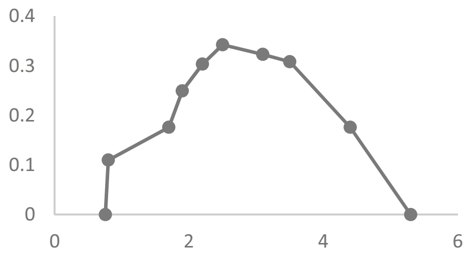


Fig. 10.3 Coverage, specificity and V as functions of b

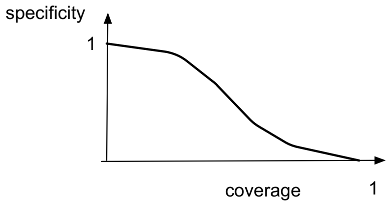


Fig. 10.4 Characteristics of the information granule in the coverage-specificity plane

b	0.8	1.7	1.9	2.2	2.5	3.1	3.5	4.4	5.3
cov	0.11	0.22	0.33	0.44	0.55	0.66	0.77	0.88	1.00
sp	0.99	0.79	0.74	0.68	0.61	0.48	0.40	0.20	0.00

The plot of V is displayed in Fig. 10.3; The optimal value of b, b_{opt} is 2.5 for which V is 0.34.

In general, the characteristics of the information granule can be conveniently visualized in the coverage-specificity plane, see Fig. 10.4.

Depending on the values of b (we are concerned with the optimization of $V(b)$) when moving with their values we position of the corresponding point (coverage, specificity) traverses the space. For $b = r$ the specificity achieves 1 however the coverage is zero. Higher values of b lead to higher values of coverage. The decreasing form of changes of the curve is data dependent.

Example 2 Determine an interval information granule A in the presence of data governed by a uniform probability density function $p(x)$ spread from -*range* to *range*.

Starting from a numeric representative $r = 0$, we determine an optimal upper bound b (the lower bound a is determined in the same manner). The coverage is computed as $cov(A) = \int_0^b p(x)dx$ while the corresponding specificity is $sp(A) = 1 - b/range = \varphi(b)$. The necessary condition for the maximum of $V(b) = cov(A)sp(A)$ is provided by taking the zero value of the derivative of $V(b)$. This leads to the following equation

$$\frac{d}{db}\left(\int_0^b p(x)dx\right)\varphi(b) + \left(\int_0^b p(x)dx\right)\frac{d\varphi(b)}{db} = 0 \tag{10.5}$$

For the uniform distribution of $p(x)$ one has cov(A) = $b/range$ and specificity $sp(A) = 1 - b/range$. The product of coverage and specificity produces $b/range(1 - b/range) = b/range - b^2/range^2$. Taking the derivative and making it equal to zero, one has 1-2$b/range = 0$ and finally $b = range/2$.

Some additional flexibility to the design of can be added to the optimized performance index by adjusting the impact of the specificity in the construction of the information granule. This is done by bringing a power factor ξ as follows

$$V(a, b) = \text{cov}(A)^{*}\text{sp}(A)^{\xi} \tag{10.6}$$

Note that the values of ξ lower than 1 discount the influence of the specificity; in the limit case this impact is eliminated when $\xi = 0$. The value of ξ set to 1 returns the original performance index whereas the values of ξ greater than 1 stress the importance of specificity by producing results that are more specific.

10.2 Design of Fuzzy Sets

The principle of justifiable granularity can be used to design of membership functions. Two directions are envisioned. The first one builds upon the representation theorem. Recall that a fuzzy set is a union of sets (intervals) indexed by the levels of membership grades (α) The construct we presented builds a certain α-cut. Combining the individually built cuts, a fuzzy set is formed. The second one is focused on the parametric optimization of the membership function whose type has been selected in advance.

Membership function formation on a basis of the representation theorem

The generalized version of the optimization performance index (6) is of interest here. Increasing values of the power ξ stress the increasing relevance of the specificity meaning that the corresponding intervals are more specific. For a given ξ, the result is the ξ-cut of the fuzzy set. The value of the power ξ is normalized to [0, 1], say by considering a linear transformation $\alpha = (\xi - \xi_{\min})/(\xi_{\max} - \xi_{\min})$, $\xi_{\min} = 0$. Then ξ-cuts A_ξ are aggregated forming a fuzzy set.

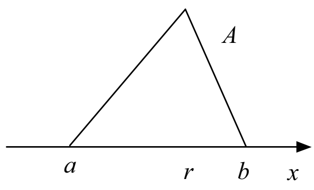
Parametric optimization of membership functions

When constructing an information granule in the form of a fuzzy set [6], the implementation of the principle has to be modified. Considering some predetermined form of the membership function, say a triangular one, the parameters of this fuzzy set (lower and upper bounds, a and b) are optimized. See Fig. 10.5.

The coverage is replaced by a σ-count by summing up the membership grades of the data in A (in what follows we are concerned with the determination of the upper bound of the membership function, namely b)

$$\text{cov}(A) = \sum_{k x_k > r} A(x_k) \tag{10.7}$$

Fig. 10.5 Triangular membership function with adjustable (optimized) bounds *a* and *b*



The coverage computed to determine the lower bound (a) is expressed in the form

$$\text{cov}(A) = \sum_{kx_k<r} A(x_k) \tag{10.8}$$

Multi-dimensional case

The principle of justifiable granularity can be applied to multidimensional (n-dimensional) data $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$. We assume that the data are normalized to $[0, 1]^n$.

As before, we proceed with a two-phase design process. We assume that the data are normalized to [0, 1] meaning that each coordinate of the normalized $\boldsymbol{x}_k$ assumes values positioned in [0, 1]. The numeric representative (prototype) is first formed; denote it by $\boldsymbol{r}$. For instance, one can calculate the average, $\boldsymbol{r} = \sum_{k=1}^{N} \boldsymbol{x_k}/N$

Around the numeric prototype $\boldsymbol{r}$ one spans an information granule $A(\boldsymbol{r}, \rho)$ whose optimal size ρ is obtained as the result of the maximization of the well-known criterion

$$\rho_{\text{opt}} = \arg \text{Max}_{\rho}[cov(A)sp(A)] \tag{10.9}$$

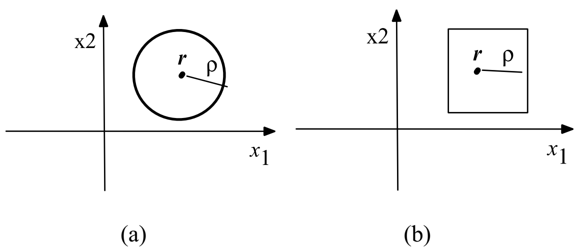
In more detail we have

$$cov(A) = \frac{1}{N}\text{card}\left\{\boldsymbol{x}_k \middle| ||\boldsymbol{x}_k - \boldsymbol{r}||^2 \le n\rho^2\right\} \tag{10.10}$$

$$\text{sp}(A) = 1 - \rho_i \tag{10.11}$$

Note that the geometry of the resulting information granule is implied by the form of the distance function ||. || used in (10.10). For the Euclidean distance, the granule is a circle. For the Tchebyshev one, we end up with a hyperbox shape, see Fig. 10.6.

Fig. 10.6 Development of information granules in the two-dimensional case when using two distance functions: **a** Euclidean distance, and **b** Tchebyshev distance



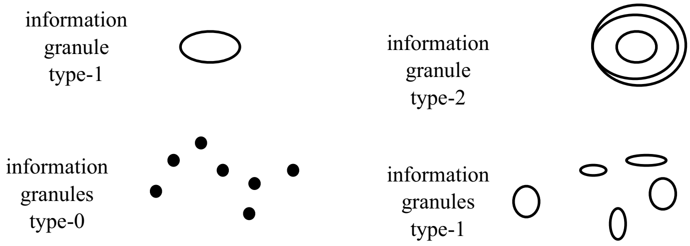


Fig. 10.7 Elevation of type of information granules

10.3 General Observations: An Elevation Effect

As a way of constructing information granules, the principle of justifiable granularity exhibits a significant level of generality in two essential ways. First, given the underlying requirements of coverage and specificity, different formalisms of information granules can be engaged. Second, experimental evidence could be expressed as information granules articulated in different formalisms and on this basis certain information granule is being formed.

The principle of justifiable granularity highlights an important facet of elevation of the type of information granularity: the result of capturing a number of pieces of numeric experimental evidence comes as a single abstract entity—information granule. As various numeric data can be thought as information granule of type-0, the result becomes a single information granule of type-1. This is a general phenomenon of elevation of the type of information granularity. The increased level of abstraction is a direct consequence of the diversity present in the originally available granules. This elevation effect is of a general nature and can be emphasized by stating that when dealing with experimental evidence composed of a *set* of information granules of type-n, the result becomes a *single* information granule of type $(n + 1)$ (Fig. 10.7).

Some generalizations and their augmentations are reported by Zhongjie and Jian [8].

10.4 The Principle of Justified Granularity in the Presence of Weighted and Inhibitory Data

Information granule can be constructed based on data with some additional characteristics, namely we consider weighted data and inhibitory data. In what follows we concentrate on the two-phase design process and look at the optimization of the upper bound only.

Weighted data

The data used to construct an information granule come with the corresponding weights assuming values in [0, 1]. We have the pairs of weighted one- dimensional

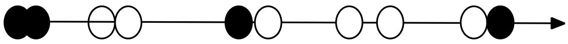


Fig. 10.8 Distribution of excitatory and inhibitory data

data (x_1, w_1) (x_2, w_2),…, (x_N, w_N). The weights quantify the relevance of the corresponding data points. The coverage criterion reflects the weights and comes as the following sum

$$\mathrm{cov}(A) = \sum\nolimits_{x_k \in [r,b]} w_k / N \tag{10.12}$$

The specificity criterion remains the same as the one used in the generic construct.

Inhibitory data

Inhibitory data arise in classification problem in which we build an information granule which is pertinent to some class of interest- excitatory data (and the data in this class imply this granule) while at the same time we would like to neglect the data belonging to some other class or classes. Those are referred to as inhibitory data. We also admit that the data come with their weights. The excitatory data are (x_1, w_1) (x_2, w_2),…, (x_N, w_N). The inhibitory data are

(y_1, f_1) (y_2, f_2),…, (y_N, f_M). See Fig. 10.8.

Now the coverage criterion comes in the following form (we are concerned with the optimization of the upper bound of the interval information granule b)

$$\mathrm{cov}(A) = \max\Big(0, \sum\nolimits_{x_k \in [r,b]} w_k - \sum\nolimits_{y_k \in [r,b]} wf_k\Big)/N \tag{10.13}$$

The above expression emphasizes that bringing more excitatory data increases coverage whereas an inclusion of inhibitory data in A brings some penalty. The maximum operation assures that the coverage assumes nonzero values.

10.5 Adversarial Information Granules

In untargeted adversarial attacks (Kurakin et al. [1, 3], one considers $\boldsymbol{x}$' such that it is close to the data coming from the training data $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ and producing a significantly different results than those reported for the neighboring data. The nature of the adversarial data $\boldsymbol{x}$' can be quantified and generalized to the idea of the granular adversarial data. In light of the essence of the adversarial property, we determine $\boldsymbol{x}$' such that it is close to $\boldsymbol{x}_k$ and $F(\boldsymbol{x}')$ is different from $f(\boldsymbol{x}_k)$ where $F(.)$ is a certain classifier or a model realizing this mapping F. $\boldsymbol{x}$' is sought as an adversarial example.

The *granular* adversarial data centered around $\boldsymbol{x}$' and denoted by $A(\boldsymbol{x}'; \rho)$ whose size (radius) ρ is the one which maximize the following ratio

$$V(\rho) = \frac{\sum_{x_k:\|x_k - x'\| \le n\rho^2} |F(x_k) - F(x')|}{\sum_{x_k:\|x_k - x'\| \le n\rho^2} |x_k - x'|} \tag{10.14}$$

viz.

$$\rho\max = \arg \mathrm{Max}_{\rho} V(\rho) \tag{10.15}$$

where ||.|| is a certain distance function, say the Euclidean one.

10.6 Designing Fuzzy Sets of Type-2

Group of experts (a typical scenario encountered in problems of Computing with Words) where type-2 fuzzy sets are used. In particular interval-valued fuzzy sets. The data are a collection of fuzzy sets of type-1 provided by N experts and defined in some discrete space. For instance, when evaluating preferences of individual solutions using the AHP method, each expert produces a certain fuzzy set. In this sense, for each solution, we obtain a collection of membership degrees $x_1, x_2, \ldots, x_N$. Applying to them the principle of justifiable granularity, an interval [a, b] bering a subset is formed. In this way we obtain an efficient way of estimating interval-valued fuzzy sets. In case we build a fuzzy set over the unit interval, the result becomes a type-2 fuzzy set.

10.7 The Accommodation of Domain Knowledge

Any available domain knowledge is diversified and could be represented in different formats. In particular, it could be coming through some dependent variable one encounters in regression and classification problems. An information granule is built on a basis of experimental evidence gathered for some input variable and now the associated dependent variable is engaged. In the formulation of the principle of justifiable granularity, this additional information impacts a way in which the coverage is determined. In more detail, we discount the coverage; in its calculations, one has to take into account the nature of experimental evidence assessed on a basis of some external source of knowledge. In regression problems (continuous output/dependent variable), in the calculations of specificity, we consider the variability of the dependent variable y falling within the realm of A. More precisely, the value of coverage is discounted by taking this variability into consideration. In more detail, the modified value of coverage is expressed as

$$\mathrm{cov}'(A) = \mathrm{cov}(A)\exp\left(-\beta\sigma_y^2\right) \tag{10.16}$$

where σ is a standard deviation of the output values associated with the inputs being involved in the calculations of the original coverage cov(A). β is a certain calibration factor controlling an impact of the variability encountered in the output space. Obviously, the discount effect is noticeable, cov'(A) < cov(A).

In case of a classification problem in which p classes are involved $\omega = \{\omega_1\ \omega_{2,\ldots},\ \omega_p\}$, the coverage is again modified (discounted) by taking into account the diversity of the data embraced by the information granule. This diversity is quantified in the form of the entropy function $h(\omega)$

$$\mathrm{cov}'(A) = \mathrm{cov}(A)(1 - h(\omega)) \tag{10.17}$$

where $h(.)$ is the entropy of data contained within the bounds of A. This expression penalizes the diversity of the data contributing to the information granule and not being homogeneous in terms of class membership. The higher the entropy, the lower the coverage cov'(A) reflecting the accumulated diversity of the data falling within the umbrella of A. If all data for which A has been formed belong to the same class, the entropy returns zero and the coverage is not reduced, cov'(A) = cov(A).

10.8 Collaborative Development of Information Granules

The discussion presented so far has been concentrated on the construction of a single information granule with some eventual auxiliary information provided by the information granules. In what follows, we consider a construction of a collection of information granules A_1, A_2, …, A_c where the design of A_i takes into account an impact from other information granules and impact their development at the same time. The maximized performance index is the following sum

$$V'(\rho_1, \rho_2, \ldots, \rho_c) = V(\rho_1) + V(\rho_2) + \cdots + V(\rho_c) \tag{10.18}$$

where

$$V(\rho_i) = cov(A_i)sp(A_i) \tag{10.19}$$

$$\begin{aligned} cov(A_i) = \max(0, \mathrm{card}\{\boldsymbol{x_k}|\ \|\boldsymbol{x_k} - \boldsymbol{v}_i\|^2 \le n\rho_i^2\} \\ - \mathrm{card}\{\boldsymbol{x_k}|\ ||\boldsymbol{x_k} - \boldsymbol{v}_i||^2 \le n\rho_i^2\}\&\exists_{j\neq i}\boldsymbol{x_k} \in A_j\} \end{aligned} \tag{10.20}$$

$$sp(A_i) = 1 - \rho_i \tag{10.21}$$

We assume that the numeric representatives $\boldsymbol{v}_1, \boldsymbol{v}_1, \boldsymbol{v}_c$ are given. The radii $\rho_1, \rho_c, \ldots, \rho_c$ have to be modified so that (10.19) attains its maximum. Here one could use any of the methods of population-based optimization, say Particle Swarm Optimization (PSO).

10.9 Fuzzy Clustering and the Principle of Justifiable Granularity

It is worth noting that there is a striking difference between clustering and the principle of justifiable granularity. First, clustering leads to the formation at least two information granules (clusters) whereas the principle of justifiable granularity produces a single information granule. Second, when positioning clustering and the principle vis-à-vis each other, the principle of justifiable granularity can be sought as a follow-up step facilitating an augmentation of the numeric representative of the cluster (such as e.g., a prototype) and yielding granular prototypes where the facet of information granularity is retained. The results of clustering are used to invoke a detailed construction of information granules. The numeric representatives are the prototypes formed by the FCM clustering.

The design situations we presented so far could be conveniently linked with the clustering results produced by the FCM algorithm. There is some compelling argument. The prototypes that are the result of clustering are numeric representatives of numeric data exhibiting substantial diversity. This diversity is not present in the numeric prototypes. We advocate that the prototypes as the sound representatives of the data should be regarded as information granules (what is underlined as the visible facet delivered by the principle of justifiable granularity). This means that while starting with the numeric prototypes, we elevate them to granular counterpart. The construction of information granules positioned around the FCM-produced prototype $\boldsymbol{v}_i$ uses weighted data with the weights being the corresponding membership values (the entries of the i-th row of the partition matrix) u_{i1}, u_{i2}, …, u_{iN}. The coverage is then computed by determining the weighted sum in the form

$$\mathrm{cov}(A_i) = \sum\nolimits_{\boldsymbol{x}_k : \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 \le \rho_i^2} u_{ik}/N \tag{10.22}$$

The above formula is the one we use when dealing with weighted data

inhibitory data Again when forming an information granule with the numeric representative vi, the inhibitory effect comes because of the associated membership grades. For the k-th data point we have u_{ik} that contributes to the formation of the i-th granule and are also supplied with the inhibitory membership degrees u_{1k}, u_{2k}, …, $u_{(i\text{-}1)k}$, $\mathrm{u}_{(i+1)k}$,…, u_{ck}

$$\mathrm{cov}(A_i) = \max\left(0, \sum\nolimits_{\boldsymbol{x}_k : \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 \le \rho_i^2} u_{ik} - \sum\nolimits_{\boldsymbol{x}_k : \|\boldsymbol{x}_k - \boldsymbol{v}_\mathrm{i}\|^2 \le \rho_i^2} u_{j*k}\right)/N \tag{10.23}$$

where j^* is an index of the cluster such that $u_{j*k} = \max_{\mathrm{j}\neq\mathrm{i}} u_{jk}$.

In summary, given a plethora of scenarios in which information granules are constructed, their overall representation can be symbolically presented as

$$G = (\boldsymbol{x}; \text{geometry, information content}) \tag{10.24}$$

where the geometry embraces all descriptors that characterize the geometric features of the granule, viz. the center, radius, and the form of the distance used in its design. The information content part might not be present or could relate to the nature of the problem at hand; say

$G = (\boldsymbol{x}, \boldsymbol{v}, \rho; \sigma^2)$ or $G = (\boldsymbol{x}; \boldsymbol{v}, \rho; h(\boldsymbol{\omega}))$.

10.10 Conclusions

The principle of justifiable granularity applies to various formalisms of information granules and this makes this approach substantially general. Several important variants of the principle are discussed below where its generic version becomes augmented by available domain knowledge.

Problems

1. Using the principle of justifiable granularity, determine an interval -valued fuzzy set for the following family of fuzzy sets

$$\begin{aligned}
A_1 &= [1.0\ 0.7\ 0.5\ 0.3\ 0.1\ 0.5\ 0.7] \\
A_2 &= [1.0\ 0.3\ 0.6\ 0.8\ 0.5\ 0.4\ 0.3] \\
A_3 &= [1.0\ 0.7\ 0.3\ 0.3\ 0.1\ 0.1\ 0.5] \\
A_4 &= [0.6\ 0.7\ 1.0\ 0.5\ 0.8\ 0.5\ 0.7] \\
A_5 &= [0.2\ 0.7\ 0.5\ 0.3\ 0.1\ 1.0\ 1.0]
\end{aligned}$$

2. For one-dimensional data shown below, determine an information granule using the principle of justifiable granularity.

$$\{-4.2, 6.7, 0.1, -3.2, 1.1, 2.7, 4.6, -2.5, -2.1, -1.3, 0.1, -0.7, 2.5, -3.1\}$$

   Plot the resulting coverage-specificity relationship.

3. With the use of the principle of justifiable granularity, determine a fuzzy set with a triangular membership function with the modal value equal to zero for data governed by a triangular probability density function $p(x)$

$$p(x) = \begin{cases} -\frac{2x}{a^2} + \frac{2}{a} \text{ if } x \in [0, a] \\ 0, \text{ otherwise} \end{cases}$$

4. The two-dimensional data (x_1, x_2) are listed below. Determine an information granule considering a Tchebyshev distance. Consider the mean vector as a numeric representative of the information granule.

$$\{(1, -4), (3.1, -1.4), (10, 4), (7.4, -2.1), (4.4, 2.7), (6.7, 2.0), (-1.0, -1.6), (4.1, 2.5), (4.1, -3.8)\}$$

5. Use the same data as in Problem 2 with some associated output variable y with the following entries

$$\{-1.0, 5.3, 10.2, -7.1, 0.2, -4.2, 5.1, -3.3, -2.9, -1.3, 3.8, -0.9, 6.1, -4.0\}$$

Determine interval information granule. Interpret the obtained result vis-à-vis the one obtained in Problem 2.

References

1. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2014). arXiv:1412.6572
2. Kosheleva, O., Kreinovich, V.: Measures of specificity used in the principle of justifiable granularity: a theoretical explanation of empirically optimal selections. In: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2018, pp. 1–7
3. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world (2016). arXiv:1607.02533
4. Pedrycz, W.: Granular Computing. CRC Press, Boca Raton, FL (2013)
5. Pedrycz, W., Homenda, W.: Building the fundamentals of granular computing: a principle of justifiable granularity. Appl. Soft Comput. **13**, 4209–4218 (2013)
6. Pedrycz, W., Wang, X.: Designing fuzzy sets with the use of the parametric principle of justifiable granularity. IEEE Trans. Fuzzy Syst. **24**, 489–496 (2016)
7. Wang, X., Pedrycz, W., Gacek, A., Liu, X.: From numeric data to information granules: a design through clustering and the principle of justifiable granularity. Knowl.-Based Syst. **1011**, 100–113 (2016)
8. Zhongjie, Z., Jian, H.: Stabilizing the information granules formed by the principle of justifiable granularity. Inf. Sci. **503**, 183–199 (2019)

Chapter 11
Granulation-Degranulation Processes

Abstract In this chapter, we discuss the concept of information granulation and degranulation as one of the fundamental computing paradigms of Granular Computing supporting interaction with real-world environment. In the setting of fuzzy sets, the wording fuzzification-defuzzification is commonly used. The related terms concern encoding-decoding and this terminology is used in data compression. We elaborate on the underlying concept, discuss algorithms of granulation and degranulation along with their propose a way of evaluating the quality of this transformation. Several design alternatives are discussed.

11.1 The Concept of Information Granulation and Degranulation

11.1.1 Granulation

The concept of information granulation can be succinctly outlined as follows. Having a finite collection of reference fuzzy sets $\{A_i\}$ referred to as a codebook and defined in some space $\boldsymbol{X}$, we represent a given information granule (fuzzy set, in particular

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_11

or a numeric input) A with the aid of A_is. Symbolically, we regard a granulation operation G as a mapping

$$G: \boldsymbol{X} \rightarrow [0, 1]^c \tag{11.1}$$

The result of granulation forms a representation of any A in an internal format delivered by the reference fuzzy sets. The essence is to determine a way in which any input datum matches the corresponding information granules A_i.

Example 1 Describe the granulation (encoding) result when A is a numeric entity x_0, $x_0 \in \boldsymbol{R}$, while $\{A_i\}$ are fuzzy sets defined in $\boldsymbol{R}$.

The granulation process returns a c-dimensional vector of membership grades $[A_1(x_0)\ A_2(x_0)\ldots\ A_c(x_0)]$.

Example 2 If A is a numeric entity and A_i are numeric intervals, the granulation process returns a c-dimensional Boolean vector [0 0…0 1 0…0].

It becomes apparent that the requirement for the reference information granules to return a meaningful result is that that $\{A_i\}$ "covers" the entire space. There are no "gaps" which could prevent the granulation process from delivering sound results.

Example 3 Assuming the same reference information granules as discussed in Example 1, expressing information granule A becomes more demanding. The commonly encountered process involves computing possibility (Poss) and necessity (Nec) degrees of A [formulas (11.13) and (11.14)] with respect to A_is. In this situation, the result becomes a 2c-dimensional vector with the entries [Poss(A, A_1), Poss(A, A_2) … Poss(A, A_c) Nec(A, A_1), Nec(A, A_2) … Nec(A, A_c)].

Consider the subset of the space of real numbers $\boldsymbol{X} \subset \boldsymbol{R}$. Having a collection of intervals or fuzzy sets, the granulation realizes a nonlinear transformation (normalization) and elevates the dimensionality of the representation space (which is an element of the c-dimensional unit hypercube).

Let us look at the granulation process in case when $\boldsymbol{X} \subset \boldsymbol{R}^n$. The granulation result is produced by the minimization of the following performance index

$$Q = \sum_{i=1}^{c} u_i^m \|\boldsymbol{x} - \boldsymbol{v}_i\|^2$$
$$\text{subject to } \sum_{i=1}^{c} u_i = 1, u_i \in [0, 1] \tag{11.2}$$

where $\boldsymbol{v}_i$s are the numeric prototypes (representatives) of the reference fuzzy sets A_i, $m > 1$, and ||.|| is the Euclidean distance.

The minimization of Q is completed under the constraint stating that the elements of the internal representation u_i sum up to 1. Obviously $u_i \in [0, 1]$. Under these circumstances, the problem with constraints becomes transformed into unconstrained optimization with the aid of the Lagrange multiplier (λ)

$$Q\prime = \sum_{i=1}^{c} u_i^m \|\boldsymbol{x} - \boldsymbol{v}_i\|^2 + \lambda\left(\sum_{i=1}^{c} u_i - 1\right) \tag{11.3}$$

Taking the derivatives of Q' and solving the system of equations

$$\begin{aligned} \frac{dQ'}{d\lambda} &= 0 \\ \frac{dQ'}{du_i} &= 0 \end{aligned} \tag{11.4}$$

the solution becomes

$$u_i = \frac{1}{\sum_{j=1}^{c}\left(\frac{\|\boldsymbol{x}-\mathbf{v}_i\|^2}{\|\boldsymbol{x}-\boldsymbol{v}_j\|^2}\right)^{1/(m-1)}} \tag{11.5}$$

In this sense, for any $\boldsymbol{x}$ one obtains the internal representation of A as a vector in the $[0,1]^c$ hypercube.

Note that the result is the same as developed for the FCM algorithm when optimizing the partition matrix.

11.1.2 Degranulation

The degranulation process considers some internal representation of the original datum and produces its reconstruction. In other words, we move from the space where internal representation has been completed (encoding, granulation) and returns the result in the original space we started with. This phase is commonly referred to as degranulation, decoding, or defuzzification

With some slight abuse of notation, the degranulation process is denoted by G^{-1}. Ideally, we may expect that the combined granulation followed by the degranulation process returns the original input $\boldsymbol{x}$, namely $G^{-1}(G(\boldsymbol{x})) = \boldsymbol{x}$. More realistically, instead of this outcome, one has $\boldsymbol{x} \approx \hat{\boldsymbol{x}}$ where $\hat{\boldsymbol{x}} = G^{-1}(G(\boldsymbol{x}))$. See also Fig. 11.1.

In virtue of the mapping G^{-1} which moves from more abstract representation to the detailed one at the lower level of abstraction (higher specificity), by no means the degranulation process is unique. One can refer to a way of representing a numeric function by a single number; there is an infinite number of possible mappings delivering this transformation. This is profoundly visible when discussing the degranulation process (defuzzification)—there is a great number of sound possibilities investigated in the literature.

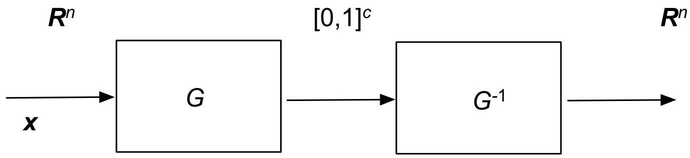


Fig. 11.1 A tandem of granulation and degranulation processes; note a transformation between spaces of physical variables and spaces of information granules

The selection of the decoding could be guided by the reconstruction performance measure (reconstruction error) measuring the departure of $\hat{\boldsymbol{x}}$ from $\boldsymbol{x}$, viz.

$$V = \sum_{i=1}^{c} u_i^m \left\| \boldsymbol{v}_i - \hat{\boldsymbol{x}} \right\|^2 \tag{11.6}$$

The solution comes in the form

$$\hat{\boldsymbol{x}} = \frac{\sum_{i=1}^{c} u_i^m \boldsymbol{v}_i}{\sum_{i=1}^{c} u_i^m} \tag{11.7}$$

In what follows, we discuss two general and commonly studied situations. The first one is focused on the decoding in a one-dimensional space and a family of reference fuzzy sets. The second one concerns a multivariable case.

11.2 Degranulation Process—A One Dimensional Case

An interesting result of lossless degranulation (reconstruction) where $V = 0$, in one-dimensional case. Interestingly, the results in the case of one-dimensional and vector quantization are well reported in the literature when dealing with compression methods [2, 3, 5, 6, 11]. While the quantization error could be minimized, the method is in essence lossy (coming with a nonzero quantization error). In contrast, a family of fuzzy sets can lead to the zero reconstruction error. With this regard we show a surprisingly simple yet powerful result whose essence could be summarized as follows: a family of fuzzy triangular fuzzy sets (fuzzy numbers) with an overlap of ½ present at any two neighboring elements, see Fig. 11.2, leads to the lossless granulation-degranulation mechanism. The essence of the scheme can be formulated in the form of the following proposition.

Proposition 1
The following two assumptions are satisfied:

(i) *Given is a collection of c triangular fuzzy sets A_i with ½ overlap whose modal values are $v_1 < v_2 < \ldots < v_c$. The space of interest is $[v_1, v_c]$; see* Fig. 11.2.

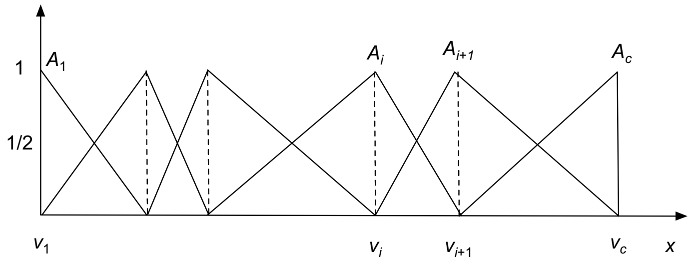


Fig. 11.2 An example of the codebook composed of triangular fuzzy sets with an overlap of ½ between each two neighboring elements of the codebook

(ii) *The granulation process returns membership grades of* A_i*s. The degranulation process returns the result which is a weighted sum of the modal values.* *Then for any* $x \in [v_1, v_c]$ *the granulation-degranulation process produces an error-free reconstruction error.*

Proof In virtue of (i) the membership functions are described as follows

$$A_i(x) = \begin{cases} \frac{x-v_{i-1}}{v_i-v_{i-1}} \text{ if} x \in [v_{i-1}, v_i] \\ \frac{x-v_{i+1}}{v_i-v_{i+1}} \text{ if} x \in [v_i, v_{i+1}] \end{cases} \tag{11.8}$$

Consider any $x \in \left[v_{i-1}, v_i\right]$. In light of (ii) one has $\hat{x} = A_{i-1}(x)v_{i-1} + A_i(x)v_i$. Note that only two fuzzy sets are activated with a nonzero membership degree. Plugging (11.8) into the degranulation formula for $\hat{x}$ returns the following expression

$$\hat{x} = \frac{x - v_i}{v_{i-1} - v_i} v_{i-1} + \frac{x - v_{i-1}}{v_i - v_{i-1}} v_i = \frac{1}{v_{i-1} - v_i}((x - v_i)v_{i-1} + (x - v_{i-1})v_i)$$
$$= \frac{1}{v_{i-1} - v_i}(x(v_{i-1} - v_i) = x \tag{11.9}$$

Interestingly enough, triangular fuzzy sets have been commonly used in the development of various fuzzy set constructs as it will discussed later on (models, controllers, classifiers, etc.) yet the lossless character of such codebooks has not been generally acknowledged and utilized, perhaps with very few exceptions [8].

The obtained result about the lossless information granulation with the use of fuzzy sets stands in a sharp contrast with the lossy granulation-degranulation realized with sets (intervals). In essence, in this case we encounter well-known schemes of analog-to-digital A/D conversion (granulation) and digital-to- analog D/A conversion (degranulation). It is obvious that such conversions produce an inevitable error (quantization error).

11.3 Degranulation Process with the Clustering Environment

Fuzzy clustering produces a collection of fuzzy sets. We investigate capabilities of the produced fuzzy sets to realize the granulation-degranulation process. For any $\boldsymbol{x}$ (either being a part of the data clustered or any new data not used in clustering), the granulation formula is the one producing the partition matrix. The prototypes $\boldsymbol{v}_i$ entail $u_i(\boldsymbol{x})$. The degranulation is determined with the aid of (11.7).

If all data coming from some data set $\boldsymbol{D}$ are used, the reconstruction error comes in the form of the following sum

$$V = \sum_{\boldsymbol{x} \in \boldsymbol{D}} \left\| \boldsymbol{x} - \hat{\boldsymbol{x}} \right\|^2 \tag{11.10}$$

Generally, if the data are governed by some probability density function $p(\boldsymbol{x})$, the reconstruction error reads as follows

$$V = \int\limits_{\boldsymbol{x} \in \boldsymbol{D}} \left\| \boldsymbol{x} - \hat{\boldsymbol{x}} \right\|^2 p(\boldsymbol{x}) d\boldsymbol{x} \tag{11.11}$$

For the uniform $p(\boldsymbol{x})$ one obtains

$$V = \int\limits_{\boldsymbol{x} \in \boldsymbol{D}} (\boldsymbol{x} - \hat{\boldsymbol{x}})^2 d\boldsymbol{x} \tag{11.12}$$

The quality of the granulation-degranulation scheme depends upon a number of parameters and they can be optimized. The two of them are the most visible: (a) the number of information granules (clusters), and (b) fuzzification coefficient (m). The role of the first one is apparent: the increase of the cardinality of the codebook $\{A_i\}$ leads to the reduction of the reconstruction error. The second parameter changes the form of the associated membership; the values close to 1, $m \to 1$ lead to the membership functions being close to the characteristic functions present in set theory. Its optimal value is problem-dependent. A collection of typical curves presenting the dependence of the reconstruction error is visualized in Fig. 11.3; refer to [9, 10] for extensive experimental studies. The larger the codebook is, the smaller the reconstruction error becomes. The optimal value of m, however, is data dependent. Interestingly, in light of the criterion discussed here the choice $m = 2$, which is commonly encountered in the literature is not always justified; the optimal values of the fuzzification can vary in a quite broad range of values.

Example 4 Considering the clustering carried out for the data studied in Example 1, Chap. 8. We investigate an impact of the fuzzification coefficient on the reconstruction abilities of the information granules. In the previous discussion with the regard to the FCM algorithm, we remarked that commonly used value of m is 2.0, however

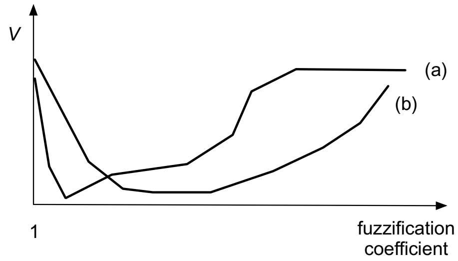


Fig. 11.3 Reconstruction error versus the fuzzification coefficient for selected values of c

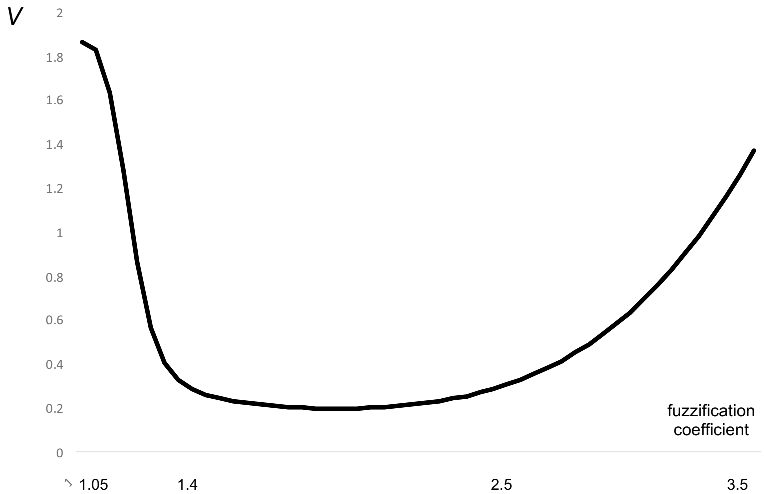


Fig. 11.4 Reconstruction error V displayed as a function of m

this choice does not seem to have a strong experiment support. In particular case the optimal value of m is 1.95; also the relationship is flat in the range of in-between 1.4 and 2.5. The choice of m is evidently data dependent (Fig. 11.4).

The optimal value of m very close to 1 indicate a limited superiority of the FCM over the K-Means algorithm.

11.4 Granulation and Degranulation for Granular Data

So far, we have discussed various ways of realizing granulation and degranulation when the input datum is numeric. Here we consider a situation when the datum is an information granule A, say some set or fuzzy set.

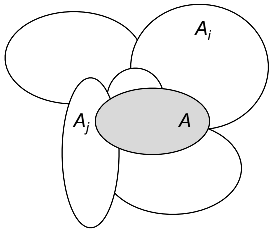


Fig. 11.5 A collection of information granules A_i and information granule A

Intuitively, one could envision a number of possible ways of describing A in terms of the elements forming this codebook. A way, which is quite appealing is to describe a relationship between A and A_i in terms of coincidence (overlap) of these two and an inclusion of A_i in the information granule of interest, see Fig. 11.5.

The degree of overlap (intersection) quantified in terms of the possibility measure of A with A_i is computed in the form

$$\mathrm{Poss}(A,\ A_i) = \sup_{\boldsymbol{x}\in \boldsymbol{X}}[A(\boldsymbol{x})tA_i(\boldsymbol{x})] \tag{11.13}$$

The degree of inclusion of A_i in A is determined the in the following way

$$\mathrm{Nec}(A,\ A_i) = \inf_{\boldsymbol{x}\in \boldsymbol{X}}[(1 - A_i(\boldsymbol{x})sA(\boldsymbol{x})] \tag{11.14}$$

where t and s are some t-norm and t-conorm, respectively. In virtue of (11.13) and (11.14), the following relationship holds $\mathrm{Nec}(A,\ A_i) \le \mathrm{Poss}(A,\ A_i)$. If the information granule A comes as a numeric form, namely

$$A(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_0) = \begin{cases} 1, if\boldsymbol{x} = \mathbf{x}_0 \\ 0, if\boldsymbol{x} \ne \mathbf{x}_0 \end{cases} \tag{11.15}$$

then the degree of overlap coincides with the degree of inclusion; one has Poss(A, A_i) = Nec(A, A_i) = $A(x_0)$. In case A is a fuzzy set with a continuous membership function and A_i are intervals $A_i = [a_i, a_{i+1}]$ then the possibility and necessity degrees are equal to the maximal and minimal value A assumes over the interval $[a_i, a_{i+1}]$ that is $\mathrm{Poss}(A, A_i) = \max_{\mathrm{x}\in[\mathrm{a_i},\mathrm{a_{i+1}}]}A(x)$ and Nec(A, A_i) = $A(x)$. The plot shown in Fig. 11.6 illustrates these details of the calculations.

It is worth noting that the concepts of overlap and inclusion (no matter how they are realized) are intuitively appealing in the determination of the relationships between two information granules.

In summary, the granulation process returns a 2c-dimensional vector whose coordinates are possibility and necessity measures; refer to Example 3.

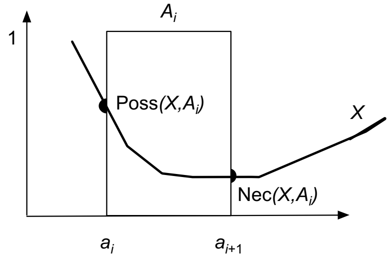


Fig. 11.6 Computing matching and inclusion degrees for the interval information granules of the codebook

11.4.1 Degranulation

Having the possibility and necessity values determined for each element of A_i, we can formulate a reconstruction problem as follows: determine A given the family $\{A_i\}$ and the levels of inclusions and overlap, Poss(A, A_i) and Nec(A, A_i).

The solution is developed in the setting of relational calculus and relational equations, in particular [1, 4, 7]. The expressions (11.13) and (11.14) can be viewed as sup-t and inf-s fuzzy relational equations that need to be solved with respect to A. As discussed in Chap. 4, the solutions to the relational equations are not unique. The extreme solutions are determined in the following way:for (11.13) the upper bound solution is

$$\hat{A}(\boldsymbol{x}) = \min_{i=1,2,\ldots,c} [A_i(\boldsymbol{x})\phi \text{Poss}(A, A_i)] \tag{11.16}$$

for (11.14) the lower bound solution is

$$\breve{A}(\boldsymbol{x}) = \max\nolimits_{\mathrm{i}\,=\,1,2,\,\ldots,\mathrm{c}}[(1 - A_i(\boldsymbol{x})\text{Nec}(A, A_i)] \tag{11.17}$$

Recall that the two aggregation operators are defined as follows, refer to Chap. 4

$$a\phi b = \sup\{c \in [0, 1]|atc \le b\}ab = \inf\{c \in [0, 1]\,|asc \ge b\}, a,\ b \in [0, 1] \tag{11.18}$$

In virtue of the lower and upper bound solutions formed in this way, the result of degranulation comes as an interval-valued fuzzy set. In other words, we witness that there is an effect of the elevation of the type of the fuzzy set as the direct result when dealing with the representation and the ensuing reconstruction problem. Continuing with the example of interval information granules forming the family of reference sets, we obtain the results:

$$\hat{A}(x) = \begin{cases} 1 & \text{if } x \notin \left[a_i, a_{i+1}\right] \\ \text{Poss}(A, A_i) & \text{otherwise} \end{cases} \tag{11.19}$$

and

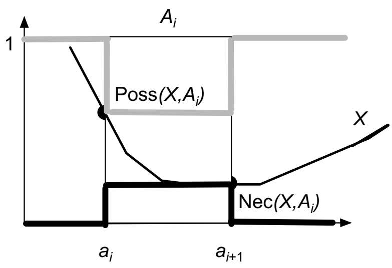


Fig. 11.7 Result of reconstruction realized with the use of A_i

$$\breve{A}(x) = \begin{cases} 0 & \text{if } x \notin [a_i, a_{i+1}] \\ \mathrm{Nec}(A, A_i) & \text{otherwise} \end{cases} \tag{11.20}$$

see also Fig. 11.7. It becomes apparent that the result in an interval-valued construct with the bounds determined by the maximal and minimal values A takes over $[a_i, a_{i+1}]$. It is noticeable that A_i serves as probe using which we reconstruct a part of A that falls within the interval. For all xs outside this interval, the reconstruction bounds are non-specific as the result is returned as the whole unit interval. Collectively, the usage of all A_is helps reconstruct A in a meaningful way; see Fig. 11.7. It is also apparent the quality of granulation-degranulation depends on the number of the information granules in the codebook; this point is made clear when contrasting the degranulation results shown in Fig. 11.8a, b.

Consider the granulation- degranulation process realized for all As coming from a certain category of information granules $\boldsymbol{F}$. The optimization problem itself can

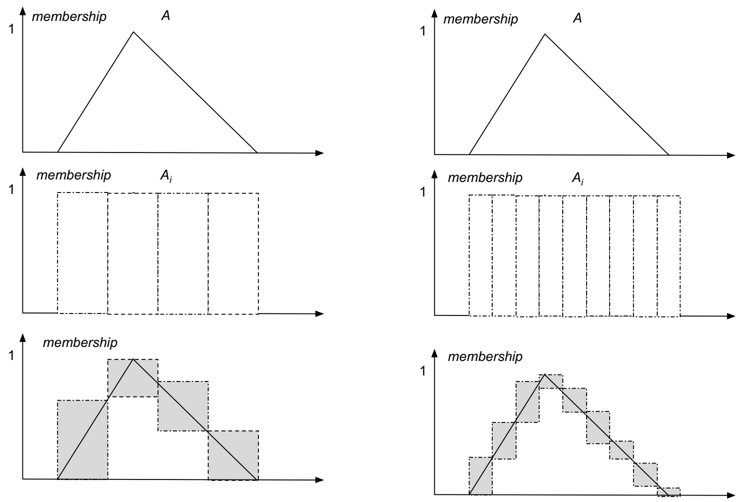


Fig. 11.8 Reconstruction results in the form of interval -valued fuzzy set: **a** $c = 4$ and **b** $c = 9$

be structured in several ways. For instance, assuming the form of the information granules (say, Gaussian membership functions), one optimizes the parameters so that the difference between the bounds A and A becomes minimized. The more advanced formulation of the problem could involve a selection of the form of the information granules (in case of fuzzy sets, we can talk about triangular, trapezoidal, *S*-shape, Gaussian membership functions), this structure—oriented aspect of the optimization has to be considered. Typically, population-based optimization tools are highly suitable here.

11.5 Conclusions

The quality of granulation and degranulation process is central to a large array of constructs involving fuzzy sets. In the chapter, it was shown that any representation of information granules and numeric entities is associated with the degranulation error, which could be quantified. The optimization of the reference information granules is a well-formulated optimization problem. As a result, the codebook of the granules becomes instrumental in the discovery and quantification of the essential relationships in the collection of numeric or granular data.

The granulation and degranulation mechanisms are also essential in system modeling realized with the aid of information granules. The granulation error has to be taken into consideration when assessing the quality of models realized in such a framework. With this regard, in the one-dimensional case, the role of triangular fuzzy sets becomes highly advantageous as delivering the lossless granulation-degranulation scheme.

Problems

1. Consider a one-dimensional granulation-degranulation problem where the codebook is formed by parabolic membership functions $P(x;\ m,\ a)$ described as follows

$$P(x;\ m,\ a) = \begin{cases} 1 - \left(\frac{x-m}{a}\right)^2, & \text{if } x \in [m-a, m+a] \\ 0, & \text{otherwise} \end{cases}$$

The fuzzy sets are given as $P(x;\ 1, 4)$, $P(x;\ 3, 4)$, $P(x;\ 6, 5)$, and $P(x;\ 10, 4)$.

Determine the reconstruction error by computing the following integral $V = \int_0^{10} (x - \hat{x})^2 dx$

2. Repeat Problem 1 where V involves a probability density function $p(x)$

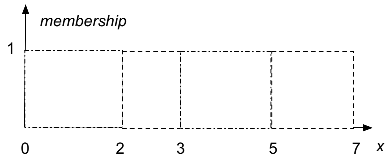


Fig. 11.9 Interval-based codebook

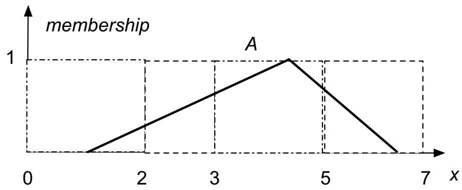


Fig. 11.10 Fuzzy set *A* and an interval-based codebook

$$V = \int_{0}^{10} \left(x - \hat{x}\right)^{2} p(x)dx$$

where $p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-(x-m)2/(2\sigma^2)$, m = 4 and $\sigma = 1.5$.

3. The codebook is composed of intervals shown in Fig. 11.9.

Plot a reconstruction error as a function of x. If you increase the number of intervals by splitting the existing intervals, how does it affect the reconstruction error?

4. Show a reconstruction of a triangular fuzzy set A realized with the aid of the codebook formed by a collection of intervals; see Fig. 11.10.

References

1. Di Nola, A., Sessa, S., Pedrycz, W., Sanchez, E.: Fuzzy Relational Equations and Their Applications in Knowledge Engineering. Kluwer Academic Publishers, Dordrecht (1989)
2. Gersho, A., Gray, R.M.: Vector Quantization and Signal Compression. Kluwer Academic Publishers, Boston, MA (1992)
3. Gray, R.: Vector quantization. IEEE Acoust. Speech Signal Process. Mag. **1**, 4–29 (1984)
4. Hirota, K., Pedrycz, W.: Fuzzy relational compression. IEEE Trans. Syst. Man Cybern. **29**(3), 407–415 (1999)
5. Lendasse, A., Francois, D., Wertz, V., Verleysen, M.: Vector quantization: a weighted version for time-series forecasting. Future Gener. Comput. Syst. **21**, 1056–1067 (2005)
6. Linde, Y., Buzo, A., Gray, R.: An algorithm for vector quantizer design. IEEE Trans. Commun. COM **28**(1), 84–95 (1988)
7. Nobuhara, H., Pedrycz, W., Hirota, K.: Fast solving method of fuzzy relational equation and its application to lossy image compression/reconstruction. IEEE Trans. Fuzzy Syst. **8**(3), 325–335 (2000)
8. Pedrycz, W.: Why triangular membership functions? Fuzzy Sets Syst. **64**, 21–30 (1994)

9. Pedrycz, W., Valente de Oliveira, J.: Optimization of fuzzy models. IEEE Trans. Syst. Man Cybern. Part B **26**(4), 627–636 (1996)
10. Pedrycz, W., Valente de Oliveira, J.: A development of fuzzy encoding and decoding through fuzzy clustering. IEEE Trans. Instrum. Meas. **57**(4), 829–837 (2008)
11. Yair, E., Zeger, K., Gersho, A.: Competitive learning and soft competition for vector quantizer design. IEEE Trans. Signal Process. **40**(2), 294–309 (1992)

Chapter 12
Fuzzy Models: Fundamentals and Design

Abstract In this chapter, we introduce a concept of fuzzy models and elaborate on their architectures. We offer a unified structural view at the diversity of the topologies of fuzzy models and outline a general methodological framework within which all the design processes are carried out. The two main directions of the evaluation (numeric and granular) of the performance of the models are outlined. Main classes of representative fuzzy models are outlined and their key characteristics are identified and stressed.

12.1 Fuzzy Models: A Concept

In a nutshell, fuzzy models are constructs which use fuzzy sets as building blocks or/and their design involves the design methodology of fuzzy sets. In a general setting their architectures are composed of two main functional modules: interfaces (input and output) and a processing module, see Fig. 12.1. Fuzzy models are built and carry out processing at the higher level of abstraction than the level of abstraction of the data themselves originated by the physical environment of real-world systems. This entails the general topology of the fuzzy models and makes the interface modules indispensable. In what follows, we discuss a role and functionality of the key modules.

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_12

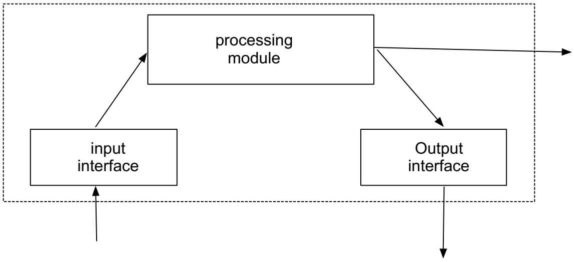


Fig. 12.1 General topology of fuzzy models composed of interfaces and the processing module: note that the results produced by the processing module can be used directly or after the transformation completed by the output interface

12.1.1 Input Interface

The role of the input interface is to facilitate an interaction of the model with the physical world. The environment generates numeric data or streams of data. The interface transforms these data to the format required by the processing module. Formally, the interface forms a mapping from $\boldsymbol{R}$ (or $\mathbf{R}^n$) to the membership values of the fuzzy sets forming this interface. In virtue of the properties of the family of fuzzy sets composing the input interface, this mapping is nonlinear. The suitably structured nonlinearity helps enhance the capabilities of the processing module. In sum, any input $x \in \boldsymbol{R}$ or $(\mathbf{R}^n)$ gives rise to a vector of membership grades in the $[0,1]^c$. The change in dimensionality of spaces (by moving from n and c) is worth noting as well; typically, c could be far lower than n and this relationship implies dimensionality reduction delivered by the input interface.

Another alternative is to admit a single fuzzy set associated with an individual input variable such that its membership function is monotonic (to assure a one-to-ne mapping of any input to the corresponding element in the unit interval). In this encoding, there is no dimensionality reduction. If the membership function is nonlinear, the nonlinearity of the mapping occurs. Apparently, a linear normalization of the interval $[a, b]$ to $[0,1]$ is realized by a triangular fuzzy set, Fig. 12.2. A piecewise linear or nonlinear membership function realizes a nonlinear normalization (scaling) of the input space as visualized in Fig. 12.2 as well. Alternatively, one can consider a

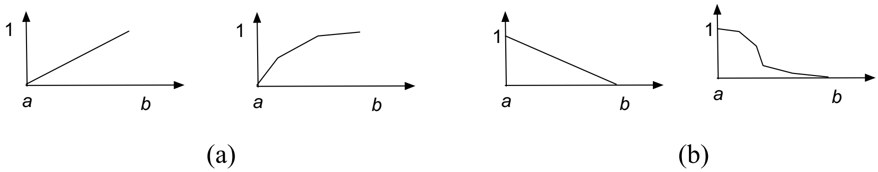


Fig. 12.2 Examples of linear and nonlinear membership functions realizing input transformation: **a** monotonically increasing, and **b** monotonically decreasing

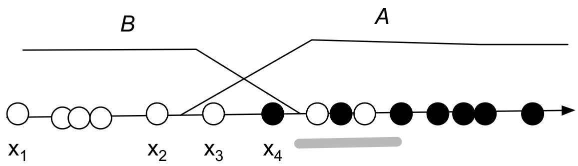


Fig. 12.3 Data and their representation in the space of fuzzy sets (membership grades)

linearly decreasing membership function or its nonlinear generalization to carry out the transformation.

Example 1 Consider a classification problem where two classes of patterns are to be dealt with. The distribution of data belonging to the two classes is shown in Fig. 12.3. For the classifier, the region shown in grey color is the one where the classes overlap and where the design focus of the classifier is on. To facilitate this development, we "magnify" these regions while at the same time supress (squash) the regions that are homogeneous with respect to the classes present there. x_1 and x_2 produce the same vector in the three-dimensional space of membership grades although they are very different in the space of the real values. In contrast, x_1 and x_3 are very distinct in terms of the membership values even though they are very close in the original feature space. Note that fuzzy sets distributed non-uniformly across the space deliver the desired effect of nonlinear transformation, see Fig. 12.3.

12.1.2 Output interface

The role of this interface is to convert a result produced by the processing module and located at the more abstract level into the numeric format required and recognized at the level of the physical world; see Fig. 12.1. In the area of fuzzy sets and fuzzy models, computing done by the output interface is referred to a defuzzification. The processing result delivered by this interface is not unique: in essence, we produce different numeric results that could serve as sound representative of the fuzzy set, viz. calculate a numeric representative of a membership function. This discussion relates also to the granulation-degranulation studies discussed in Chap. 10 (Granulation and degranulation processes). Several commonly encountered methods are listed below; refer also the Figs. 12.4, 12.5 outlining different scenarios and displaying a way how the degranulation mechanism operates. Consider some fuzzy set A.

Maximum of membership function. As the name states, the numeric representative is determined as

$$x_0 = \arg\max_x A(x) \tag{12.1}$$

x_0 is just the element of the space for which the membership function attains one. If (A is normal, $A(x_0) = 1$). There are several variants of this method. If the membership

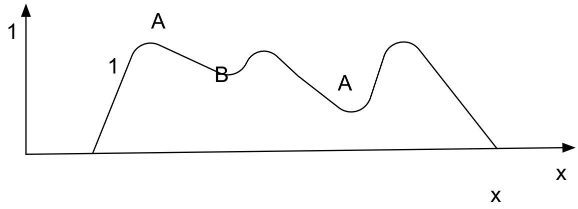


Fig. 12.4 Multimodal fuzzy set A and ways of determining its numeric representative

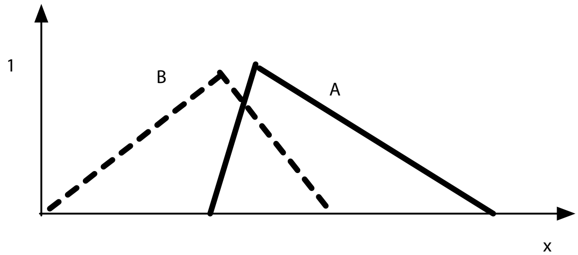


Fig. 12.5 Fuzzy sets A and B resulting in the same result of degranulation when using the maximum decoding rule

function is multimodal, one can choose the average of the maxima, the lowest value of the argument or the highest value of the argument where the maximum occurs. Say, we have p modal values of A, $x_{01}, x_{02}, \ldots, x_{0p}$. Then we determine the average $1/p \sum_{i=1}^{p} x_{0i}$, $\min_i x_{0i}$ and $\max_i x_{0i}$. see Fig. 12.4.

The method is sound in the sense that it is based on the maximal membership grades so x_0 is the most typical numeric value of A, however, it does not take into account the shape of the membership function; see Fig. 12.5—in both cases, even though the membership functions of A and B are different, the result of degranulation is the same.

12.1.3 Center of Gravity

In this method, the numeric representative is expressed as

$$x_0 = \int_R A(x)x dx / \int_R A(x) dx \tag{12.2}$$

As a matter of fact, this representative is well motivated and comes as a solution to the following minimization problem

$$\int_R A(x)(x - x_0)^2 dx \tag{12.3}$$

x_0 is a solution to the following equation

$$\int_R A(x)(x - x_0)dx = 0 \tag{12.4}$$

namely

$$\int_R A(x)xdx = x_0 \int_R A(x)dx \tag{12.5}$$

and hence the solution comes as (12.2). One can note that the expressions above can be sought as the 1st and 0th moment of the membership function A, namely $m_1 = \int_R A(x)xdxm_0 = \int_R A(x)dx$. In other words, $x_0 = m_1/m_0$. The method exhibits superiority over the maximum of membership by reflecting the shape of the membership function in the obtained result. The drawback comes with some computing overhead associated with the integration (or summation) required to produce the result.

A number of modifications are also considered. For instance, to avoid a bias caused by long tails of the membership function, a certain threshold γ is introduced, which yields the following expression

$$x_0 = \int_{x:A(x)\geq\gamma} A(x)xdx / \int_{x:A(x)\geq\gamma} A(x)dx \tag{12.6}$$

If $\gamma = 0$, then the above formula converts to (12.2) while γ =1 for a normal fuzzy set A implies the maximum decoding rule. The use of the output interface in fuzzy models is dependent upon the way how the model is being used:

In situations when the model operates in a user-centric mode, the output interface could not be beneficial as conveying the result in the form of a fuzzy set rather than a single numeric value is far more informative. Consider a certain decision variable such as predicted inflation. The numeric outcome is far less informative than a fuzzy set, see Fig. 12.6. The shape of the membership function, which is highly asymmetric, underlines that the higher values of inflation are very likely than those lower than the numeric outcome (a). It is needless to say that A is far more useful and informative for any well-informed decision-making activities.

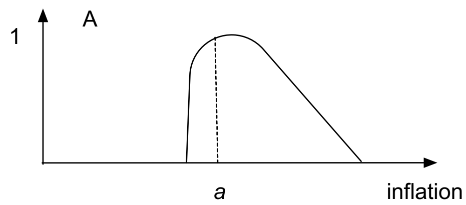


Fig. 12.6 Numeric versus fuzzy set-based result produced by the fuzzy mode

12.1.4 Processing Module

The module operates at the level of information granules (fuzzy sets) and determines the output for some given input. There is a plethora of architectures of fuzzy models.

Overall fuzzy model forms a nonlinear mapping from $\boldsymbol{R}^n$ to $\boldsymbol{R}$. In contrast to some commonly use nonlinear models (say, neural networks), fuzzy modeling delivers an interpretability feature, namely the structure of the model can be analyzed and interpreted. In particular, we elaborate on this aspect in the context of rule-based fuzzy models and fuzzy logic networks (Fig. 12.7).

Interestingly, when looking at digital processing, the models formed there exhibit the same general topology showing the three functional modules. The role of the input interface is to facilitate transformation of continuous data into the digital format acceptable to all faculties of digital processing. More specifically, we have the mapping realized by the analog-to-digital (A/D) converter and transforming a continuous variable to a strong of bits (input interface formed by a collection of intervals).

Processing of digital data is completed in the processing module (here depending upon an application area, a spectrum of advanced digital processing algorithms is involved). The results are returned through an D/A converter which serves as an output interface.

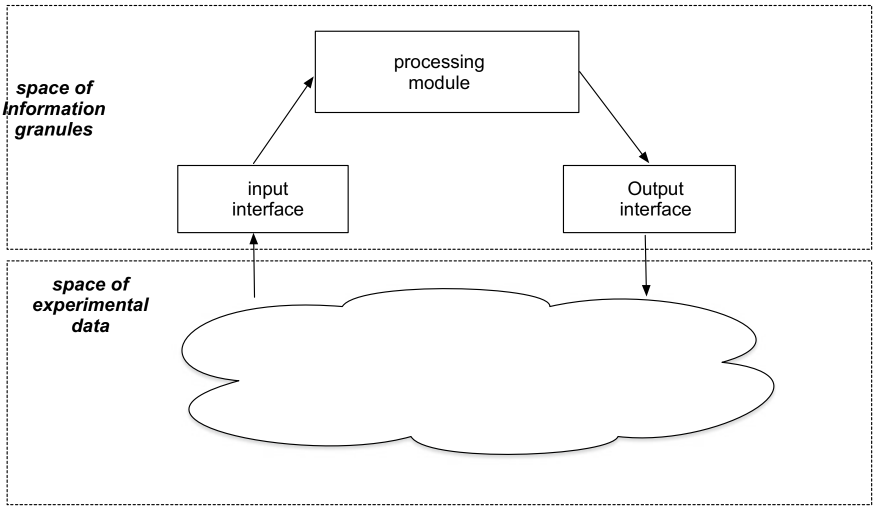


Fig. 12.7 A general architecture of a fuzzy model

12.2 The Design of Fuzzy Models

The development of the fuzzy model is viewed as an optimization problem; more specifically, a problem of supervised learning. Given is a training set composed of input-output pairs ($\boldsymbol{x}_k$, $target_k$), k=1, 2,…, N where $\boldsymbol{x}_k \in \boldsymbol{R}^n$ and $target_k \in R$ and a certain performance index Q, which quantifies the performance of the fuzzy model.

In the design, two main aspects are considered, viz. structural design and parametric design. The structural optimization, the most challenging phase, is about optimizing a structure of the model so that the given performance index Q becomes minimized. In virtue of the diversity of the architectures of the models, there are a vast number of topologies whose structures are formed. On the other hand, the parametric optimization is concerned with the adjustments of the parameters of the model. Structural design always precedes parametric design. The development process is iterative. We start with a simple structure (Occam razor principle) and based on the performance the model is made more sophisticated.

Owing to the higher level of abstraction at which the fuzzy model is built, two development strategies are considered: a so-called internal and external optimization are distinguished.

12.2.1 Internal Level of the Design

As the name stipulates, the optimization is focused at the abstract level viz. the level of fuzzy sets rather than numeric data. The essence of the design is outlined in Fig. 12.8.

In the ensuing optimization, one operates on the internal representation of the data viz. the strings located in the unit hypercubes, $\boldsymbol{z}_k$ in $[0,1]^c$ and $\boldsymbol{target}_k$ in $[0,1]^r$.

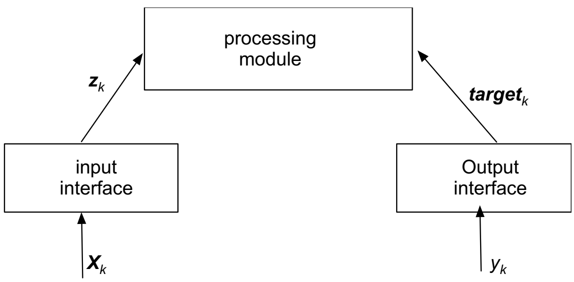


Fig. 12.8 Internal level of design of fuzzy models

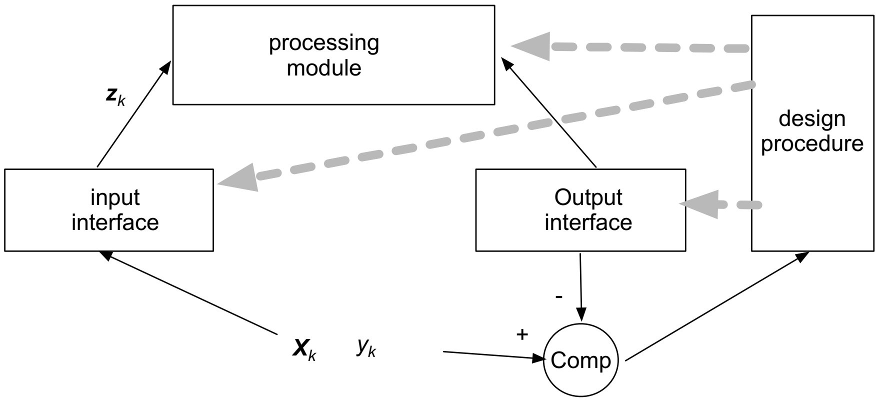


Fig. 12.9 External level of design of the fuzzy model

12.2.2 External Level of the Design

Here the performance of the fuzzy model is evaluated as a nonlinear mapping viz. one determines a sum of squared errors between y_k and $target_k$, $k = 1, 2, \ldots, N$. The optimization is carried out in the structure shown in Fig. 12.9.

The design of the model involves all functional module of the architecture. As such, the development process is more demanding than the one considered in the first scenario.

The development of the fuzzy model follows the same identification principles as any other models when it comes to the split of data in several subsets. Two commonly encountered options are considered

(i) split of data in training and testing data subsets. The development of the model is completed on the training data where usually this data set consists of 60–70% of all data. Next the performance of the model is assessed with the use of the testing data formed by the remaining 40–30% of data.
(ii) use of training, validation, and testing data subsets. Here additional validation set is formed which is used to navigate structural optimization of the model, say selecting a suitable number of rules in the rule-based model, etc.

The development of the model can be completed in a 10-fold cross validation mode meaning that the data are split 10 times into the training and testing part by randomly picking a subset of training data and testing data. In this way, some statistical analysis could be carried out and more credibility becomes associated with the model constructed in this way.

Example 2 Discuss situations when internal optimization is a sound design strategy and when it is not essential.

Internal optimization is of relevance when we anticipate that the model will be used in the user mode so the results are presented in the form of some fuzzy set.

However if we anticipate that the model has to deliver some numeric result, one has to focus on external optimization.

12.3 Categories of Fuzzy Models: An Overview

Fuzzy models form a highly diversified landscape of architectures and related development schemes. Our intent is to identify a role of the functional modules and show a role played by fuzzy sets. These models will be studied in detail in the subsequent chapters.

Rule-based models They are commonly used structures that are constructed in the form of "if condition-then conclusion" statements where conditions and/or conclusions are described by fuzzy sets [9]. Rules come as a generic scheme of knowledge representation discussed in the area of Artificial Intelligence [4, 11]. Information granularity of fuzzy sets forming conditions and conclusions plays a pivotal role by setting up a level of modularity of the model.

Two main categories of rules are identified, namely relational and functional rules. The relational rules read as follows

- if condition is A_i then conclusion is B_i

where A_i and B_i are fuzzy sets defined in the spaces of conditions and conclusions. The functional rule-based models are structured with the aid of the rules in the form [12].

- if condition is A_i then conclusion is f_i.

Instead of fuzzy sets B_i forming the conclusions, here we have some local (commonly linear) functions f_i: $\boldsymbol{R}^n \rightarrow R$. Rule-based models reflect the fundamental idea of splitting a problem into a family of subproblems and solving/modeling them to arrive at a solution to the overall problem.

Fuzzy decision trees are architectures where classification of any datum is achieved by traversing the tree until reaching a certain terminal node associated with a given class. Each node of the tree associates with some attribute while a finite number of attributes is associated with the edges leaving this node [2, 10]. As shown in Fig. 12.10, given an input datum described as a vector $[a_1\ b_2\ c_1\ d_2]$, we traverse the tree starting from its root following a path implied by the values of the attributes (a_1, b_2, d_2) until one of the terminal nodes has been reached, see Fig. 12.10a. The values of the attribute are intervals, for instance, a_1 is an interval.

There have been a number of design methods producing fuzzy decision trees and their refinements [6]. Fuzzy sets are used instead of intervals or sets of values at each node. The result of traversal (classification) is no longer a binary one (yes–no) but degrees of membership are accumulated, Fig. 12.10b. The classification result comes with the confidence level, which reflects the class membership grades. In this way, a certain flagging effect is produced and patterns with low classification confidence can be identified and eventually inspected in detail.

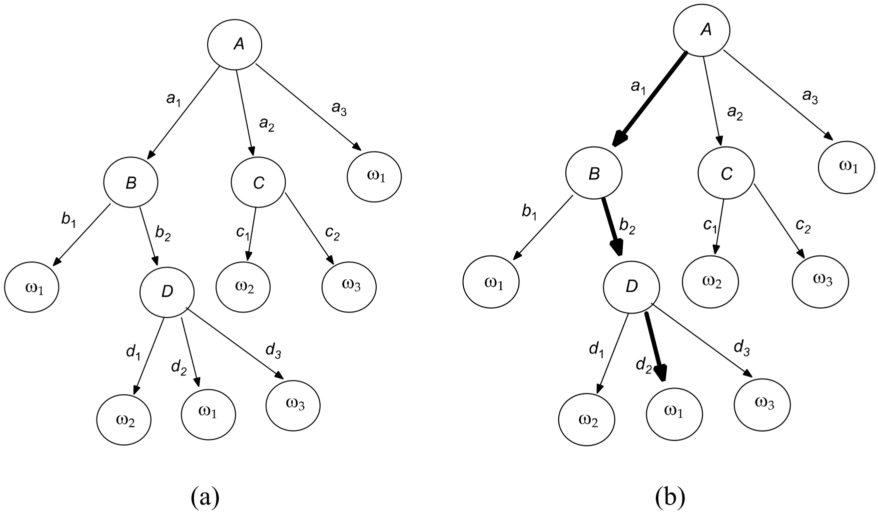


Fig. 12.10 Example of decision tree (**a**) and its traversal for input pattern $[a_1\ b_2\ d_2]$ (**b**)

Example 3 The constructed decision tree for a tree class problem with classes ω_1, ω_2, ω_3 is shown in Fig. 12.11. We classify the incoming patterns [3 2 6.2] and [−2 4 13]. For the first pattern, we traverse this tree reaching the node associated with class ω_2. Recall that a_is, b_is, c_is and d_is are intervals. For the second one the traversal of the tree yields class ω_3.

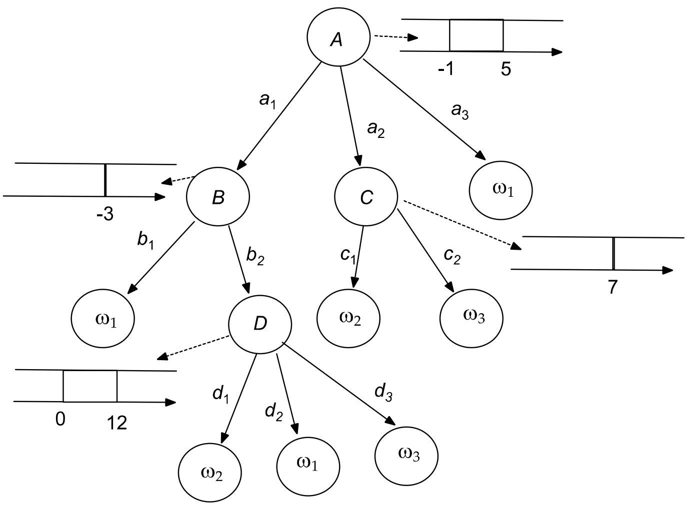


Fig. 12.11 Decision tree and its traversal by patterns

Example 4 Considering the fuzzy decision tree shown in Fig. 12.11 and assuming that the values of the attributes are described by fuzzy sets,

node A A_1, A_2, A_3
node B B_1, B_2
node C C_1, C_2
node D D_1, D_2, D_3

determine the classification result for the pattern characterized by (a', b', c', d').

In contrast to set-based attributes, here all nodes of the tree are activated to some degree and thus several terminal nodes are reached. The degrees of traversal the edges from A to B, C, and D are $A_1(a')\,A_2(a')$, and $A_3(a')$, respectively. Next the node D is reached at the degree $A_1(a')tB_2(b')$ where t stands for any t-norm. The level of reaching the terminal nodes are shown in Fig. 12.12b.

As several terminal nodes correspond to the same class, the results are combined with the use of some t-conorm. We obtain:

The degree of belongingness of (a', b', c', d') to class ω_1

$$\left[A_1\left(a'\right)tB_1\left(b'\right)\right]s\left[A_1\left(a'\right)B_2\left(b'\right)D_2\left(d'\right)\right]sA_3\left(a'\right)$$

The degree of belongingness of (a', b', c', d') to class ω_1

$$\left[A_1\left(a'\right)B_2\left(b'\right)D_1\left(d'\right)\right]s\left[A_2\left(a'\right)tC_1\left(c'\right)\right]$$

The degree of belongingness of (a', b', c', d') to class ω_1

$$\left[A_1\left(a'\right)B_2\left(b'\right)D_3\left(d'\right)\right]s\left[A_2\left(a'\right)C_2\left(c'\right)\right]$$

Fuzzy regression models The structure of these models is the same as in commonly encountered numeric regression [3] however the parameters of the model are described by fuzzy sets (fuzzy numbers) whose membership functions are triangular. The underlying formula for an n-input regression model is presented in the following way

$$Y = A_0 \oplus A_1 \otimes x_1 \oplus \cdots \oplus A_n \otimes x_n \tag{12.7}$$

where the symbols $\oplus$ and $\otimes$ being used here underline that the addition and multiplication consider non-numeric arguments. The parameters help account for noisy numeric data. In this structure, the input interface is non-existent while the results (Y) are expressed as fuzzy sets (because of the presence of the fuzzy sets of parameters), so the output interface is required.

Example 5 The fuzzy regression model is described in the following way

$$Y = [-3\;1\;1.5] \oplus [0.3\;0.6\;0.9] \otimes x_1 \oplus [0.3\;0.7\;1.4] \otimes x_2 \oplus [-1.1\;\;-0.6\;0.3] \otimes x_3$$

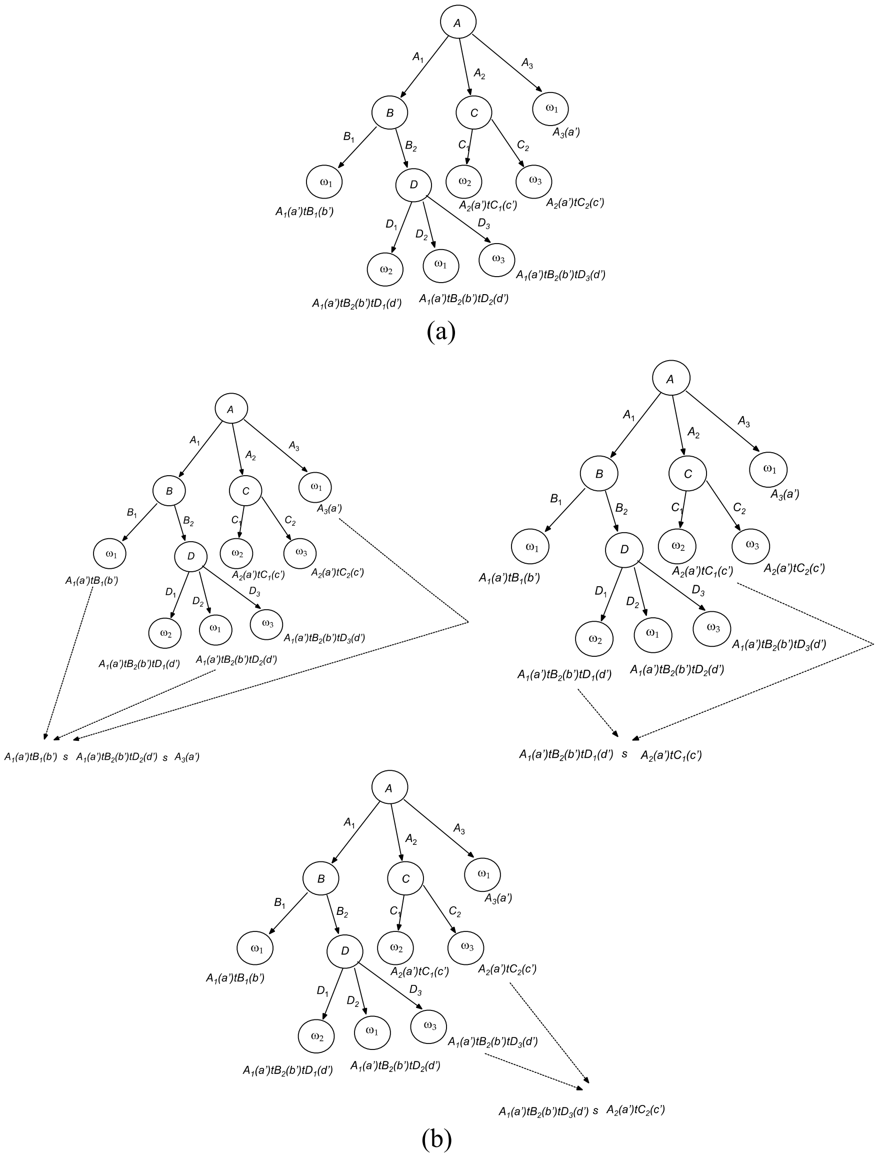


Fig. 12.12 Degrees of reaching terminal nodes of the decision tree: **a** levels of reaching terminal nodes, and **b** computing class membership grades

Determine output Y for $\boldsymbol{x} = [1.5\ 2.0\ {-3.1}]$.
Recalling the formulas of fuzzy arithmetic, we obtain

$$
\begin{aligned}
Y = {} & [-3\ 1\ 1.5] \oplus [0.3\ 0.6\ 0.9] \otimes 2 \oplus [0.3\ 0.7\ 1.4] \\
& \otimes 2 \oplus [-1.1\ -0.6\ 0.3] \otimes (-3.1) \\
& = [-3\ 1\ 1.5] \oplus [0.45\ 0.90\ 1.35] \oplus [0.6\ 1.4\ 2.8] \oplus [-0.93\ 1.86\ 3.41] \\
& = [-2.88 5.16 9.06]
\end{aligned}
$$

Example 6 Determine the output of the fuzzy regression model $Y = A \otimes X \oplus B$ for

$$A = [3,\ 4],\ B = [2,\ 5] \text{ and the input } X = [-3,\ 2].$$

The resulting granular output is computed as follows

$$
\begin{aligned}
Y = {} & [3, 4] \otimes [-3, 2] \oplus [2, 5] = [\min(-9, 6, -12, 8), \max(-9, 6, -12, 8)] \\
& \oplus [2,\ 5] = [-12,\ 8] \oplus [2,\ 5] = [-10,\ 13].
\end{aligned}
$$

Fuzzy associative memories are fuzzy models which concentrate on capturing relationships (associations) among pairs of items described as fuzzy sets. Consider that A and B are two fuzzy sets defined in the corresponding spaces. Associations are modeled by fuzzy relations. Formally we say that A and B are related (described as ARB) with R being a fuzzy relation.

Associative memories are bidirectional constructs meaning that one can recall B associated with A and alternatively recall A that is associated with B, Fig. 12.13.

The term association (linkage) occurs commonly in many applications; one can talk about associations existing between categories of movies and groups of movie goers (as studied in the area of recommender systems), signatures and photos of individuals. There are multiway associations which are represented by tensors (say, associations among signatures, audio traces, and photos). When designing associative memories, two main problems are studied: (i) how to store collections of pairs of associations for pairs (A_k, B_k), k=1, 2, …, N, (ii) how to realize a recall process viz. recall B for given A and vice versa, namely recalling A for given B.

With regard to these design tasks, optimization is concerned about maximizing capacity of the memory, viz. storing the largest number of pairs followed by a minimal recall error which delivers the result of recall being as close to the corresponding elements of the pairs of items stored in the memory. There are a number of different

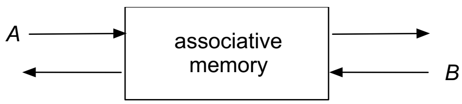


Fig. 12.13 Bidirectional recall realized by fuzzy associative memory

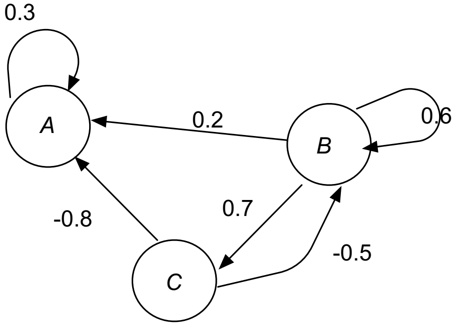


Fig. 12.14 Example fuzzy cognitive map

generalizations of the maps; one can refer to studies reported in Belohlavek [1], Li et al. [8], Sussner and Schuster [13], Valle and Sussner [14].

Fuzzy cognitive maps In contrast to other models where input and output varaibles are clearly identified, the maps are examples of graph models with no specification of the nature of the variables [7]. One views a system as a collection of concepts and associations among them. In graph notation, a fuzzy cognitive map is composed of nodes representing individual concepts while the directed edges describe linkages among the nodes (concepts). The edges point at causal relationships between the concepts, say concept *a* causes concept *b*. Refer to Fig. 12.14. The linkage between two nodes is quantified by its strength. The strength of impact of one node on another is quantified in the range $[-1,1]$ where 1 stands for a strong excitatory interaction (one concept excites another one) and -1 denotes inhibitory interaction. The values of the linkage around zero indicate that there is no relationship between the concepts.

Example 7 Propose a cognitive map describing the dynamics of a town where the following five concepts are considered: population, taxes, revenue, business, infrastructure spending.

The concepts interact among themselves as illustrated in Fig. 12.15.

The polarity of the connections is self-explanatory. For instance, increasing population promotes more infrastructure spending and vice-versa: higher infrastructure spending implies more people to live there (population increase). Increase of population leads to higher taxes yet increase in taxes has an inhibitory influence on the population. Note that the strengths of links are not symmetric: there might be a stronger impact of increasing taxes on the population than the link in the opposite direction. Weak positive impact of increased revenue of infrastructure spending could be a sign of some stagnation and poor management.

The example weights are also shown; they are typically a result of some learning.

Example 8 An ideal gas is contained in an adjustable container, Fig. 12.16. If we know that two concepts such as pressure (p) and volume (v) are related in the form $pv =$ constant, describe a behavior of gas in the form of a fuzzy cognitive map.

We can envision a fuzzy cognitive map composed of two nodes describing the concept of pressure and volume, Fig. 12.16. There is an inhibitory linkage: increase

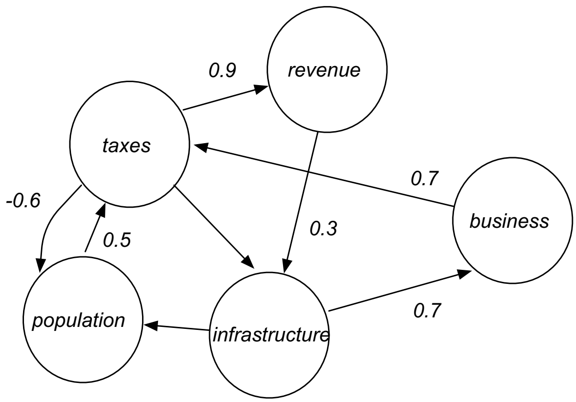


Fig. 12.15 Fuzzy cognitive map of town dynamics

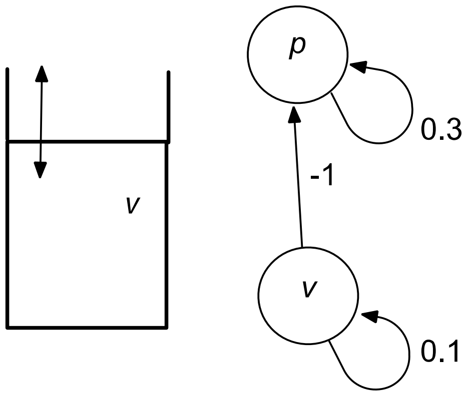


Fig. 12.16 Example of fuzzy cognitive map modeling pressure-volume relationships

in volume implies lower pressure. Each concept supports itself (positive weights) which reflects a certain level of inertia when changes occur.

Fuzzy neural networks are hybrid constructs positioned at the junction of the technology of fuzzy sets and neurocomputing. Those two conceptual and algorithmic streams forming a backbone of Computational Intelligence dwell upon the synergy of neurocomputing characterized by learning abilities and knowledge representation. Neural networks deliver learning capabilities with the dominant agenda of constructing learning algorithms. Neural networks are black boxes. Fuzzy sets lead to interpretable constructs (say, rule-based models) but they do not focus on learning. The highly anticipated synergy helps alleviate existing limitations by creating hybrid topologies. There is a diversity of architectures as well as hybrid learning schemes. An example is shown in Fig. 12.17. Here the synergy is apparent: the individual neurons are so-called logic neurons, *and* and *or* neurons whose processing uses logic operators (*t*-norms and *t*-conorms). Hence the topology of the system is the one commonly seen in neural networks while processing is typical for fuzzy set processing. The learning is possible and efficient learning schemes as those studied in

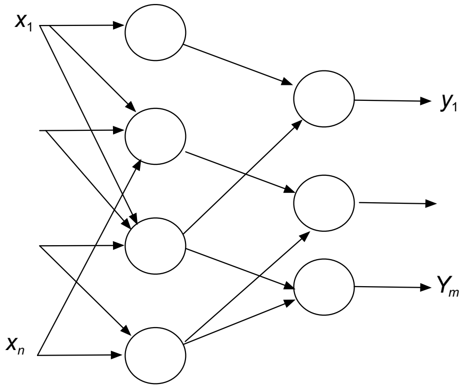


Fig. 12.17 Example of n-input m-output fuzzy neural network composed of *and* and *or* neurons

neural networks design. The resulting network obtained after learning is interpretable because of the logic nature of the neurons. When propagating any inputs through the network, the relationship between the obtained output is a logic expression of the inputs.

The role of fuzzy sets varies from one category of the models to another. In some situations, fuzzy sets are used to model parameters of the existing numeric counterpart such as this was present in case of fuzzy regression built on a basis of regression models or in fuzzy decision trees. In some other cases fuzzy sets bring forward a new architecture such as encountered in case of relational models, say fuzzy associative memories or fuzzy neural networks augmenting the existing topologies and functionalities present in neurocomputing (Fig. 12.18).

Example 9 Discuss some practical examples where the model can be represented as (i) function and (ii) relation.

If we are faced with systems whose behavior is governed by some laws of physics or chemistry then the relationships among the variables present in such systems, are likely to be modeled in the form of some linear or nonlinear functions. Note that we might know the detailed law but at least envision there is some dependency of this type. If there are no known laws describing the dependencies, the use of relations

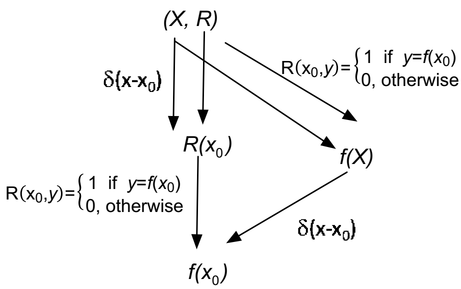


Fig. 12.18 Information granularity at the level of inputs and structural dependencies

is more suitable here. This happens in a broad range of social, economical, political systems, say social networks, recommender systems and alike.

12.4 Searching for a Structure of the Model: Striking a Sound Bias-Variance Trade-off

More advanced (complex) models exhibiting a large number of parameters usually produce lower values of the performance index Q reported on a training set but this comes with higher values of Q reported on the testing set. The phenomenon is associated with the bias-variance dilemma [5].

Consider that the system of interest is governed by the expression $f(\boldsymbol{x}) + \varepsilon$ where ε is an additive random noise with a zero mean value and some variance σ_{ε}^{2}. The constructed model is $\hat{f}(\boldsymbol{x})$.

Let us calculate the prediction result produced by the model for some value of the input denoted by $\boldsymbol{x}_0$. The quality of the prediction result is taken as the following expected value of the squared prediction error

$$E\left[\left(y - \hat{f}(\boldsymbol{x}_0)\right)^2\right] = E\left[\left(f(\boldsymbol{x}_0) + \varepsilon - \hat{f}(\boldsymbol{x})\right)^2\right] \tag{12.8}$$

where $E(.)$ is an operator of expected value. Let us complete the detailed calculations

$$\begin{aligned}
E\left[(y - \hat{f}(\boldsymbol{x}_0)^2\right] &= E\left[\left(f(\boldsymbol{x}_0) + \varepsilon - \hat{f}(\boldsymbol{x}_0)\right)^2\right] \\
&= E\left[\varepsilon^2 + 2\varepsilon\left(f(\boldsymbol{x}_0) - \hat{f}(\boldsymbol{x}_0)\right) + \left(f(\boldsymbol{x}_0) - \hat{f}(\boldsymbol{x}_0)\right)^2\right] \\
&= E(\varepsilon^2) + E\left[\left(f(\boldsymbol{x}_0) - \hat{f}(\boldsymbol{x}_0)\right)^2\right]
\end{aligned} \tag{12.9}$$

We restructure the above expression as follows

$$\begin{aligned}
E[(f(\boldsymbol{x}_0) - \hat{f}(\boldsymbol{x}_0))^2] &= E[(f(\boldsymbol{x}_0) - E\hat{f}(\boldsymbol{x}_0) + E\hat{f}(\boldsymbol{x}_0) - \hat{f}(\boldsymbol{x}_0))^2] \\
&= E[(f(\boldsymbol{x}_0) - E\hat{f}(\boldsymbol{x}_0))^2 + (E\hat{f}(\boldsymbol{x}_0) - \hat{f}(\boldsymbol{x}_0))^2 - 2(f(\boldsymbol{x}_0) \\
&\quad - E\hat{f}(\boldsymbol{x}_0))(\hat{f}(\boldsymbol{x}_0) - E\hat{f}(\boldsymbol{x}_0))] = E[(f(\boldsymbol{x}_0) - \hat{f}(\boldsymbol{x}_0))]^2 \\
&\quad + E[(\hat{f}(\boldsymbol{x}_0) - E\hat{f}(\boldsymbol{x}_0))^2] - 2E[(f(\boldsymbol{x}_0) - E\hat{f}((\boldsymbol{x}_0))(\hat{f}(\boldsymbol{x}_0) \\
&\quad - E\hat{f}(\boldsymbol{x}_0))] = [f(\boldsymbol{x}_0) - E\hat{f}(\boldsymbol{x}_0)]^2 + E[(\hat{f}(\boldsymbol{x}_0) - E\hat{f}(\boldsymbol{x}_0))^2]
\end{aligned} \tag{12.10}$$

Introduce the notation

$$bias = E\left(\hat{f}(\boldsymbol{x}_0) - f(\boldsymbol{x}_0)\right)$$
$$variance = E\left[\left(\hat{f}(\boldsymbol{x}_0) - E\hat{f}(\boldsymbol{x}_0)\right)^2\right] \tag{12.11}$$

Next we have

$$E\left[\left(y - \hat{f}(\boldsymbol{x}_0)\right)^2\right] = \sigma_\varepsilon^2 + bias^2 + variance \tag{12.12}$$

The above expression is composed of the three components: the first one is an irreducible error, the second one is a squared bias and the third one is the variance of the model.

Formula (12.12) constitutes an essence of a bias-variance dilemma. Note that given value of $E\left[\left(y - \hat{f}(\boldsymbol{x}_0)\right)^2\right]$, it can be distributed between the second and the third term. Increasing bias reduces the variance and vice versa: lowering bias comes with the higher variance. In terms of the structure of the model, its higher complexity (larger number of parameters, in a particular the larger number of rules in the fuzzy rule-based model) brings higher variance but reduces the bias. A smaller number of parameters of the model is a contributor to the increasing bias. In terms of the training and testing data, the typical relationship between the squared error and the complexity P of the model (say, the number of parameters of the model, for instance the number of rules) is shown in Fig. 12.19.

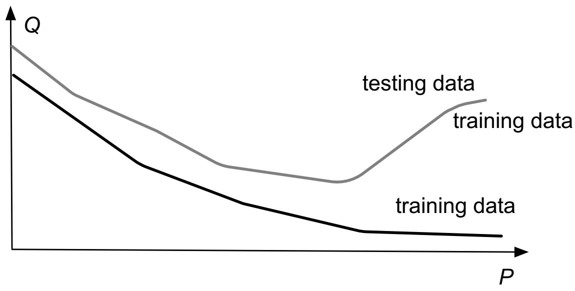
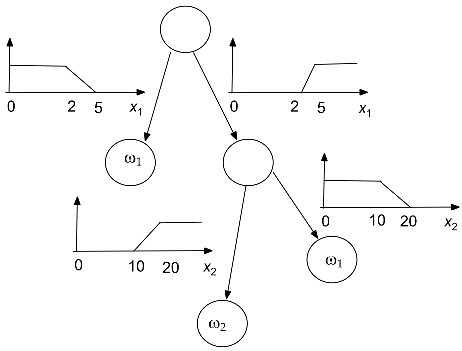


Fig. 12.19 Error for the training and testing data regarded as a function of the complexity of the model (number of parameters of the model)

Fig. 12.20 Fuzzy decision tree



12.5 Conclusions

We showed that fuzzy models are modeling constructs that inherently involve fuzzy sets in their design. Two key modes of usage (numeric and granular) are emphasized along with the associated construction strategy of external and internal optimization. The plethora of architectures is presented by highlighting the key features of their underlying architectures and the role of fuzzy sets played in the underlying structure.

Problems

1. A fuzzy decision tree is shown in Fig. 12.20.

   (i) Classify the patterns $\boldsymbol{x} = [2.5\ 1.6]$, $\boldsymbol{x} = [5.0\ 1.1]$, and $\boldsymbol{x} = [1.5\ 0.0]$

   (ii) Determine the region in the (x_1, x_2) space in which patterns belong to class ω_1. Consider that the t-norm is taken as the product while the t-conorm is the maximum operator.

2. In the decoding (defuzzification) the result is considered as a solution to the optimization problem

$$\int_R A(x)|x - x_0|dx$$

   What is the advantage of this decoding method in comparison with other decoding algorithms.

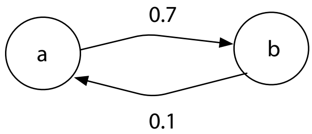
3. Fuzzy set A is defined as shown below

x	−3	−2.5	−1.0	0.0	0.6	1.7	3.6	5.0	5.5
$A(x)$	0.3	0.5	0.7	1.0	0.9	0.5	0.3	0.1	0.1

Determine the result of decoding x_0 for $\gamma = 0$, 1.0, and 0.5.

Fig. 12.21 Distribution of one-dimensional in input space

Fig. 12.22 Fuzzy cognitive map with two concepts



4. Determine the result of decoding x_0 for which minimizes the following performance index

$$Q = \int_R (A(x) t B(x))(x - x_0)^2 dx + \int_R (A(x) s B(x))(x - x_0)^2 dx$$

where A and B are fuzzy sets defined in the space of real numbers. Complete the detailed calculations for triangular fuzzy sets $A(x; -1, 3, 7)$ and $B(x; 3,5,6)$ and the product and probabilistic sum.

5. Alluding to Example 1, what nonlinear transformation of input space would you consider by proposing a single fuzzy set for the data belonging to two classes and visualized Fig. 12.21.
6. Elaborate on linkages between decision trees and rule-based systems.
7. When the depth of the decision tree increases, what happens to the rules one derives from the tree.
8. Complete detailed calculations for the tree presented in Fig. 12.20 when $\boldsymbol{x} =$ [3.5 15]. Assume that *t*-norm is realized as the Lukasiewicz *and* operator and *t*-conorm is the Lukasiewicz *or* operator.
9. Give an example of a situation when causal dependencies are depicted in the form of the following fuzzy cognitive map, Fig. 12.22.

References

1. Bělohlávek, R.: Fuzzy logical bidirectional associative memory. Inf. Sci. **128**, 91–103 (2000)
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth, Belmont, CA (1984)
3. Draper, N.R., Smith, H.: Applied Regression Analysis. Wiley, New York (1998)
4. Dougherty, E., Giardana, C.R.: Mathematical Methods of Artificial Intelligence and Autonomous Systems. Prentice Hall, Englewood Cliffs (1988)
5. Hastie, T., Tibisharini, R., Friedman, J.: The Elements of Statistical Learning. Data Mining, Inference, and Prediction, 2nd edn. Springer, New York (2009)
6. Janikow, C.Z.: Fuzzy decision tree: Issues and methods. IEEE Trans. Syst. Man Cybern. Part B Cybern. **28**, 1–14 (1998)

7. Kosko, B.: Neural Networks and Fuzzy Systems. Prentice Hall, Englewood Cliffs, NJ (1992)
8. Li, L., Pedrycz, W., Qum, T., Li, Z.: Fuzzy associative memories with autoencoding mechanisms. Knowl. Based Syst. **191**(5) (2020)
9. Ojha, V., Abraham, A., Snášel, V.: Heuristic design of fuzzy inference systems: a review of three decades of research. Eng. Appl. Artif. Intell. **85**, 845–864 (2019)
10. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**, 81–106 (1986)
11. Shukla, A.K., et al.: Engineering applications of artificial intelligence: a bibliometric analysis of 30 years (1988–2018). Eng. Appl. Artif. Intell. **85**, 517–532 (2019)
12. Sugeno, M., Kang, G.T.: Structure identification of fuzzy model. Fuzzy Sets Syst. **28**, 15–33 (1988)
13. Sussner, P., Schuster, T.: Interval-valued fuzzy morphological associative memories: Some theoretical aspects and applications. Inf. Sci. **438**, 127–144 (2018)
14. Valle, M.E., Sussner, P.: A general framework for fuzzy morphological associative memories. Fuzzy Sets Syst. **159**, 747–768 (2008)

Chapter 13
Fuzzy Rule-Based Models

Abstract The chapter is focused on the fundamentals of fuzzy rule-based models, their key properties, design and development.

13.1 The Principles of Rule-Based Computing

Fuzzy rule-based models and modeling are the most visible area of system modeling present within a broad spectrum of fuzzy models. Rules are the prominent way of knowledge representation commonly encountered in Artificial Intelligence. The fundamental paradigm of modularity (where a complex problem is prudently decomposed into a series of subproblems) is supported in system modeling as each rule focuses on local modeling of the entire complex system.

Rules are "if-then" statements composed of conditional statements linking some condition (antecedent) with the associated conclusion. Rules come hand in hand with information granules as conditions and conclusions are commonly represented in the form of information granules, and fuzzy sets, in particular. Fuzzy sets are pieces of knowledge and the calculus of fuzzy sets delivers the operational framework to

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_13

elicit knowledge about the problem, represent, process the knowledge and produce conclusion.

Rule-based fuzzy models exhibit surprisingly rich diversity. A general taxonomy of rule-based models involves two main categories:

relational rules The rules assume a general format

$$-\text{if condition is } A_i \text{ then conclusion is } B_i \tag{13.1}$$

$i = 1, 2, \ldots, c$ and conclusions and conditions are fuzzy sets defined in the input space. The term *relational* reflects the fact that each rule captures a relationship between fuzzy sets present in the condition and conclusion part.

functional rules The rules exhibit the following format

$$-\text{ if condition is } A_i \text{ then conclusion is } f_i(\boldsymbol{x}; \boldsymbol{a}_i) \tag{13.2}$$

$i = 1, 2, \ldots, c$ and conditions are fuzzy sets defined in the input space while the conclusion is a local function being a mapping from the input to the output space $f_i : \boldsymbol{x} \varepsilon\, \boldsymbol{R}^n \rightarrow \boldsymbol{R}$ and $\boldsymbol{a}_i$ is a vector of parameters of the local function.

13.2 Relational Rule-Based Models

As categorized above, the relational rules capture locally present relationships between fuzzy sets in the input and output space [6]. The rules can exhibit different semantics depending upon the form of the relationship. To illustrate the essence of rule-based modeling, consider single input-single output data displayed in Fig. 13.1. The cloud of data can be covered by a collection of patches where each patch represents some rule. The conditions and conclusions come with fuzzy sets exhibiting a clearly visible semantics. Alluding to the data, there are three rules read as follows

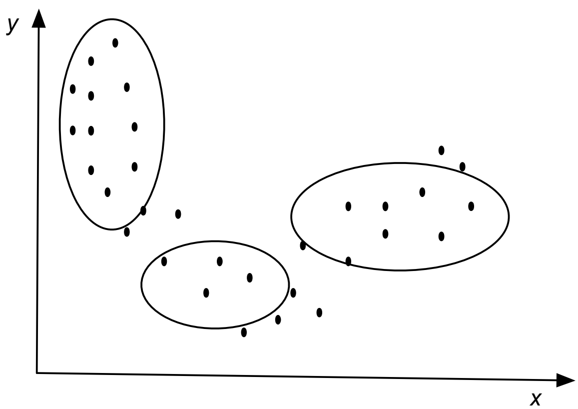


Fig. 13.1 One-dimensional input-output data and patches delivered by relational rules

– if x is $small$ then y is $l\arg e$
– if x $is\ medium$ then y is $small$
– if x is $l\arg e$ then y is $medium$

It becomes apparent that even though the data have some nonlinear character the rules are quite simple. Also, the patches are located over the data without too much hesitation to reflect the cloud of data. Obviously in case of multidimensional data, the formation of rules becomes more challenging although for several (2–3) input variables, it is still doable.

In what follows, we look at the underlying processing, which predominantly engages calculus of fuzzy sets and relations along with the pertinent composition operators.

13.2.1 Representation of Rules

Each patch is built by taking a Cartesian product of fuzzy sets of condition and conclusion. This yields

$$R_i = A_i \times B_i$$
$$R_i(x, y) = \min(A_i(x), B_i(y)) \tag{13.3}$$

There are several patches coming with each rule. One takes a union of all the patches to cover all data,

$$R = R_1 \cup R_2 \cup \ldots \cup R_c$$
$$R(x,\ y) = \max_{i=1,2,\ldots,c} R_i(x, y) \tag{13.4}$$

Processing is about inferring for any input A the corresponding output B. This is completed by taking the sup-min (max-min) composition of A and R

$$B = A \circ R \tag{13.5}$$

Recall that the membership function of B comes in the form

$$B(y) = \sup_x(\min(A(x), R(x,\ y)) \tag{13.6}$$

An illustration of the processing is presented in Fig. 13.2.

Let us re-organize the computing by bringing explicitly the membership functions

$$B(y) = \sup_x\left[\min\left(A(x),\ \max_{i=1,2,..,c}(\min(A_i(x), B_i(y)))\right)\right]$$

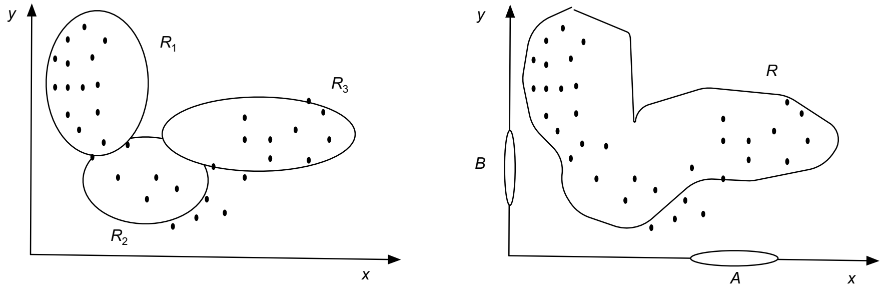


Fig. 13.2 Construction of fuzzy relations and computing conclusion B for given condition A

$$= \max_{i=1,2,..,c} \left\{ \min(B_i(y)),\ \sup_x[\min(A(x), A_i(x)]) \right\} \tag{13.7}$$

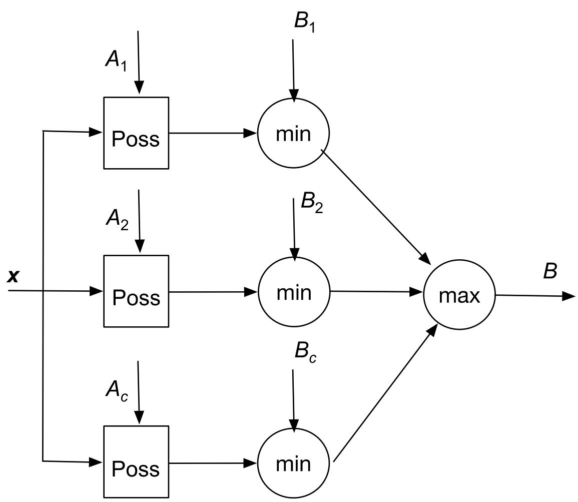
Denote by λ_i an overlap between A and A_i; it just a possibility measure of A and A_i. Next one proceeds with the following expression

$$B(y) = \max_{i=1,2,..,c} [\min(\lambda_i, B_i(y))] \tag{13.8}$$

In terms of rules, the possibility measure is a degree of firing the i-th rule caused by A_i. The above expression can be interpreted as a union of conclusions B_i truncated by the corresponding activation levels λ_i. If for some rule i, $\lambda_i = 1$ then B_i fully contributes to the conclusion. In contrast, if $\lambda_i = 0$, there is no contribution of B_i to the generated result.

The flow of processing can be structured as illustrated in Fig. 13.3. There are two main processing phases: rules are fired (activated) producing levels of activation

Fig. 13.3 Overall flow of processing realized in rule-based modeling



$\lambda_1, \lambda_2, \ldots, \lambda_c$ and a fuzzy set of conclusion is formed by combining fuzzy sets of conclusion and the activation levels.

Two interesting properties of the processing are to be noted:

(i) The result is reflective of the degrees of firing of the rules; several rules could be fired in parallel. The structure exhibits an evident level of parallelism.
(ii) modularity of the structure is present. Adding new rules to the existing set is immediate. For instance, with c rules resulting in R and adding $c + 1$ rule one has

$$R' = R \cup (A_{c+1} \times B_{c+1}) \tag{13.9}$$

and

$$B'(y) = \max(B(y),\ \min(B_{c+1}(y), \lambda_{c+1})) \tag{13.10}$$

Owing to the form of the composition operator, the boundary conditions of the inference procedure are considered for A being an empty set $A(x) = 0$ for all x and forming an entire space; $A(x) = 1$ for all x. In the first case B is also an empty set. In the second boundary case $B(y) = \max_{i\,=\,1,2,\ldots\,c} B_i(y)$.

The resulting fuzzy set B has a clear interpretation. If A is unknown ($A(x)$ being identically equal to 1) then the only sound result we can infer is a union of B_is. In this case the lack of specificity is apparent.

The extension to the multi-input rules is straightforward. Consider a two-input rules

$$-\text{if } x \text{ is } A_i \text{ and } z \text{ is} C_i \text{ then } y \text{ is } B_i \tag{13.11}$$

with the fuzzy sets defined in the corresponding spaces of conditions and conclusions. The fuzzy relation is now a tensor–a three-way Cartesian product

$$\begin{aligned} &R_i = A_i \times C_i \times B_i \\ &R_i(x,\ z,\ y)\ =\ \min(A_i(x), C_i(z), B_i(y)) \end{aligned} \tag{13.12}$$

The aggregation of fuzzy relations is completed in the same way as before

$$R(x,\ z,\ y)\ =\ \max_{i=1,2,\ldots,c} R_i(x, z,\ y) \tag{13.13}$$

The conclusion is obtained through the sup-min composition

$$B\ =\ (A \times C) \circ R \tag{13.14}$$

Recall that the membership function of B comes in the form

$$B(y) = \sup_{x,\,z}(\min(A(x), C(z), R(x,\ z,\ y)) \tag{13.15}$$

13.2.2 Types of Rules

Depending upon the nature of the problem and properties of the domain knowledge, rules may come in the different formats, such as certainty-qualified and gradual rules.

In the certainty-qualified rules, instead of allocating full confidence in the validity of the rules, we allow to treat them as being satisfied (valid) at some level of confidence. The degree of uncertainty leads to certainty-qualified expressions of the following form

$$\text{if } X \text{ is } A \text{ and } Y \text{ is} B \text{ then } Z \text{ is} C \text{ with certaint } y\ \mu \tag{13.16}$$

where $\mu \in [0, 1]$ denotes the degree of certainty of this rule. If $\mu = 1$, we say that the rule is certain.

Rules may also involve gradual relationships between objects, properties, or concepts. For example, the rule

$$\text{the}\, more\ X \text{ is } A;\ \text{the}\, more\ Y \text{ is } B \tag{13.17}$$

expresses a relationship between changes in Y triggered by the changes in X. In these rules, rather than expressing some association between antecedents and consequents, we capture the tendency between the information granules; hence the term of graduality occurring in the condition and conclusion part. For instance, the graduality occurs in the rule “the *higher* the income, the *higher* the taxes” or “typically the *higher* the horsepower, the *higher* the fuel consumption”, the *lower* the temperature, the *higher* heating bills”.

13.2.3 Quality of Rules

The quality of the rules can be assessed by analyzing the specificity of conditions and conclusions; Fig. 13.4 visualizes several typical situations.

Highly specific conditions and conclusions imply low quality of the rules. For instance, the rule

$$-\text{if}\, x \text{ is } 3.2 \text{ then } y \text{ is } -1.5 \tag{13.18}$$

comes with a numeric condition so it is only applicable to this particular situation; the rule exhibits a very low applicability. The rule with specific condition and low

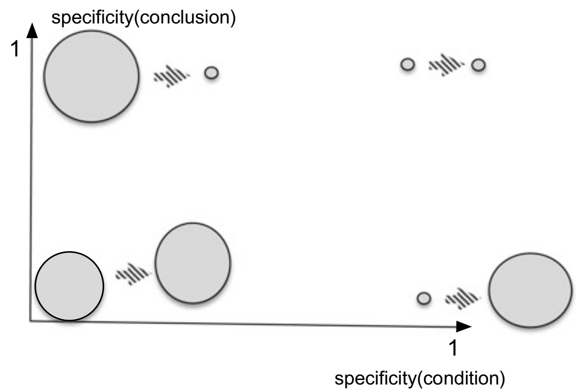


Fig. 13.4 Visualization of rules in the specificity coordinates of conditions and conclusions

specificity, say

$$-\text{if } x \text{ is } 3.2 \text{ then } y \text{ is } \textit{more or less negative large} \tag{13.19}$$

exhibits low quality; again the rule applies only for a single numeric input. The rule with conditions and conclusion of limited specificity, say

$$-\text{if } x \text{ is } \textit{negative medium} \text{ then } y \textit{ is positive small} \tag{13.20}$$

is general enough in terms of the condition and specific in terms of the conclusion so its quality is good. The highest quality is noted for the rule in which the specificity of the condition is as low as possible and the conclusion is highly specific. An example is the rule in the form

$$-\text{ if } x \text{ is } \textit{more or less medium} \text{ then } y \text{ is } \textit{about } 4.5 \tag{13.21}$$

The following ratio of specificity of the condition and conclusion can be sought as a measure of relevance (usefulness) of the rule

$$\mathrm{rel}(A_i, B_i) = \mathrm{sp}(B_i)/(\mathrm{sp}(A_i) + \varepsilon) \tag{13.22}$$

where sp(.) assumes values in [0,1]; $\varepsilon \approx 0$ standing in the denominator is used to assure that there is no division by zero. The higher the value of (13.22), the better the quality of the rule.

Example 13.1 Given are two rules

$$- \text{ if } x \text{ is } A \text{ then } y \text{ is } B$$
$$- \text{ if } x \text{ is } A' \text{ then } y \text{ is } B'$$

where A, B, A; and B' are triangular fuzzy sets $A(x; -1, 1, 5)$, $A'(x; 3, 6, 15)$, $B(y; -5, 1, 6)$ and $B'(y; -3, 1, 4)$. The input space is an interval [− 20, 20] and the output space is [− 10, 10]. Evaluate the relevance (usefulness) of these rules.Recall that the specificity of a triangular fuzzy set $A(x; a, m, b)$ is $1 - (b - a)/range$. Completing the calculations we obtain sp(A) = 1 − 6/80 = 0.925, sp(A') = 1 − 12/80 = 0.85, sp(B) = 1 − 11/40 = 0.725, sp(B') = 1 − 7/40 = 0.825. Next rel(A, B) = sp(B)/sp(A) = 0.725/0.925 = 0.783, sp(B')/sp(A') = 0.825/0.85 = 0.97 (the value of ε is set to 0 as the specificity of the condition is nonzero). The second rule is characterized by the higher relevance.The rules are evaluated with respect to their completeness and consistency. These two features are essential in the assessment of the quality of any rule-based model and need to quantified in case the rules are elicited by experts. The domain knowledge might exhibit some gaps (which is quantified by the property of incompleteness) and inconsistencies.

completeness We say the rules are incomplete if the rules do not "cover" the entire input spaces. The rules "if x is A_i then y is B_i" are sought incomplete if there are some inputs x for which none of the rules fires. This occurs if A_is do not cover the entire input space. Both the detection and alleviation of this problem is straightforward; new rules have to added to eliminate the gaps.

With regard to the larger number of input variables (n) where the rules assume the following format

$$-if\, x_1 \text{ is } A_i \text{ and } x_2 \text{ is } B_i \text{ and } \ldots\ x_n \text{ is } W_i \text{ then conclusion is } Z_i$$

the completeness property goes hand-in-hand with the curse of dimensionality. If all fuzzy sets standing in the conditions of the rules have a finite support (say, all of them have triangular membership functions) and the number of fuzzy sets is c, the number of rules is c^n; this number increases rapidly when the number of variables increases. Even for several number of input variables, say $n = 4$ and $c = 5$ fuzzy sets one ends up with $5^4 = 25*25 = 625$ rules

inconsistency This property characterizes a situation when two rules have similar conditions but differ substantially in their conclusions. For instance, the rules

$$-\text{ if } x \text{ is small then } y \text{ is } l\arg e$$
$$-\text{ if } x \text{ is } very\ small \text{ then } y \text{ is } small$$

while activated by some x_0 invoke a fuzzy set that is evidently bimodal "*large or small*" which is confusing from the perspective of any follow-up decision making. Indeed if $\lambda_1 = \lambda_2$ then $B(y) = \max(B_1(y), B_2(y))$. The level of inconsistency could be quantified by determining the closeness of conditions and conclusions and expressing the relationship between them. Denote the closeness (similarity) measure assuming values in the unit interval by sim(A_i, A_j) and sim(B_i, B_j). The inconsistency level *incons* is introduced with the use of the f-operator implied by some t-norm

$$\text{incons}\big((A_i,\ B_i),\ (A_j,\ B_j)\big) = \sup\{c \in [0, 1] |\ \text{sim}(A_i, A_j) tc \le \text{sim}(B_i, B_j)\} \tag{13.23}$$

The higher the value of the index (13.23), the higher level of inconsistency is reported. In case of c rules, they can be arranged so that the most inconsistent rule can be identified and the problem rectified. The alleviation of the problem is completed by bringing an additional condition to the rule that reduces a level of overlap.

For instance, the rules

$$- \text{if the weather is } good \text{ drive } fast$$
$$- \text{if the weather is } good \text{ drives } lowly$$

are evidently in conflict. The conflict disappears with the involvement of an additional condition in one of these rules

$$-\text{if the weather is } good \text{ and there is } heavy \text{ traffic, drive slowly}$$

13.3 Functional Rule-Based Models

This category of the rules is focused on realizing modeling through a collection of simple local functions whose application is restricted to some area of the input space specified by the corresponding fuzzy sets defined in this space. These rules realize the concept of decomposition: a complex problem is split into a collection of local subproblems for which solutions (models) are easier to determine.

The rules assume the following form [5]

$$-\text{if } \boldsymbol{x} \text{ is } A_i \text{ then } y = f_i(\boldsymbol{x}; \boldsymbol{a}_i) \tag{13.24}$$

$I = 1, 2, \ldots, c$. Here A_i is an information granule (fuzzy set) defined in the n-dimensional input space $\boldsymbol{R}^n$. The function f_i standing in the conclusion part of the rule is a local function; quite commonly it is regarded as a linear function. A_i is a vector of parameters of the i-th function. The crux of rule-based modeling is to realize the model through a collection of simple locally applicable models. Consider Fig. 13.5. The nonlinearly distributed data can be conveniently described by three rules where each of them is focused only on a portion of the data falling within the scope of A_i and for which a simple linear model f_i is a sound description.

The main difference between relational and functional models lies in the way in which the conclusion is described; here we are concerned with local functions instead of fuzzy sets.

As in case of relational models, the processing is composed of the two main phases:

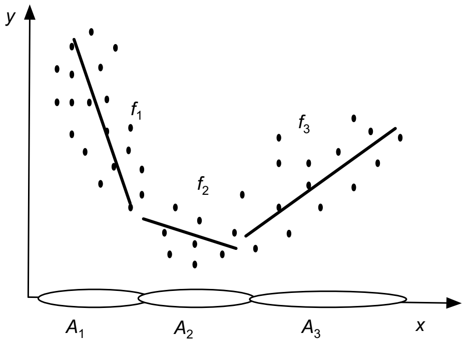


Fig. 13.5 Modeling data with the use of the three functional rules

(i) matching. For each rule one determines a level of firing $A_i(\boldsymbol{x})$
(ii) aggregation. The results of firing are aggregated in an additive way. The weighted sum with $A_i(\boldsymbol{x})$ serving as weights is computed

$$\hat{y}=\sum_{i=1}^{c} A_i(\boldsymbol{x}) f_i(\boldsymbol{x}; \boldsymbol{a}_\mathrm{i}) \tag{13.25}$$

The functional rules exhibit a great deal of diversity; here we list several selected examples

classification rules Classification rules in the form describing an allocation of data to one of the classes $\{\omega_1, \omega_2, \ldots, \omega_p\}$

$$-\text{if } \boldsymbol{x} \text{ is } A_i \text{ then } y = \omega_i \tag{13.26}$$

dynamical model The conclusion parts capture the dynamics of the system in the form of differential equations governing the behavior of the system in the subspace defined by A_i

$$-\text{if } \boldsymbol{x} \text{ is } A_i \text{ then } d\mathbf{x}/dt = f_i(\boldsymbol{x}; \boldsymbol{a}_i) \tag{13.27}$$

From the architectural viewpoint, the rule-based model can be implemented in the form of a so-called functional neuron (the connections of the neurons are local functions forming the conclusions of the corresponding rules (Fig. 13.6).

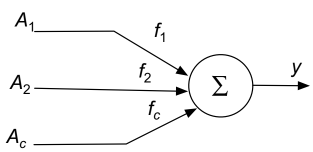


Fig. 13.6 Functional rules forming a functional neuron

13.3.1 Input-Output Characteristics of Rule-Based Model

Consider single input–single output rules with the conditions described by triangular membership functions with 1/2 overlap between two successive fuzzy sets. If the conclusions are constants, the input-output relationship is piecewise linear, see Fig. 13.7.

Example 13.2 Show that in rule-based model with the rules

$$-\text{if } x \text{ is } A_i \text{ then } y = P_i(x; a_i)$$

where A_i are triangular membership functions with ½ overlap between the consecutive fuzzy sets and P_i is a polynomial of order p, the input-output characteristics is a polynomial of order $p + 1$.

For any x in the interval $[m_i, m_{i+1}]$ there are two rules activated; m_i and m_{i+1} are the modal values of the corresponding fuzzy sets. This yields the output in the following way

$y = m_i P_i(x, a_i) + m_{i+1} P_{i+1}(x, a_i)$. Note that m_i and m_{i+1} are linear functions of x, say $m_i = b_{0i} + b_{1i}\, x$ and $m_{i+1} = b_{0,i+1} + b_{1,i+1} x$ Therefore $y = (b_{0i} + b_{1i}\, x) P_i(x, a_i) + (b_{0,i+1} + b_{1,i+1} x) P_{i+1}(x, a_i)$. It is apparent that y is a polynomial of order $p + 1$.

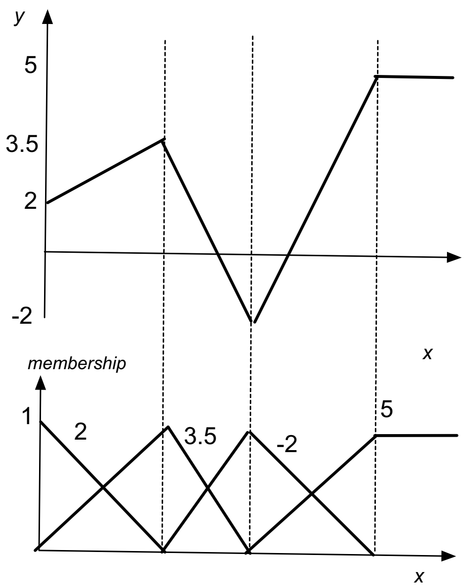


Fig. 13.7 Input-output piecewise-linear characteristics of the fuzzy rule-based model

13.4 The Design of the Rule-Based Model

The generic design process of this class of rule-based models is composed of two phases. We assume a collection of training data in the form of input-output pairs

– development of condition parts of the rules. Fuzzy sets forming the conditions are constructed with the use of the FCM clustering. The number of clusters is equal to the number of rules. Typically, the clustering algorithm is completed in the space of input data. As the fuzzy sets form the conditions of the rules, n-dimensional input data are clustered yielding c clusters with prototypes $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, …, $\boldsymbol{v}_c$. The membership functions of the corresponding A_1, A_2, …, A_c are described in the well-known form

$$A_i(\boldsymbol{x}) = \frac{1}{\sum_{j=1}^{c}\left(\frac{\|\boldsymbol{x}-\boldsymbol{v}_i\|^2}{\|\boldsymbol{x}-\boldsymbol{v}_j\|^2}\right)^{1/(m-1)}} \tag{13.28}$$

The fuzzification coefficient m determines the shape of the membership functions and subsequently impacts the determination of the output.

– development of conclusion parts of the rules. The local functions are selected and their parameters are estimated.

In what follows, we concentrate on the development of the conclusion parts (local models). We consider linear functions forming the conclusions of the rules and coming in the following form

$$f_i(\boldsymbol{x}, \boldsymbol{a_i}) = a_{i0} + a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \tag{13.29}$$

The parameters of the local functions are estimated by minimizing a certain performance index. Prior to the derivation of detailed formulas, we organize some notation. The parameters of (13.29) are arranged in an $(n + 1)$-dimensional vector $\mathrm{a_i} = [\mathrm{a_{i0}}, \mathrm{a_{i1}}, \ldots, \mathrm{a_{in}}]$. The output of the model is expressed in the following form:

$$\hat{y} = \sum_{i=1}^{c} A_i(\boldsymbol{x})a_i^T\begin{bmatrix}1\\ \boldsymbol{x}\end{bmatrix} = \sum_{i=1}^{c} a_i^T\begin{bmatrix}A_i(\boldsymbol{x})\\ A_i(\boldsymbol{x})\boldsymbol{x}\end{bmatrix} \tag{13.30}$$

We use the following notation

$$z_i(\boldsymbol{x}) = \begin{bmatrix} A_i(\boldsymbol{x}) \\ A_i(\boldsymbol{x})x_1 \\ A_i(\boldsymbol{x})x_2 \\ \dots \\ A_i(\boldsymbol{x})x_n \end{bmatrix} \tag{13.31}$$

This leads to the expression

$$\hat{y} = \sum_{i=1}^{c} \boldsymbol{a}_i^T z_i(\boldsymbol{x}) = \sum_{i=1}^{c} z_i^T(\boldsymbol{x})\boldsymbol{a}_i \tag{13.32}$$

Given c rules, the parameters of the local models are organized in a single $c(n+1)$-dimensional vector

$$\boldsymbol{a} = \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \boldsymbol{a}_3 \\ \dots \\ \boldsymbol{a}_c \end{bmatrix} \tag{13.33}$$

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} z_1(\boldsymbol{x}) \\ z_2(\boldsymbol{x}) \\ \dots \\ z_c(\boldsymbol{x}) \end{bmatrix} \tag{12.34}$$

The collection of N-input-output data used in the estimation problem is structured in the following way

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ y_N \end{bmatrix}$$
$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{f}^T(\boldsymbol{x}_1) \\ \boldsymbol{f}^T(\boldsymbol{x}_2) \\ \dots \\ \boldsymbol{f}^T(\boldsymbol{x}_N) \end{bmatrix} = \begin{bmatrix} z_1^T(\boldsymbol{x}_1) & z_2^T(\boldsymbol{x}_1) & \dots & z_c^T(\boldsymbol{x}_1) \\ z_1^T(\boldsymbol{x}_2) & z_2^T(\boldsymbol{x}_2) & \dots & z_c^T(\boldsymbol{x}_2) \\ \dots & \dots & \dots & \dots \\ z_1^T(\boldsymbol{x}_N) & z_2^T(\boldsymbol{x}_N) & \dots & z_c^T(\boldsymbol{x}_N) \end{bmatrix} \tag{13.35}$$

This gives rise to the set of equations in the form

$$\hat{\boldsymbol{y}}=\boldsymbol{F}\boldsymbol{a} \tag{13.36}$$

The performance index Q is formulated as a sum of squared errors

$$Q = (\boldsymbol{y} - \hat{\boldsymbol{y}})^T(\boldsymbol{y} - \hat{\boldsymbol{y}}) = (\boldsymbol{F}\boldsymbol{a} - \boldsymbol{y})^T(\boldsymbol{F}\boldsymbol{a} - \boldsymbol{y}) \tag{13.37}$$

Its minimization with respect to $\boldsymbol{a}$ leads to the following optimal vector of parameters $\boldsymbol{a}_{\text{opt}}$

$$\boldsymbol{a}_{\text{opt}} = \left(\boldsymbol{F}^T\boldsymbol{F}\right)^{-1}\boldsymbol{F}^T\boldsymbol{y} \tag{13.38}$$

13.5 Some Design Alternatives of the Rule-Based Models

The design scheme presented above can be sought as a generic development framework. In what follows, we discuss some alternatives both in terms of some refinements of the design and some variations in terms of the underlying architecture.

13.5.1 Clustering in Combined Input-Output Space

In the generic version, clustering is carried out as a result producing prototypes in the input space and the corresponding fuzzy sets A_i. Alternatively, clusters can be formed in the $(n + 1)$-dimensional space $\boldsymbol{R}^{n+1}$. As the results, the obtained prototypes are constructed in the input and output space at the same time; we obtain $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c$ and the corresponding $w_1, w_2, \ldots, w_c$. Both of then can be used in the construction of the rules. The rules are expressed in the following form

$$-\text{if } \boldsymbol{x} \text{ is } A_i \text{ then } y = w_i + \boldsymbol{a}_i^T(\boldsymbol{x} - \boldsymbol{v}_i) \tag{13.39}$$

The local functions are linear which pass through the points $(\boldsymbol{v}_i, w_i)$ determined by the FCM method. In this situation, we form local linear functions in the following form

$$f_i(\boldsymbol{x}, \boldsymbol{a}_i) = w_i + \boldsymbol{a}_i^T(\boldsymbol{x} - \boldsymbol{v}_i) \tag{13.40}$$

where $\boldsymbol{v}_i$ is a cluster (prototype) located in the input space $\boldsymbol{R}^n$ and w_i is the associated prototypes in the output space. Note that in comparison to the generic version of the local model (13.29), the vector of parameters to be estimated $\boldsymbol{a}_i$ is n-dimensional instead of $(n + 1)$-dimensional one. The output of the rule-based model is a weighted sum of the local functions

$$\hat{y} = \sum_{i=1}^{c} A_i(\boldsymbol{x})\big(w_i + \boldsymbol{a}_i^T(\boldsymbol{x} - \boldsymbol{v}_i)\big) \tag{13.41}$$

As before we use a concise notation

$$\boldsymbol{z}_i = A_i(\boldsymbol{x})(\boldsymbol{x} - \boldsymbol{v}_i) \tag{13.42}$$

$$q = \sum_{i=1}^{c} A_i(\boldsymbol{x})w_i \tag{13.43}$$

Next we have

$$\hat{y} = q + \sum_{i=1}^{c} \boldsymbol{a}_i^T \boldsymbol{z}_i \tag{13.44}$$

We introduce the vector notation

$$\boldsymbol{p} = [y_1 - q_1, y_2 - q_2, \ldots, y_N - q_N]^T \tag{13.45}$$

The parameters of all local models are organized in a vector form $\boldsymbol{a}$ while the data are structured as

$$Z = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1c} \\ z_{21} & z_{22} & & z_{2c} \\ \vdots & & \ddots & \vdots \\ z_{N1} & z_{N2} & \cdots & z_{Nc} \end{bmatrix} \tag{13.46}$$

The performance index is defined as before

$$Q = \sum_{k=1}^{N} \left(y_k - q_k - \sum_{i=1}^{c} \boldsymbol{a}_i^T \boldsymbol{z}_{ki} \right)^2 = (\boldsymbol{p} - Z\boldsymbol{a})^T(\boldsymbol{p} - Z\boldsymbol{a}) \tag{13.47}$$

and the optimal solution becomes

$$\boldsymbol{a}_{\text{opt}} = \left(Z^T Z\right)^{-1} Z^T \boldsymbol{p} \tag{13.48}$$

13.5.2 Modifications to the Local Functions

The local models are linear. They can be made simpler or more advanced (nonlinear):

(i) constant conclusion part. The linear function forming the conclusion of the rule reads in the form $y = a_{0i}$. These parameters can be optimized by solving the standard LSE problem. There are two other options:

FCM is run in the input space and a_{0i} is determined as the weighted sum of the corresponding output values $a_{0i} = \sum_{k=1}^{N} y_k A_i(x_k) / \sum_{k=1}^{N} A_i(x_k)$

When FCM is run in the combined input-output space $\boldsymbol{R}^{n+1}$, the alternative approach is to select a_{0i} as w_i.

(ii) polynomial conclusion parts. Instead of linear local functions, the conclusions are described by polynomials. In spite of this, the estimation problem is linear as before however the dimensionality of the problem increases. For instance, if $n = 2$, the second-order polynomial comes in the form

$$y = a_{0i} + a_1x_1 + a_2x_2 + a_3x_1^2 + a_4x_2^2 + a_5x_1x_2 \tag{13.49}$$

Introduce the notation $z = [1\ x_1\ x_2\ x_1^2\ x_2^2\ x_1x_2]$. It is apparent that the dimensionality is now increased to 6 in comparison to the original dimensionality of 3. In general, more complex (nonlinear) conclusions require less rules to model the problem however, the complexity associated with the increasing difficulty to interpret the rules becomes higher.

13.6 Fuzzy Controllers

Control engineering has a long tradition and exhibits a plethora of applications. In essence, a general control system can be portrayed as illustrated in Fig. 13.8.

The objectives of control could be different. For instance, one is interested in reaching a certain reference *ref* and staying there; this is a typical stabilization task, Fig. 13.9a. One can look at the tracking problem, viz. an efficient way of following

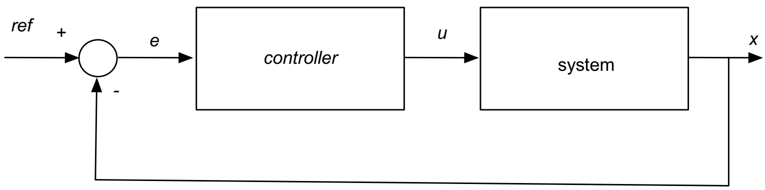


Fig. 13.8 Overall architecture of feedback control architecture: a general concept

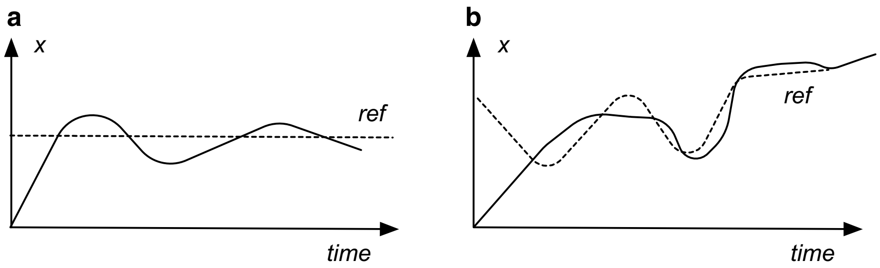


Fig. 13.9 Examples of control problems: **a** stabilization, and **b** tracking

a reference point, which varies over time, Fig. 13.9b.

To solve control problem outlined above, one has to design a control algorithm (controllers). The omnipresent types of controllers are so-called PI and PID controllers. They are described in the following form

PI controller (Proportional–Integral)

$$u(t) = k_P e(t) + k_I \int_0^T e(\tau)d\tau \tag{13.50}$$

PID controller (Proportional-Integral-Derivative)

$$u(t) = k_P e(t) + k_D \frac{de}{dt} + k_I \int_0^T e(\tau)d\tau \tag{13.51}$$

The control formula states that control u is a function of error (e) and derivative (change) of error (de/dt). The error input e is defined as the difference *ref*-x. In discrete time version, one has $e(k) = ref(k) - x(k)$ and $\Delta e(k) = e(k) - e(k-1) = ref(k) - x(k) - [ref(k-1) - x(k-1)] = ref(k) - ref(k-1) + x(k-1) - x(k)$. If *ref* is constant over time, one has $\Delta e(k) = x(k-1) - x(k)$. The design of the controller is concerned with the optimization of their parameters (k_P, k_D, k_I).

There are two imperative components that are necessary to develop any control strategy, namely

(i) the model of the system under control. It has to be stressed that precisely we are concerned about *model* control not *system* control. Of course, we anticipate that the model is accurate or relevant enough so that the control solution solved in this way is pertinent to the real-world system.

(ii) performance index Q that is optimized (minimized) through the design of the controller. The performance index has to reflect the objective of control. For instance, several obvious examples are shown below

$$Q = \int_0^T e^2(\tau)d\tau \tag{13.52}$$

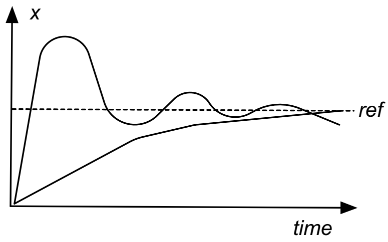


Fig. 13.10 Example of control responses

$$Q = \int_0^T |\tau e(\tau)| d\tau \tag{13.53}$$

$$Q = \max_t \int_{[0,T]} |e(t)| \tag{13.54}$$

The above indices quantify the performance of the controller. The index (13.52) is typical stating how far the response of the system is far from the reference target. The performance index (13.53) stresses the importance of the fast pace of achieving the reference value (*ref*) and penalizes situations when the error persists in later time moments. Likewise (54) stresses the temporal aspect of the control quality. Figure 13.10 illustrates several responses of the system under control and stresses the conflicting character of the obtained solution: one has a quick response but with significant overshoot or the overshoot can be minimized but at the expense of the slow response.

Fuzzy controller has emerged as a sound alternative to the existing standard control algorithms.

The rules link the current status of the system expressed in terms of error and change of error and the resulting control. In terms of the control strategy, it is expressed in terms of rules assuming the following format [3, 4].

$$-\text{if error is } A_i \text{ and change of error is } B_j \text{ then control is } C_k \tag{13.55}$$

Alternatively, the rules come in the following way

$$-\text{if error is } A_i \text{ and change of error is } B_j \text{ then change of control is } D_k \tag{13.56}$$

where A_i, B_j, C_k, D_k are fuzzy sets defined in the corresponding spaces or error, change of error and control and change of control. From the point of view, the rules (13.55) can be regarded as PD controller and (13.53) are PI controllers.

The set of rules are then processed in the way discussed before. For any input x described in the form (error, change of error) we determine the output (control or change of control) coming in the form of some fuzzy set.

The set of rules captures the essence of the control strategy and identifies main relationships between the error, change of error and the resulting change of control.

The underlying control metaheuristics is expressed in the form of the intuitively appealing rules:

- -if error and change of error are close to zero then maintain the current control setting
- -if error tends to zero, then maintain the current control setting

Such metarules rules apply to situations where there is some self-correcting situation and this means that the current control (change of control) is not changed. In all other situations, one has to make changes to control depending upon the sign and magnitude of error and change of error. The phase space of the control system is illustrated in Fig. 13.11; here we show the change of control associated with each location on the spiral.

As the control systems are predominantly numeric, a numeric control action is developed through a certain decoding mechanism, say a center of gravity.

The architecture of the fuzzy controller is visualized in Fig. 13.12; again, the three modules: encoder-processing module-decoder are clearly present.

As any approach, fuzzy controllers exhibit some advantages and limitations

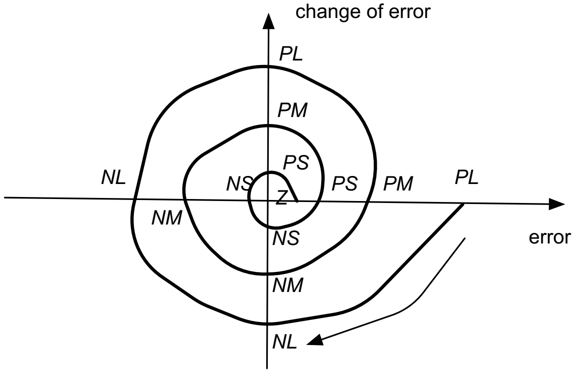


Fig. 13.11 Phase plane (error, change of error) along with the linguistic quantification of the change of control

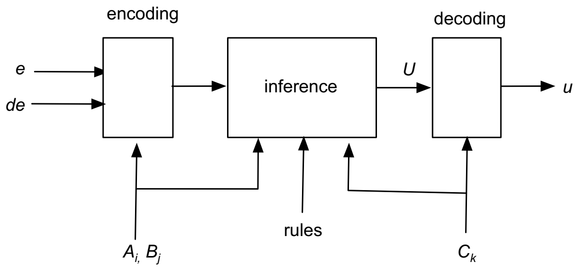


Fig. 13.12 An architecture of the fuzzy controller

13.6.1 Advantages

(i) Fuzzy controller is designed by accommodating the domain knowledge about the system and the objective (goal) of control which is articulated at some level of abstraction implied by fuzzy sets. In this sense, the control strategy is in rapport with the essence of the real-world problem.
(ii) fuzzy controllers are nonlinear controllers, which could lead to better dynamic performance of the system control than PID controllers
(iii) the same generic control strategy applies to a significant extent to a variety of systems given the underlying metaheuristics expressed by the rules. Given the nature of the rules, optimizing a scaling factors (adjusting the spaces of input spaces) offers a substantial level of flexibility.

Fuzzy controllers exhibit some limitations

(i) there is a lack of the underlying model of the system and an implicit nature of the performance index. This raises a question of stability and optimality of the controller. As there is no model given an explicit manner, the answer cannot be given in a formal manner.
(ii) while the intuitive underpinnings of the rules are convincing, fuzzy controllers come with a suite of parameters whose values require adjustments. The refinement process may require some computing overhead.

13.7 Hierarchy of Fuzzy Controllers

The rules (as being guided by some commonly acceptable metaheuristics) can be regarded as being invariant. What needs adjustments, is the adjustments made with respect to the space of error and change of error in which the controller has to function. The essence of the approach is illustrated in Fig. 13.13.

The transformation is expressed through the use of some scaling factor. Considering the nominal range $[e_{\min}, e_{\max}]$ for which fuzzy sets have been defined for the rules, the controller at the higher level of specificity operates within the narrower range $[e^-, e^+]$. The generic controller is used here by bringing a calibration of the space by introducing a scaling factor κ where

$$K = \left| \frac{e_{\max} - e_{\min}}{e^+ - e^-} \right| \tag{13.57}$$

see also Fig. 13.14.

The hierarchy of the controller can involve several levels by moving to more locally operating fuzzy controller with the successive scaling coefficients associated with the corresponding level of hierarchy. In this way, when encountering some input e in $[e^-, e^+]$, one carries out the scaling $e' = \kappa e$ and for this value computes the firing levels of the generic set of the rules.

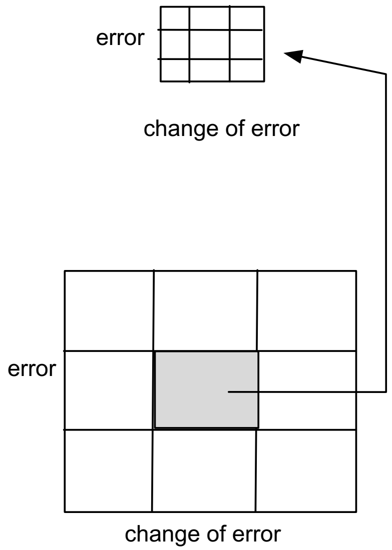


Fig. 13.13 A hierarchy of fuzzy controllers; focus on a subspace of error and change of error

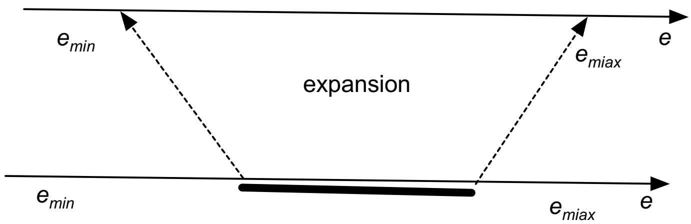


Fig. 13.14 Expansion process of spaces

Example 13.3 Consider the nominal ranges for error and change of error to be [− 10, 10] and [− 5, 5], respectively. The controller at the higher level of hierarchy operates in the narrower ranges of error and change of error specified as [− 2, 2] and [− 1, 1], respectively. Determine the scaling factors.

In virtue of (13.57) we have $\kappa = 20/10 = 2$ for the error variable and $\kappa = 4/2 = 2$ for the change of error.

13.8 Functional Rule-Based Model of Variable State of the PID Controller

Commonly PID controllers come with a suite of optimized parameters which after optimization, are kept unchanged. The rule-based model can lead to a variable parameter structure. The rules are formed as follows

$$-\text{if error is } A_i \text{ and change of error is } B_j \text{ then } k_P = k_{P0},\ k_D = k_{D0},\ k_I = k_{I0} \tag{13.58}$$

$i = 1, 2, \ldots, c$ where the conclusion parts include some constant values of the parameters (k_{P0}, k_{D0}, and k_{I0}) of the local PID controllers which have been optimal in the local region in the (error, change of error) space.

In this way, the combination of rule-based model switches among several PID controllers thus endowing an overall control architecture with high flexibility.

13.9 Rule-Based Models for Highly-Dimensional Data

The design of rule-based model faces some challenges when coping with data with a large number of input variables. One of the solutions is to focus on a modular design, in particular, develop a suite of one-dimensional rule-based models and then combine their results.

13.9.1 Design and Aggregation of One-Dimensional Rule-Based Models

The design of a collection of low-dimensional models and their further aggregation is motivated by the two arguments. Firstly, for highly dimensional data, the concentration effect becomes prominently present meaning that the concept of distance (which is used in a spectrum of clustering algorithms) becomes less relevant. This implies that the design of fuzzy rule-based models for which the use of Fuzzy C-Means (FCM) is common has to be revisited. As already discussed, the standard way of designing fuzzy rule-based model is composed of two phases, namely (i) the design of fuzzy sets forming the condition part of the rules, and (ii) parametric optimization of the local (usually linear) functions forming the conclusion part of the rules. There is also a growing computing overhead when clustering highly-dimensional data. Secondly, it has been convincingly demonstrated that the development of modular models (with each of them constructed with the aid of only few input variables) is a sound design alternative, [2]. Obviously, low-dimensional models are not ideal and this development strategy has to be augmented by carefully formed aggregation schemes. It is worth stressing that these issues have not received a great deal of attention and a formation of a development approach by taking these two observations into account deserves attention and a careful investigation.

13.9.2 Concentration Effect

In highly dimensional space, any distance starts losing its relevance in the sense the distance between any two points starts to be the same; the phenomenon referred to as a concentration effect [1]. As the distance is a central concept used in the FCM algorithm, this deterioration has a highly detrimental impact on the quality of the generated clusters. To take a closer look at the concentration effect, let us consider 200 data $\boldsymbol{x}_k$ located randomly on a unit n-dimensional hypersphere. We compute distances $||\boldsymbol{x}_k - \boldsymbol{x}_l||^2$ and display a histogram of their values reported for each pair of the data. The results are displayed in Fig. 13.15. The tendency is highly visible: with the increase in the values of n, the histogram becomes more centralized with a rapidly decreasing variance. The average of the distances tends to zero; for $n = 10$, we have 0.066 while for $n = 200$, the average is 0.003 and drops down to 0.001 for $n = 500$.

The one-dimensional model does not require any design effort and associated computing overheard. The simplicity of the development, however, is a genuine asset of this approach.

When dealing only with a single input variable when forming input-output dependency, we encounter an emerging relational phenomenon of data. This means that for two very closely located inputs, the outputs could vary significantly. In particular, because all but a single variable are eliminated, one might encounter an extreme situation when for the same input we have two or more different outputs.

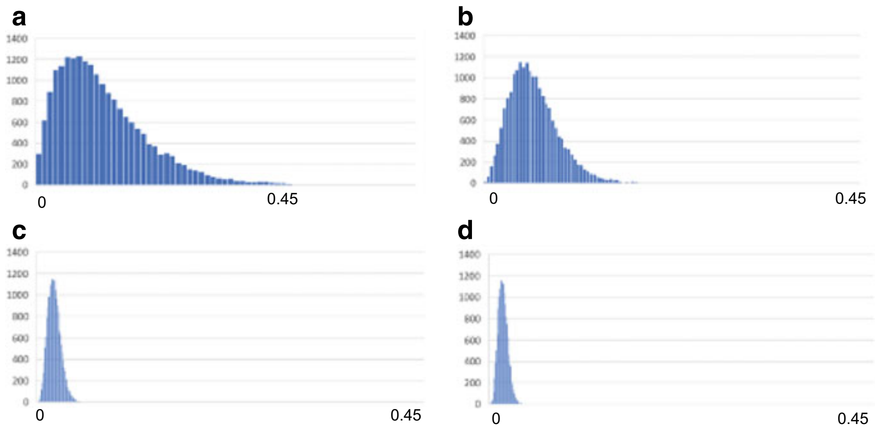


Fig. 13.15 Histograms of values of distances for selected values of n: **a** $n = 5$, **b** $n = 10$, **c** $n = 25$, and **d** $n = 30$

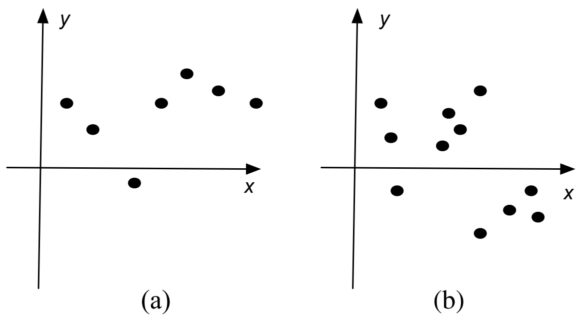


Fig. 13.16 Input-output data and their relational nature

Consider input-output data (x_k, y_k) normalized to [0,1] interval. We are interested in the determination of the degree to which extent the data are of relational nature, viz.

rel = degree (data exhibit relational character)

For instance, we envision that the data in Fig. 13.16b exhibit more relational character than the one shown on Fig. 13.16a.

Having in mind the property of high closeness of two inputs x_k and x_l associated with different values (low closeness) of the corresponding outputs y_k and y_l, we propose the degree of relational nature computed in the following way

$$rel_{kl} = \begin{cases} 0, \text{if}|y_k - y_l| \le |x_k - x_l| \\ 1 - \frac{|x_k - x_l|}{|y_k - y_l|} + \delta > 0, \text{if}|y_k - y_l| > |x_k - x_l| \end{cases} \tag{13.59}$$

Note that when $|x_k\text{-}x_l|$ becomes lower for the same value of $|y_k\text{-}y_l|$ higher than $|x_k\text{-}x_l|$, this increases the value of the degree. The global index is determined by taking a sum of rel_{kl} for the corresponding pairs of the data, namely

$$rel = \sum_{k>l} rel_{kl} \tag{13.60}$$

The higher the value of *rel*, the more evident the relational nature of the one-dimensional data ***D***.

Distributed rule-based model

We proceed with the series of one-dimensional rule-based models (It is worth noting that the use of a single input variable is also motivated by the experimental observations reported in [17] and [15] [16]. Next such models are organized in the structure visualized in Fig. 13.17.

The architecture is composed of two main functional modules as outlined in Fig. 13.17.

Each variable $x_1, x_2, \ldots, x_n$ is transformed with the use of c membership functions (say, triangular membership functions) resulting in total of cn membership grades organized in the form of some vector $\boldsymbol{u}$. Next the membership grades are transformed

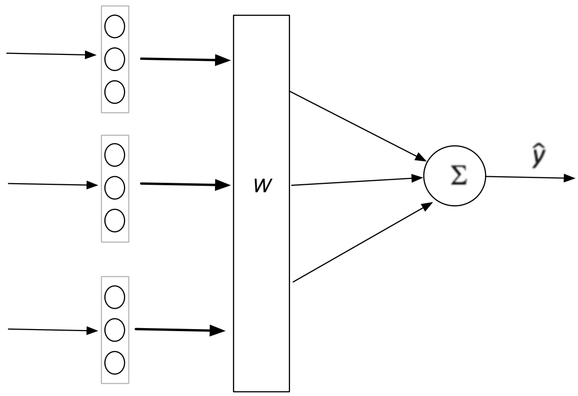


Fig. 13.17 Distributed rule-based models

linearly by some weight (linkage) matrix W into a vector $\boldsymbol{z}$ of dimensionality p

$$\boldsymbol{z} = W\mathbf{u} \tag{13.61}$$

viz. for each vector $\boldsymbol{x}(k) = [x_1(k)\ x_2(k)\ \ldots\ x_n(k)]$ one has

$$z_i(k) = \sum_{j=1}^{cn} w_{ij} u_{jk} \tag{13.62}$$

Next the entries of $\boldsymbol{z}(k)$ are used as inputs of the weighted sum

$$\hat{y} = \sum_{\mathrm{i}=1}^{\mathrm{p}} z_i \bar{y}_i \tag{13.63}$$

where $\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_{\mathrm{p}}$ are the representatives (prototypes) positioned in the output space.

In the design process, we consider pairs of input–output data $\boldsymbol{x}(k)$, $y(k)$, $k = 1, 2, \ldots, N$. In the above scheme, the design process involves the minimization of the performance index Q

$$Q(W) = \sum_{k=1}^{N} (y(k) - \hat{y}(k))^2 \tag{13.64}$$

Let us proceed with the detailed estimation process. The original output data, output of model and the numeric prototypes are arranged in the vector format

$$\boldsymbol{y} = \begin{bmatrix} \mathrm{y}_1 \\ \mathrm{y}_2 \\ \vdots \\ \mathrm{y}_\mathrm{N} \end{bmatrix}, \hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_\mathrm{N} \end{bmatrix}, \bar{\boldsymbol{y}} = \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \bar{y}_p \end{bmatrix} \tag{13.65}$$

Regarded as a function of W and y two optimization tasks are considered.

(i) We have the optimization where $W_{\text{opt}} = \arg \operatorname{Min}_W Q(W)$

Furthermore, in virtue of the matrix notation $\hat{\boldsymbol{y}} = (\mathrm{WU})^\mathrm{T} \bar{\boldsymbol{y}}$. Next we rewrite the optimization problem (13.10) in the following concise format

$$\min_W \left\| \boldsymbol{y} - \hat{\boldsymbol{y}} \right\|^2 = \min_W \left\| \boldsymbol{y} - (WU)^T \bar{\boldsymbol{y}} \right\|^2 \tag{13.66}$$

Proceeding with the detailed calculations, we obtain

$$\left\| \boldsymbol{y} - (WU)^T \bar{\boldsymbol{y}} \right\|^2 = \boldsymbol{y}^T \boldsymbol{y} + (\bar{\boldsymbol{y}}^T WUU^T W^T \bar{\boldsymbol{y}}) - (\bar{\boldsymbol{y}}^T WU)\boldsymbol{y} - \boldsymbol{y}^T (U^T W^T \bar{\boldsymbol{y}}) \tag{13.67}$$

Taking the gradient of (13.67), we have the following expression

$$\nabla_W \left\| \boldsymbol{y} - (WU)^T \bar{\boldsymbol{y}} \right\|^2 = 2\bar{\boldsymbol{y}}\bar{\boldsymbol{y}}^T WUU^T - 2\bar{\boldsymbol{y}}\boldsymbol{y}^T U^T = 0 \tag{13.68}$$

Finally, the analytical solution to (13.15) comes in the form

$$\begin{aligned} 2\bar{\boldsymbol{y}}\bar{\boldsymbol{y}}^T W_{opt} UU^T &= 2\bar{\boldsymbol{y}}\boldsymbol{y}^T U^T \\ W_{opt} &= (\bar{\boldsymbol{y}}\bar{\boldsymbol{y}}^T)^{-1} \bar{\boldsymbol{y}}\boldsymbol{y}^T U^T (UU^T)^{-1} \end{aligned} \tag{13.69}$$

(ii) the minimum of Q with respect to the vector $\bar{y}$ is expressed as

$$\min_{\bar{y}} \| \boldsymbol{y} - \bar{\boldsymbol{y}} \|^2 = \min_{\bar{y}} \left\| \boldsymbol{y} - (WU)^T \bar{\boldsymbol{y}} \right\|^2 \tag{13.70}$$

Let us introduce $A = WU$. Then we have

$$\left\| \boldsymbol{y} - (WU)^T \bar{\boldsymbol{y}} \right\|^2 = \left\| \boldsymbol{y} - A^T \bar{\boldsymbol{y}} \right\|^2 = \boldsymbol{y}^T \boldsymbol{y} + (\bar{\boldsymbol{y}}^T AA^T \bar{\boldsymbol{y}}) - (\bar{\boldsymbol{y}}^T A)y - y^T (A^T \bar{\boldsymbol{y}}) \tag{13.71}$$

Taking the gradient of the expression we obtain

$$\nabla_{\bar{y}} \left\| \boldsymbol{y} - A^T \bar{\boldsymbol{y}} \right\|^2 = 2AA^T \bar{\boldsymbol{y}} - 2A\boldsymbol{y} = 0 \tag{13.72}$$

Proceeding with the detailed computing, one has

$$
\begin{aligned}
AA^T\bar{\boldsymbol{y}}_{opt} &= A\boldsymbol{y} \\
\bar{\boldsymbol{y}}_{opt} &= (AA^T)^{-1}A\boldsymbol{y} \\
\bar{\boldsymbol{y}}_{opt} &= (WU(WU)^T)^{-1}(WU)\boldsymbol{y}
\end{aligned}
\tag{13.73}
$$

As W and $\bar{\boldsymbol{y}}$ are subject to the optimization of $Q(W,\bar{\boldsymbol{y}})$, the expressions (13.69) and (13.73) are used iteratively proceeding with some initial condition, say considering a uniformly distributed output prototypes, and the successively updating the values of W and $\bar{\boldsymbol{y}}$.

13.10 Conclusions

We highlighted the diversity of fuzzy rule-based models and identified two main classes of relational and functional rules. The well-established, comprehensive, and systematic design process is an asset of these fuzzy models. The performance of rule-based models is two-faceted: one can focus on the approximation abilities of the rules or the focus is on the quality of the rules in terms of their descriptive capabilities as pieces of knowledge built on a basis of available data. We discussed several architectural enhancements whose need is dictated by data of high volume and dimensionality.

Problems

1. Show under what conditions the relational fuzzy rule-based model is equivalent (in terms of the ensuing characteristics) with the functional fuzzy rule-based model.
2. Consider a collection of rules in the following form
   -if x is A_i then $y = a_{0i} + a_{1i}\sin(ix)$
   $i = 1,2, \ldots, 5$ where A_i is a Gaussian membership function with modal values distributed uniformly across the range [− 10, 10] and spreads equal to 1.0. Plot the input–output characteristics. Discuss how to estimate the parameters of the local functions.
3. The nominal ranges for error and change of error are [− 15, 10] and [-2, 4], respectively. The controller at the higher level of hierarchy operates in the narrower ranges of error and change of error specified as [-2, 2] and [-1, 1], respectively. Determine the values of the scaling factors.
4. What are the advantages and disadvantages of functional rules with local functions formed by polynomials.
5. For each input variable, determine the degree of relational nature for the following data

x_1	x_2	y
1.0	0.7	0.5
2.1	0.6	−2.6
3.1	0.5	4.1
4.5	−0.2	2.7
1.1	−0.3	1.7
−1.6	1.5	4.5

6. How to determine parameters in the conclusions of the rules assuming the following form
   (i) if x is A_i then $y = a_i\sin(3ix)$
   and
   (ii) if x is A_i then $y = b_i\text{scos}(3ix + \phi_i)$
   Which estimation problem is more difficult?
7. **7**. Determine the number of rules coming in the form
   -if x_1 is A_i and x_2 is B_j and … x_n is W_l then conclusion is Z_k
   with fuzzy sets in the condition part having a finite support for the following pairs of values of (c, n): (5, 10), (7, 20), (9, 30).
   How could you avoid the curse of dimensionality.

References

1. Knox, S.W.: Machine Learning: A Concise Introduction, pp. 149–152. Wiley, Hoboken, NJ, USA (2018)
2. Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. Mach. Learn. **11**(1), 63–91 (1993)
3. Ishaque, K., Abdullah, S.S., Ayob, S.M., Salam, Z.: Single input fuzzy logic controller for unmanned underwater vehicle. J. Intell. Rob. Syst. **59**(1), 87–100 (2010)
4. Mamdani, E.H.: Application of fuzzy algorithms for control of simple dynamic plant. Prpc. IEE **121**, 1585 (1974)
5. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybern **15**, 116–32 (1985)
6. Zadeh, L.A.: The concept of linguistic variables and its application to approximate reasoning I, II, III. Inf. Sci. **8**, 199–249, 301–357, 43–80 (1975)

Chapter 14
Fuzzy Neurocomputing

Abstract We are concerned with a fundamental idea of fuzzy neurocomputing manifesting as a realization of the synergy of fuzzy sets and neural networks. These two frameworks are complementary to a significant extent. The architectures of neurofuzzy systems benefit from mechanisms of *explicit* knowledge representation supported by fuzzy sets and a spectrum of *learning* methods being a genuine forte of neurocomputing. Several categories of fuzzy (logic) neurons are introduced, the topologies of the networks built with the aid of these neurons are discussed and learning schemes are presented. The linkages with Boolean networks are highlighted and mechanisms enhancing the interpretability of the network. The gradient-based learning is discussed as a generic mechanism of supervised learning. Selected architectures of neurofuzzy systems involving autoencoders and relational factorization are put forward.

14.1 Neural Networks: Concise Prerequisites

Neural networks are distributed computing architectures composed of neurons—simple nonlinear processing units [1]. A neuron realizes an n-input single-output mapping $\boldsymbol{x} \in \boldsymbol{R}^{n} \to y \in [0, 1]$ governed by the following expression

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_14

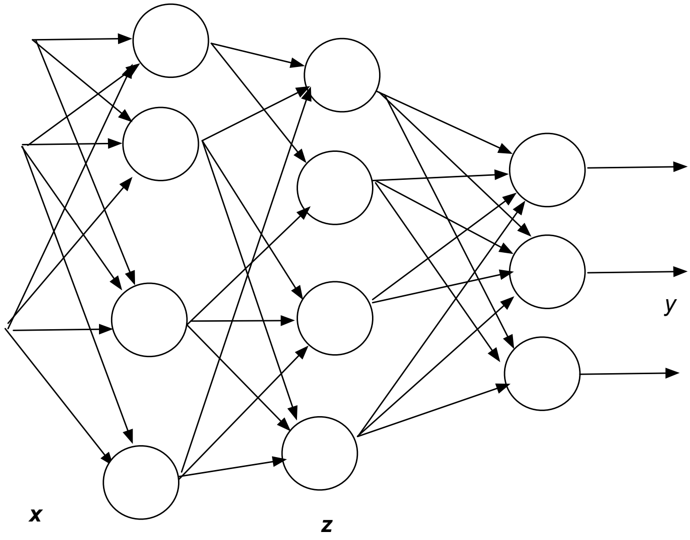


Fig. 14.1 Example neural network

$$y = f(1, x_1, x_2, \ldots, x_n, w_0, w_1, w_2, \ldots, w_n) \tag{14.1}$$

where f is a nonlinear function and $\boldsymbol{w} = [\, w_0, w_1, w_2, \ldots, w_n]$ is a vector containing weights and bias (w_0). A typical function is a sigmoidal one, namely $f(u) = 1/(1 + \exp(-u))$. The neuron completes a linear transformation $\boldsymbol{w}^T\boldsymbol{x}'$ where $\boldsymbol{x}' = [1\ \boldsymbol{x}]^T$ followed by a nonlinear function f, $f(\boldsymbol{w}^T\boldsymbol{x}')$. Geometrically, the linear part of the neuron is a hyperplane and w_0 serves as a bias term. Neural networks usually consist of layers of neurons, Fig. 14.1.

Processing is carried out in a forward manner; one starts with $\boldsymbol{x}$, computes z, and proceeds with the computing the output y. The fundamental result is the universal approximation theorem stating that any continuous function $\boldsymbol{R}^n \rightarrow \boldsymbol{R}$ can be approximated to any required accuracy by a neural network with a single hidden layer.

The essence of neural networks is to learn unknown nonlinear mappings. The adjustable weights make the learning possible. There are a number of efficient learning. The backpropagation algorithm is one of the commonly used learning method [1]. The term backpropagation stems from a way of processing initiated from the output layer and moving to the input through successive layers and adjusting weights.

In virtue of the distributed architecture (a collection of neurons), neural networks are black boxes so the input-output mapping is not interpretable. One cannot come up with a relationship which could be easily comprehended (such as a quadratic function, or lower order polynomials, rules).

14.2 Fuzzy Sets and Neural Networks: A Synergy

Fuzzy sets and neural networks are complementarity in terms of their advantages and limitations. The brief discussion above highlights learning capabilities as a genuine advantage of neural networks. At the same time, neurocomputing associates with opaque architectures—the black box nature is an evident limitation. Fuzzy set technology delivers transparent mechanisms of knowledge representation: fuzzy sets describe concepts that are semantically sound. Logic operations on fuzzy sets are easily interpretable. At the same time fuzzy set constructs are not endowed with learning capabilities. The synergy is inevitable: interpretable aspects of knowledge representation are supported by fuzzy sets and neurocomputing faculties endow the hybrid neurofuzzy systems with learning capabilities. The essence of neurofuzzy systems [2] is to form architectures and learning by bringing these two technologies. There is a spectrum of possible topologies and ways of design to be explored in such systems.

14.3 Main Classes of Fuzzy Neurons

Fuzzy (logic) neurons are constructs whose structure is the same as in neurons and processing is realized with the aid of t norms and t conforms [3]. Two categories of logic neurons are distinguished, namely aggregative and referential neurons.

14.3.1 Aggregative Neurons

Two types of logic neurons are considered.

OR neuron or neurons OR($\boldsymbol{x}$; $\boldsymbol{w}$) complete a mapping $[0, 1]^n \rightarrow [0, 1]$

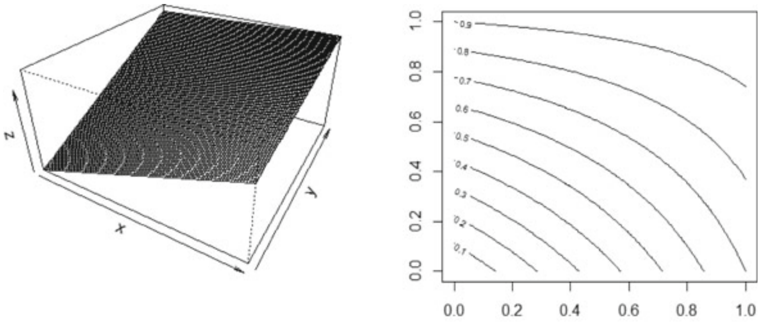
$$y = \mathrm{OR}(x;\, w) = S_{i=1}^{n}(w_i t x_i) \tag{14.2}$$

where $\boldsymbol{w}$ is a vector of adjustable weights which are modified in the process of learning. Let us look closer at the main properties of the neuron:

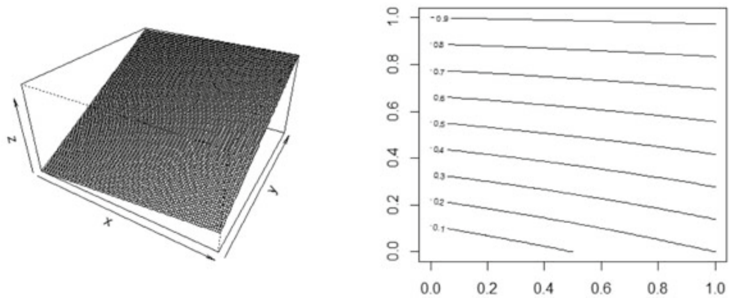
If $\boldsymbol{w} = \mathbf{1}$, $y = \mathrm{OR}(\boldsymbol{x}; \mathbf{1}) = x_1$ or x_2 or … or x_n. The inputs are combined or-wise; hence the name of the neuron.

Higher values of w_i associated with x_i indicate higher impact of this input on the output. The highest impact happens for $w_i = 1$. In limit, w_i= 0 eliminates any contribution of x_i to the output. Different t-norms and t-conorms could be used to implement the neuron. Their choice impacts the characteristics of the neuron, see Fig. 14.2 where t-norm is the product while the t-conorm is the probabilistic sum. The impact of the values of the weights are clearly visible.

The OR neuron realizes a logic expression



(a)



(b)

Fig. 14.2 Input-output characteristic of OR neuron: **a** $w_1 = 0.7$, $w_2 = 0.2$; **b** $w_1 = 0.1$, $w_2 = 0.9$

$$y = (w_1 \text{ and } x_1) \text{ or } (w_2 \text{ and } x_2) \text{ or} \ldots \text{or}(w_n \text{ and } x_n) \tag{14.3}$$

weighting the inputs *and*-wise. If the weights w_i and the inputs are Boolean coming from $\{0, 1\}$ the OR neuron reduces to the OR gate encountered in digital systems.

AND neuron and neurons AND($\boldsymbol{x}$; $\boldsymbol{w}$) complete a mapping $[0, 1]^n \rightarrow [0, 1]$

$$y = \text{AND}(x;\, w) = T_{i=1}^{n}(w_i s x_i) \tag{14.4}$$

where $\boldsymbol{w}$ is a vector of adjustable weights which are modified in the process of learning. Let us look closer at the main properties of the neuron (Fig. 14.3).

If $\boldsymbol{w} = \boldsymbol{0}$, $y = \text{AND}(\boldsymbol{x}; \boldsymbol{0}) = x_1$ and x_2 and … and x_n The inputs are combined and-wise; hence the name of the neuron. Lower values of w_i associated with xi indicate higher impact of this input on the output. The highest impact happens for $w_i = 0$. In limit,the weight w_i= 1 eliminates any contribution of x_i to the output. t-norms and t-conorms are reflected in the characteristics of the neuron, see Fig. 14.4 where the t-norm is the product while the t-conorm is the probabilistic sum.

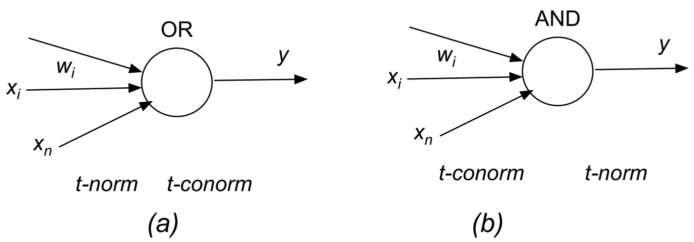


Fig. 14.3 Aggregative neurons (OR and AND)

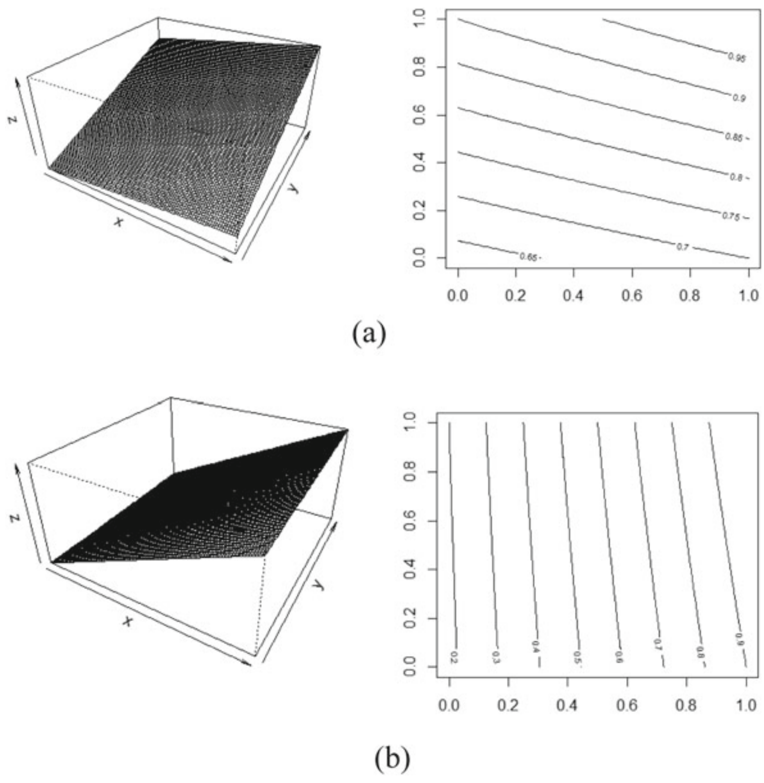


Fig. 14.4 Input-output characteristic of AND neuron: **a** $w_1 = 0.1, w_2 = 0.9$, **b** $w_1 = 0.9, w_2 = 0.1$

AND and OR neurons generalize two-valued (Boolean) digital gates; Fig. 14.5. If the weights are fixed (0 or 1) and the inputs are Boolean, then the neurons reduce to the digital (two-valued) gates.

Example 1 Consider an OR neuron with two inputs and *t*-norm and *t*-conorm specified as the algebraic product and probabilistic sum, respectively. Discuss the input-output characteristics of the neuron.

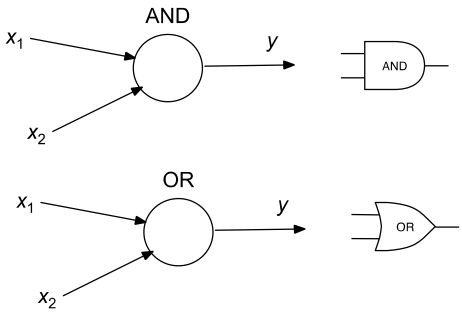


Fig. 14.5 Logic neurons as generalizations of digital gates

The formula describing the neuron is $y = w_1x_1 + w_2x_2$ - $w_1w_2x_1x_2$. It becomes clear that the characteristics are nonlinear because of the product x_1x_2. For 3-input neuron, we obtain products of the pairs of inputs and the product of all inputs. These terms help capture joint dependencies among the input variables.

14.3.2 Referential Neurons

Referential neurons are neurons which are regarded as a serial structure composed of some reference operation followed by the AND neuron, see Fig. 14.6. The reference operations are fuzzy set-based formulas exhibiting some geometric features being a generalization of two-valued predicates of *less than*, great than, and *equal*; refer to Table 14.1. In their representation used are *t*-norms, *t*-conorms and implication operators (ϕ-operator).

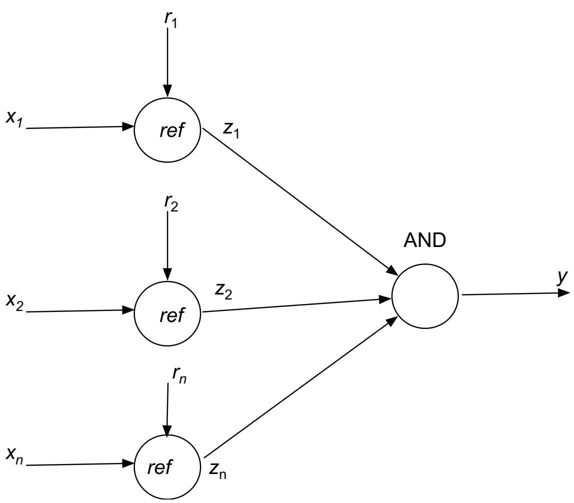


Fig. 14.6 An architecture of the referential neuron REF(***x***; ***w***, ***r***)

Table 14.1 Examples of referential operators ref(x, r)

Operation	Interpretation	Computing
Inclusion *incl*(x, r)	A degree x is included in r	$incl(x, r) = x \phi r$
Dominance *dom*(x, r)	A degree x dominates r (r is included in x)	$dom(x, r) = r \phi x$
Similarity *sim*(x, r)	A degree x is similar to r	$sim(x, r) = incl(x, r)\ t\ dom(r, x)$

Recall that the ϕ-operator is implied by a continuous t-norm, namely $a \phi b = \sup\{c\ \varepsilon\ [0, 1] \mid at\ c \leq b\}$.

The referential neuron realizes a two-phase processing: first the reference operator is applied to the inputs and next the results are combined *and*-wise.

The reference operators can be regarded as constraints—the result of satisfaction of the reference operation is a constraint. Next these results are combined and-wise. From this perspective, the reference neuron can be sought as a two-level fuzzy neural network.

We can express the computing in the form

$$y = \mathrm{AND}(\boldsymbol{z}; \boldsymbol{w}) \tag{14.5}$$

where $\boldsymbol{z} = [\mathrm{ref}(x_1, r_1)\ \mathrm{ref}(x_2, r_2) \ldots \mathrm{ref}(x_n, r_n)]$.

The essential parameters of the referential neuron are: the type of the reference operation, reference vector $\boldsymbol{r} = [r_1\ r_2 \ldots r_n]$ and the weights $\boldsymbol{w} = [w_1\ w_2 \ldots w_n]$

The reference neuron can be viewed as a system that aggregates and-wise weighted constraints implied by the reference operation imposed for each input x_i. The processing is described as $y = \mathrm{REF}(\boldsymbol{x}; \boldsymbol{w}, \boldsymbol{r})$.

Compensatory neuron realizes an aggregation exhibiting *and* and *or* characteristics. The processing is carried out by AND and OR neurons whose results are then aggregated as weighted sum, Fig. 14.7. The weights of the linear unit are γ and 1 − γ, respectively.

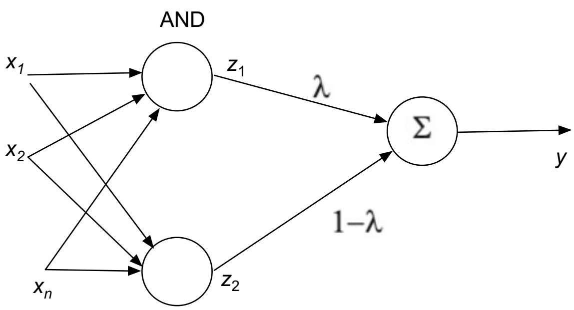


Fig. 14.7 Compensatory logic neuron

14.4 Logic Processor: A Generic Architecture of Logic-Based Mapping

Logic neurons are arranged to build a network. A generic architecture of the network referred to as a logic processor is composed of two layers of neurons. The first one consists of AND neurons followed by the second one formed by OR neurons, Fig. 14.8.

The topology of the network is motivated by the representation of Boolean functions [4–6]. Recall that in light of the Shannon theorem any Boolean function can be represented by a two-layer network of logic gates with the first layer composed of *and* gates and the second one built using *or* gates. AND and OR neurons generalize the gates. There is an obvious analogy and a strong motivating factor behind the formation of the logic processor. There is a visible difference, though. Any Boolean function represents Boolean data. A logic processor approximates pairs of data in $([0, 1]^n, [0, 1]^m)$ in a logic-driven fashion.

The alternative topology of the logic processor is a structure composed of OR neurons followed by AND neurons, Fig. 14.9.

The logic processor arises as a straightforward generalization of digital circuits, see Fig. 14.10.

The analogy is evident in terms of the underlying topologies of these two networks. There are two key differences: the inputs are binary as opposed to continuous, and (ii) the logic processor realizes approximation of input-output data through learning.

Fig. 14.8 A generic topology of a logic processor

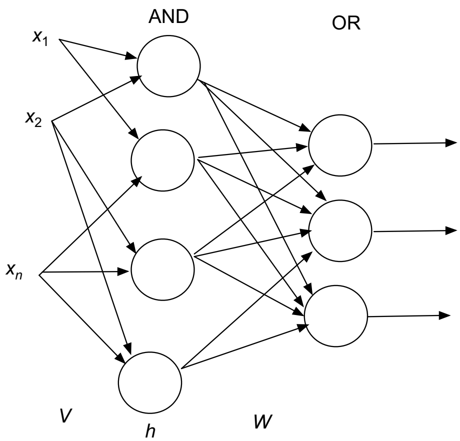
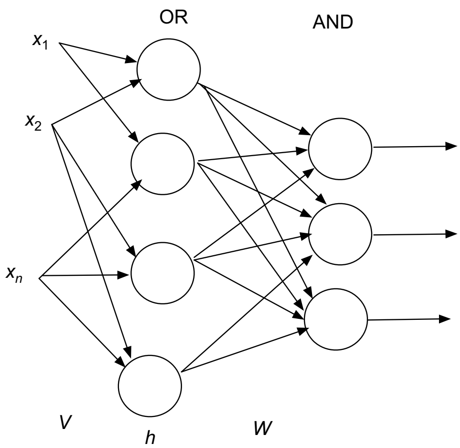


Fig. 14.9 Topology of the logic processor formed by a layer of OR neurons followed by AND neurons



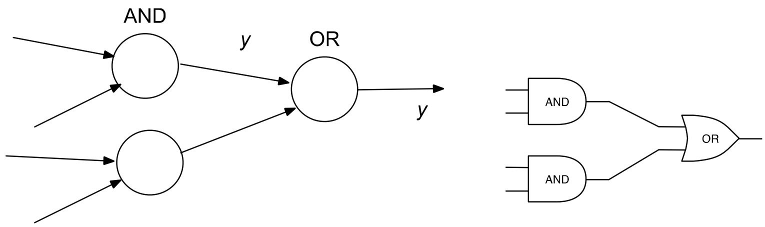


Fig. 14.10 Two -layer digital circuit as a special case of logic processor

14.5 Learning of Fuzzy Neural Networks

The design of any neurofuzzy neural network [7] is composed of two phases: (i) structural development, and (ii) parametric optimization. The structural optimization concerns the following parameters that define the structure (topology) of the network such as

- the number of layers and the number of neurons in each layer, type of neurons
- *t*-norm and *t*-conorm.

The structural optimization is demanding and calls for some advanced techniques of non-parametric optimization; here methods of evolutionary optimization could be a sound alternative.

The parametric optimization is about adjusting the weights of the network so that a certain performance index becomes minimized. In what follows, we consider the parametric learning of the logic processor with n inputs and a single output. The learning is completed in an supervised mode by considering a collection of pairs of input-output data $(\boldsymbol{x}(k), y(k))$, $k = 1, 2, \ldots, N$. The commonly used performance index is a sum of squared error (Euclidean distance)

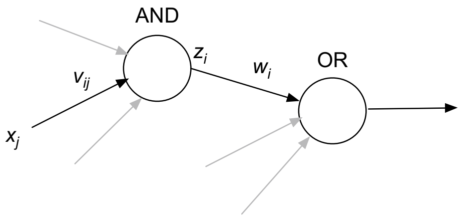


Fig. 14.11 Development of detailed learning scheme for the weights of the logic processor

$$Q = \sum_{k=1}^{N} ((y(k) - \hat{y}(k))^2 \tag{14.6}$$

Here $\hat{y}(k)$ is the output of the network for the input $\boldsymbol{x}(k)$. The gradient-based method is an iterative scheme of adjusting the values of the weights of the neurons,

$$\begin{aligned} w_i &= w_i - \alpha \frac{\partial Q}{Q w_i} \\ v_{ji} &= v_{ji} - \alpha \frac{\partial Q}{Q v_{ji}} \end{aligned} \tag{14.7}$$

where $\alpha > 0$ is a learning rate controlling intensity of changes of the weights and *iter* is an index of the iterative process. The operation $\lfloor\rfloor$ indicates that the values of the weights are retained (clipped) in the [0, 1] interval. Because of the iterative nature of the optimization method, the quality of learning depends on the initial configuration of the values of the weights. Usually one proceeds with the weights initialized in a random way (using a uniform distribution over the [0, 1] interval).

Let us look at the detailed learning realized for the logic processor. The details of the computing are presented by looking at the many-input single-output logic processor as illustrated in Fig. 14.11.

Considering the performance index (14.6), we compute

$$\frac{\partial Q}{\partial w_i} = -2 \sum_{k=1}^{N} (y(k) - \hat{y}(k)) \frac{\partial \hat{y}(k)}{\partial w_i} \tag{14.8}$$

Then

$$\frac{\partial \hat{y}(k)}{\partial w_i} = \frac{\partial}{\partial w_i} (A_i s(w_i t z_i)) \tag{14.9}$$

where A_i is expressed the result of computing the output of the OR neuron involving all inputs but z_i

$$A_i = S^h_{\substack{j=1 \\ j \neq i}} (w_j t z_j) \tag{14.10}$$

The partial derivative with regard to v_{ij} invokes the chain rule of differentiation

$$\frac{\partial \hat{y}(k)}{\partial v_{ij}} = \frac{\partial \hat{y}(k)}{\partial z_i} \frac{\partial z_i}{\partial v_{ij}} \tag{14.11}$$

Next we have

$$\frac{\partial \hat{y}(k)}{\partial z_i} = \frac{\partial}{\partial z_i}(A_i s(w_i t z_i)) \tag{14.12}$$

and

$$\frac{\partial z_i}{\partial v_{ij}} = \frac{\partial}{\partial v_{ij}}(B_j t(v_{ij} s x_j)) \tag{14.13}$$

where

$$B_j = \underset{\substack{l = 1 \\ l \neq j}}{T^n} (v_{il} s t x_l) \tag{14.14}$$

Detailed computing can be finalized once the t-norm and t-conorm standing in these expressions have been specified. For instance, for the product (t-norm) and the probabilistic sum (t-conorms) we obtain

$$\frac{\partial}{\partial w_i}(A_i s(w_i t z_i)) = \frac{\partial}{\partial w_i}(A_i + w_i z_i - A_i w_i z_i) = z_i(1 - A_i) \tag{14.15}$$

and

$$\frac{\partial}{\partial v_{ij}}(B_j t(v_{ij} s x_j)) = \frac{\partial}{\partial v_{ij}}\left(B_j \left(v_{ij} + x_j - v_{ij} x_j\right)\right) = B_j(1 - x_j) \tag{14.16}$$

The gradient-based method is the simplest learning scheme; one can consider here Newton method or stochastic approximation (see Appendix A); in this way the updates of the weights is made more efficient.

14.6 Interpretation of Logic Networks

Neurofuzzy systems can be translated (interpreted) as family of logic statements. Consider the network shown in Fig. 14.12.

This results in the three weighted logic statements combined with the use of *or* connective

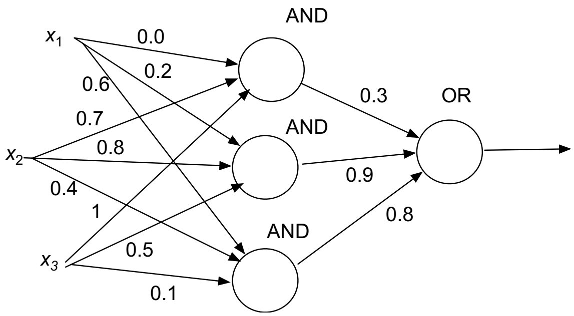


Fig. 14.12 Example logic processor

$$\left(x_{1|0.0} \text{ and } x_{2|0.7} \text{ and } x_{3|1.0}\right)_{|0.3}$$

or

$$\left(x_{1|0.2} \text{ and } x_{2|0.8} \text{ and } x_{3|0.5}\right)_{|0.9}$$

or

$$\left(x_{1|0.6} \text{ and } x_{2|0.4} \text{ and } x_{3|0.1}\right)_{|0.8}$$

The weights of the neurons come as a part of the logic statements (included in the description with the use of the bar subscripts). The interpretation can be further facilitated by focusing on the most essential parts of the network. Recall that a high relevance of the connection of the OR neuron is the one whose weight is close to 1. In contrast, the weights of the AND neuron are those assuming low values. Having this in mind, prior to its interpretation, the network could be trimmed by retaining only those significant connections. This means that only high values of the OR connections have to be retained. In case of AND neurons, connections above some threshold values are relevant and have to be kept. To reflect these observations, two threshold operations are introduced

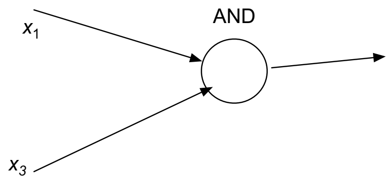
- OR neurons. The weights over some threshold level λ are kept; others are set to zero. The modified weight w_λ reads as follows

$$w_\lambda = \begin{cases} w \text{ if } w \geq \lambda \\ 0, \text{ otherwise} \end{cases} \tag{14.17}$$

- AND neurons. The weights above some threshold level μ are made irrelevant by raising their values to 1. We have

$$w_\mu = \begin{cases} w \text{ if} w \leq \mu \\ 1, \text{ otherwise} \end{cases} \tag{14.18}$$

Fig. 14.13 Reduced logic network obtained with the use of the pruning procedure



The higher the value of λ and the lower the value of μ, more connections are eliminated and more compact interpretation is formed. For instance, for the above network and its interpretation, by setting $\lambda = 0.85$ and $\mu = 0.75$ one obtains the reduced expression $(x_{1|0.7}\ and\ x_{3|0.4})_{|0.9}$.

Obviously, the threshold values result in the network whose numeric performance expressed by Q becomes worse. This offers a certain guideline as to the selection of admissible threshold level. Denote by Q as a function of λ and μ, $Q(\lambda, \mu)$: change the threshold values allowing only for a limited deterioration of the performance of the network, say $Q(\lambda, \mu)/Q$ should not exceed the value $(1 + \varepsilon)$ with the value of ε defined in advance.

The threshold operations can be made more decisive in the sense that the network is made Boolean so that only 0–1 weights are obtained once the thresholding has been completed. Here the thresholding formulas are expressed as follows

$$w_\lambda = \begin{cases} 1 \text{ if } w \geq \lambda \\ 0, \text{ otherwise} \end{cases} \tag{14.19}$$

$$w_\mu = \begin{cases} 0 \text{ if} w \leq \mu \\ 1, \text{otherwise} \end{cases} \tag{14.20}$$

Example 2 Following (14.19), (14.20), show reduced rules of the logic processor for the threshold values of λ and μ selected as 0.9 and 0.6, respectively.

The weakest connections are dropped as illustrated in Fig. 14.13 and this leads to the reduced rule in the following form y is x_1 and x_3.

The thresholding process has been applied to the network already optimized. One can take an alternative route by imposing the interpretability requirement as a part of the minimized objective function. It is now composed of the following two parts

$$Q = \sum_{k=1}^{N} ((y(k) - \hat{y}(k))^2 \beta \sum h(\text{weights}) \tag{14.21}$$

The first one is the original one responsible for the quantification of the differences between the data and the outputs of the networks while the second one is a cumulative entropy of the weights h(weight) which penalizes the values of the weights different from 0 or 1. Recall that $h(.)$ is a function attaining its maximum for the argument set to ½ while $h(0) = h(1) = 0$. For instance, for the logic processor with n inputs,

m outputs and h AND neurons, the detailed expression of the second term comes in the form

$$\sum h(\text{weights}) = \sum_{i,j} h(w_{ij}) + \sum_{i,j} h(v_{ij}) \tag{14.22}$$

An additional parameter β is helpful in striking a sound balance between the quality of approximation delivered by the network and its interpretability.

14.7 Logic Network in Modeling the Geometry of Data

Logic expressions deliver an interpretation of interfaces. Consider a two-input situation where for x_1 there are five fuzzy sets A_1, A_2, …, A_5 and three fuzzy sets B_1, B_2, and B_3 formed on x_2. The logic network composed of AND neurons followed by a single OR neuron develops a certain topology in the two-dimensional space. The AND neurons form a Cartesian product of A_i and B_j and then they are combined OR-wise; Fig. 14.14 displays the consecutive phases how the regions are formed and combined. The architecture composed of OR neurons forms the union of regions associated with the fuzzy sets in x_1 and x_2. The results can be produced by first

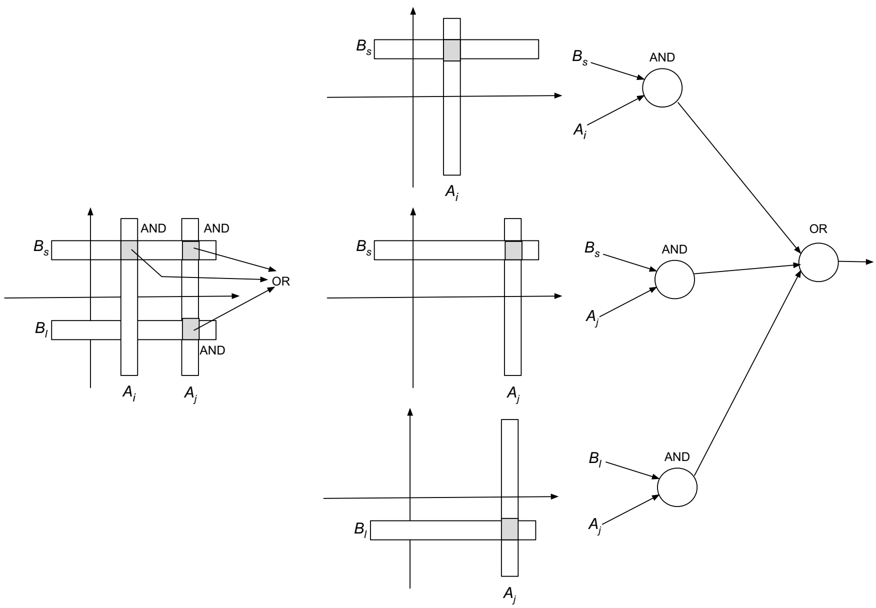


Fig. 14.14 Geometry of regions formed by AND neurons followed by a single OR neuron: consecutive steps of development

forming regions with the use of the OR operators and then they are selected (filtered) using the AND neurons.

The building the above geometric constructs is constructed in an adaptive manner because of the values of the weights produced during the learning process.

14.8 Fuzzy Relational Factorization

Let us recall the concept of nonnegative matrix factorization (NMF), which is one of the well-known methods of dimensionality reduction [8, 11]. An original matrix of data X is factorized into a matrix data Y of lower dimensionality and some matrix R linking the original data and those of reduced dimensionality, namely $X \approx YR$. The augmentation of this approach is to bring logic-based computing by accommodating t-norms and *t*-conorms.

Given is a collection of data organized in an $N \times n$ matrix X, $\dim(X) = Nn$ composed of vectors located in the $[0, 1]^n$ dimensional hypercube. The objective is to factorize the data through some fuzzy relations so that their dimensionality reduction becomes realized (Di [9, 10]. In essence, the factorization is expressed in the following form:

s-t composition

$$x_{ij} = S_{l=1}^{p}(y_{il} t r_{lj}) \tag{14.23}$$

t-s composition

$$x_{ij} = T_{l=1}^{p}(y_{il} s r_{lj}) \tag{14.24}$$

In light of the use of triangular norms and conorms, the processing is apparently logic oriented.

To solve the above factorization problem, one need to determine simultaneously the fuzzy relation R and the relation of reduced data Y. There are two phases of the design process, namely the structural optimization followed by the parametric optimization. The structural optimization involves the dimensionality of the fuzzy relation (p) and the choice *t*-norms and *t*-conorms.

The parametric optimization is about the adjustment of the entries of the fuzzy relation R. The performance index Q used in the optimization process is a sum of squared errors between the corresponding X and $\hat{X}$ expressed as the following sum of the squared differences between the individual entries of the relations

$$Q = \sum_{i=1}^{N} \sum_{j=1}^{n} (x_{ij} - \hat{x}_{ij})^2 \tag{14.25}$$

$i = 1, 2, \ldots, N; j = 1, 2, \ldots, n$. For the factorization problem (14.25) one has the following gradient-based iterative scheme (here *iter* stands for the index of the iterative optimization process)

$$y_{st}(iter+1) = y_{st}(iter) - \alpha \frac{\partial Q(iter)}{\partial y_{st}(iter)}$$

$s = 1, 2, \ldots, N; t = 1, 2, \ldots, p$.

$$r_{st}(iter+1) = r_{st}(iter) - \alpha \frac{\partial Q(iter)}{\partial r_{st}(iter)} \tag{14.26}$$

$s = 1, 2, \ldots, p; t = 1, 2, \ldots, n$.
where α is a positive learning rate.

Example 3 A special case of relational factorization is formulated as follows: Factorize n-by-n fuzzy relation R into two fuzzy relations so that R can be expressed through the following composition $R = G \circ G$ with $\circ$ being an s-t composition; assume that t-norm is the production and the t-conorm is the probabilistic sum.

The relational factorization implies that following detailed relationship

$$r_{ij} = S_{l=1}^{n}(g_{il} t g_{lj}) \tag{14.27}$$

$i, j = 1,2, \ldots, n$ where $G = [g_{ij}]$ is to be determined. The minimized performance is a sum of squared error

$$Q = \sum_{i,j=1}^{n} (r_{ij} - S_{l=1}^{n}(g_{il} t g_{lj}))^2 \tag{14.28}$$

whose minimum is determined by engaging a gradient-based method

$$g_{ij}(iter+1) = g_{ij}(iter) - \alpha \frac{\partial Q}{\partial g_{ij}} \tag{14.29}$$

(in the learning process, the values of the resulting relation are kept in the unit interval).

14.9 Fuzzy Neurofuzzy Models: Examples

Here we present of architectures of fuzzy neurofuzzy models whose design is aimed at the developments of logically inclined (interpretable structures) carried out in the presence of high-dimensional data. Recall that due to the concentration effect the "standard" construction of rule-based models involving fuzzy clustering.

14.9.1 Fuzzy Rule-Based Model with a Logic Autoencoder

An architecture and a way of the design of a rule-based model presented here is of interest when dealing with highly dimensional data. The key characteristics include: (i) interpretation of condition parts of the rules, (ii) logic-based processing carried out by an autoencoder, and (iii) ability to develop rule-based models for high-dimensional data (Fig. 14.15).

The overall architecture portraying the main functional modules are shown in Fig. 14.16.

Let us describe the underlying functionality supported by the individual modules.

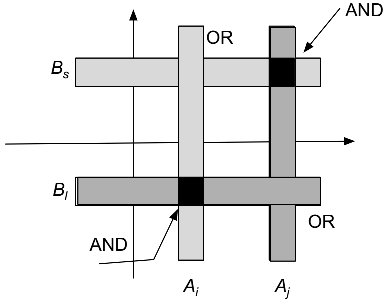


Fig. 14.15 Geometry of regions formed by OR neurons followed by a single AND neuron: consecutive steps of development

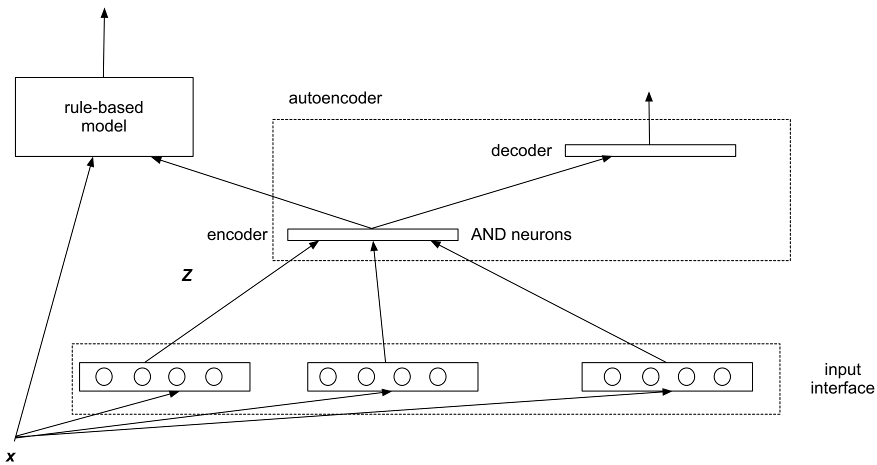


Fig. 14.16 Overall architecture of fuzzy rule-based model

Input interface For each input variables defined is a collection of fuzzy sets. They transform values assumed by the individual input variables $x_1, x_2, \ldots, x_n$ into degrees of membership of the corresponding fuzzy sets. Assuming that for each variable we have c fuzzy sets, in total the number of variables produced by the input interface is cn. Formally, the input interface transforms the vector $\boldsymbol{x} = [x_1\ x_2\ \ldots\ x_n]$ in $\boldsymbol{R}^n$ to the vector z located in the cn dimensional unit hypercube.

Logic autoencoder reduces the dimensionality of z to some vector $\boldsymbol{u}$ in $[0, 1]^p$ where $p \ll cn$. The encoding is realized as a collection of AND neurons. The optimization of the autoencoder is completed by adjusting the weights of the encoder and the decoder so that the result of decoding is as close to the original input is as close as possible

Rule-based model The model is composed of p rules coming in the following form

$$\begin{aligned} &\text{if } \boldsymbol{u} \text{ is } [1\,0\ldots0] \text{ then } y = L_1(\boldsymbol{x}; \boldsymbol{a}_1) \\ &\text{if } \boldsymbol{u} \text{ is } [0\,10\,..0] \text{ then } y = L_2(\boldsymbol{x}; \boldsymbol{a}_2) \\ &\ldots \\ &\text{if } \boldsymbol{u} \text{ is } [0\ \ldots0\,1] \text{ then } y = L_p\left(\boldsymbol{x}; \boldsymbol{a}_p\right) \end{aligned} \tag{14.30}$$

where $L_1(\boldsymbol{x}; \boldsymbol{a}_1)$, $L_2(\boldsymbol{x}; \boldsymbol{a}_2)$, …, $L_p(\boldsymbol{x}; \boldsymbol{a}_p)$ are local models forming the respective conclusions of the rules. Usually L_i are linear functions.

Autoencoder Autoencoder is commonly regarded as a system to reduce data dimensionality. Here we consider a logic-driven autoencoder in which encoding is realized by AND logic neurons. This facilitates interpretation of the condition part of the rule-based model.

In what follows, we discuss the detailed computing carried out by the individual modules.

The input interface is formed through a collection of fuzzy sets. They could be selected in advance such as triangular fuzzy sets distributed uniformly across the universe of discourse. Another option is to determine fuzzy sets through fuzzy clustering. As we are concerned with one-dimensional data, the computing overhead is not essential.

The autoencoder is a single level structure. The encoding is governed by the logic expression which in essence is a layer composed of AND neurons. The outputs u_1, u_2, …, u_p are computed as follows

$$\boldsymbol{u} = \text{AND}(z; W) \tag{14.31}$$

$$u_i = T_{j=1}^{cn}(z_j s w_{ij}) \tag{14.32}$$

$W = [w_{ij}]$, $i = 1, 2, \ldots, p$, $j = 1, 2, \ldots, cn$. The results exhibit a transparent interpretation in the form of the weighted *and* combination of the inputs of the autoencoder. The decoding is computed in the form

$$\hat{z} = f(\boldsymbol{u}, V) \tag{14.33}$$

namely

$$\hat{z}_i = f\left(\sum_{j=1}^{p} u_j v_{ij}\right) \tag{14.34}$$

where f is a monotonically increasing function assuming values in [0, 1] and $V = [v_{ij}]$, $i = 1,2, \ldots, nc, j = 1, 2, \ldots, p$. Commonly f is a sigmoidal function.

The rule-based model returns the output which is a weighted sum of the local models with weights formed as the outputs of the autoencoder $\boldsymbol{u}$. It is computed as

$$\hat{y} = \sum_{i=1}^{p} u_i^{'} L_i(x; a_i) \tag{14.35}$$

where $\boldsymbol{u}'$ is a normalized version of $\boldsymbol{u}$, viz. $u_i' = u_i / \sum_{j=1}^{p} u_j$

The design process is consistent with the architecture and consists of several main phases. The learning is carried out in a supervised mode in the presence of input-output pairs $(\boldsymbol{x}(k), y(k))$, $k = 1, 2, \ldots, N$.

Fuzzy sets of the input interface are predetermined (say, as triangular membership functions) or designed by running Fuzzy C-Means for one dimensional data $x_i(k)$, $k = 1, 2,.., N$; $i = 1, 2, \ldots, c$. The number of clusters (c) has to be fixed in advance.

The autoencoder is constructed assuming some structural information such as p entailing the level of dimensionality reduction. t-norms and t-conorms are to be set up in advance as well. Next parametric optimization is involved by minimizing a reconstruction error in the form

$$Q1 = \sum_{j=1}^{cn} (z - \hat{z})^2 \tag{14.36}$$

The fuzzy rule-based model requires an optimization of the parameters of the local linear functions $L_i(\boldsymbol{x}; \boldsymbol{a}_i) = a_{i0} + a_{i1}x_1 + \cdots + a_{ip}x_p$. The parameters are adjusted by minimizing the following performance index

$$Q = \sum_{j=1}^{cn} (y(k) - \hat{y}(k))^2 \tag{14.37}$$

so $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_p$ are those which minimize Q.

The development of the model requires that we first reduce dimensionality and choose an optimal value of p. Next the rule-based component is constructed.

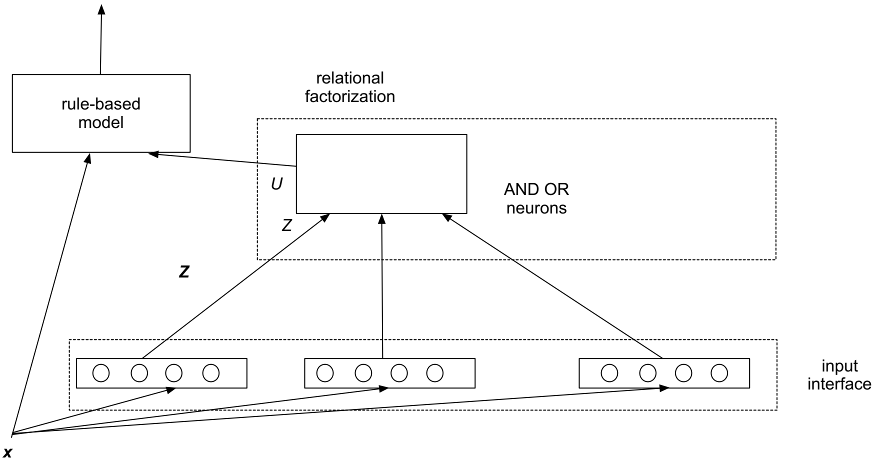


Fig. 14.17 Design of fuzzy rule-based model with the use of relational factorization

14.9.2 Design with Relational Factorization

Functionally, the architecture of the model is similar to the one as shown above and the main differences lies in the way of logic reduction of input data which in now completed by the relational factorization, see Fig. 14.17.

The input data are processed by an input interface through the use of a collection of input fuzzy sets. Thus any input $\boldsymbol{x}$ is transformed to cn-dimensional vector $\boldsymbol{z}$. In this way, all data $\boldsymbol{x}(1), \ldots, \boldsymbol{x}(N)$ give rise to $\boldsymbol{z}(1), \boldsymbol{z}(2), \ldots, \boldsymbol{z}(n)$ which are arranged in a matrix Z of dimensionality $N\,cn$. They are subject to relational factorization using the max-t or min-s composition $Z = U\,R$ or $Z = U\,R$, dim $(U) = N\,p$ with $p \ll cn$. Each row of the matrix $\boldsymbol{u}(1), \boldsymbol{u}(2), \ldots, \boldsymbol{u}(N)$ forms the input to the rule-based model whose rules come in the form

$$
\begin{aligned}
\text{if } \boldsymbol{u} \text{ is } [1\,0 \ldots 0] \text{ then } y &= L_1(\boldsymbol{x}; \boldsymbol{a}_1) \\
\text{if } \boldsymbol{u} \text{ is } [0\,10 \,..0] \text{ then } y &= L_2(\boldsymbol{x}; \boldsymbol{a}_2) \\
&\ldots \\
\text{if } \boldsymbol{u} \text{ is } [0\, \ldots 0\,1] \text{ then } y &= L_p\left(\boldsymbol{x}; \boldsymbol{a}_p\right)
\end{aligned}
\tag{14.38}
$$

where $L_1(\boldsymbol{x}; \boldsymbol{a}_1), L_2(\boldsymbol{x}; \boldsymbol{a}_2), \ldots, L_p(\boldsymbol{x}; \boldsymbol{a}_p)$ are local linear models forming the respective conclusions of the rules.

The output of the model is a weighted sum of the local models with weights being the result of the factorization. In general for any $\boldsymbol{x}$ one obtains $\boldsymbol{u}$ and then

$$\hat{y} = \sum_{i=1}^{p} u_i' L_i(x; a_i) \tag{14.39}$$

where $\boldsymbol{u}'$ is a normalized version of $\boldsymbol{u}$, viz. $u_i' = u_i / \sum_{j=1}^{p} u_j$

14.10 Conclusions

Fuzzy neurocomputing has emerged as the junction of fuzzy sets and neurocomputing to address the genuine need for developing synergy with intent to construct systems that are both adaptive (owing to neural network features of the architectures) and interpretable (thanks to the use of fuzzy sets in their topologies). We highlighted selected architectures, which can be next used in a vast array of prediction and classification problems.

Problems

1. Having the OR neuron with weights $w_1, w_2, \ldots, w_n$, approximate it by some OR neuron with all weights of the same value w; Fig. 14.18. Determine optimal value of w and interpret the resulting neuron.
2. Represent the following logic expression with the use of a suitable fuzzy neural network

$$(x_1 \text{ is smaller than } a) \text{ or } (x_2 \text{ is similar to } b) \text{and} (x_3 \text{ or } x_4 \text{ are greater than } c)$$

where a, b, and c are given and x_1, x_2, x_3, and x_4 assume values in the unit interval.

3. Develop the detailed learning formulas for the logic network shown in Fig. 14.19. t-norm: minimum, t-conorm: maximum. Note that the derivatives can be defined as follows

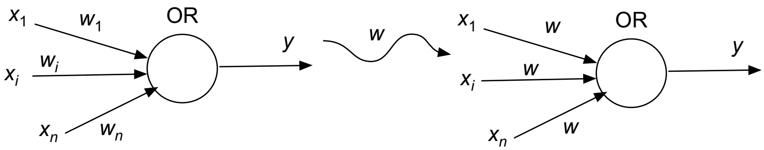


Fig. 14.18 Approximation of OR neuron

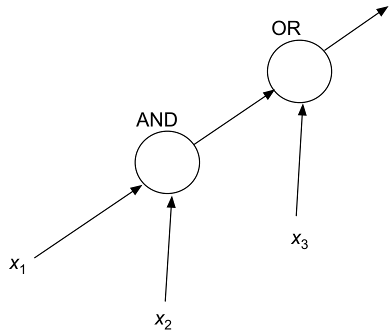


Fig. 14.19 Logic network composed of two neurons

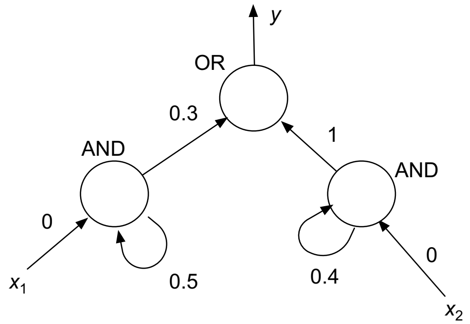


Fig. 14.20 Logic network with local feedback loops

$$\frac{d(\min(x,a))}{dx} = \begin{cases} 1, & \text{if} x \le a \\ 0, & \text{otherwise} \end{cases} \qquad \frac{d(\max(x,a))}{dx} = \begin{cases} 1, & \text{if} x \ge a \\ 0, & \text{otherwise} \end{cases}$$

4. Calculate the values $y(1)$, $y(2)$, … for the network with local feedback loops; Fig. 14.20. Initially, the neurons are at zero values $z(0) = u(0) = 0$. The input values are $x_1 = 1$ and $x_2 = 0.5$.

References

1. Haykin, S.: Neural Networks. A Comprehensive Foundation. N. York, Mcmillan (1994)
2. Nauck, D., Klawonn, F., Kruse, R.: Neuro-fuzzy Systems. J. Wiley, Chichester (1997)
3. Pedrycz, W., Rocha, A.F.: Fuzzy set-based models of neurons and knowledge-based networks. IEEE Trans. Fuzzy Syst. **1**, 254–266 (1993)
4. Crama, Y., Hammer, P.L.: Boolean Functions. Cambridge University Press, N. York (2011)
5. O'Donnel, R.: Analysis of Boolean Functions. Cambridge University Press, N. York (2014)
6. Schneeweiss, W.G.: Boolean Functions and Engineering Applications. Springer, Berlin (1989)
7. Pedrycz, W.: Neurocomputations in relational systems. IEEE Trans. Pattern Anal. Mach. Intel. **13**, 289–296 (1991)
8. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**, 788–791 (1999)

9. Nola, A.D., Sessa, S., Pedrycz, W.: Decomposition problem of fuzzy relations. Int. J. General Syst. **10**, 123–133 (1985)
10. Pedrycz, W., Hirota, K., Sessa, S.: A decomposition of fuzzy relations. IEEE Trans. Syst. Man Cybern. Part B (Cybernetics) **31**, 657–663 (2001)
11. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization, Adv. Neural Inf. Process. Syst. 556–562 (Vancouver, BC, Canada, Dec 2001)

Chapter 15
Applications: Granular Models, Granular Classifiers and Fuzzy Cognitive Maps

Abstract In this chapter, we introduce concepts and discuss algorithmic developments of several key categories of models which dwell on ideas of fuzzy sets and information granules: granular fuzzy models, granular classifiers and fuzzy cognitive maps. It is shown that granular results convey more information about the quality of results.

15.1 Granular Models

In a nutshell, granular fuzzy models are built on a basis of a collection of information granules. They serve as functional blocks whose arrangements and associations give rise to the model. The design of the models efficiently uses sound information granules which capture the essence of the training data. The granules give rise to a interpretable blueprint of the model. The aggregation of granules is intuitively appealing. In virtue of the level of abstraction delivered by information granules, there are two essential features. First, the granular format of the results delivered by granular models is more comprehensive and better suitable to assess their quality vis-à-vis the experimental data. Second, the level of abstraction delivered by the granular model enhances interpretation and comprehension capabilities of the model—a feature solidly positioned on the agenda of explainable Artificial Intelligence (XAI).

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3_15

15.1.1 Granular Model—A Design with the Use of the Augmented Principle of Justifiable Granularity

Let us consider multiple input—single output input-output data $(\boldsymbol{x}_k, y_k)$, $k = 1, 2, \ldots, N$, dim $(\boldsymbol{x}_k) = n$. Let us assume that the data are normalized. The mechanism of linear normalization can be sought here. We develop a granular model formed on a basis of information granules. The granules are formed in the n-dimensional input space by engaging the principle of justifiable granularity. The generic version of the principle is augmented by taking into account information about the output data contributing to the formation of information granules.

The following are the fundamental steps of the design.

Building a Collection of Numeric Representatives (Prototypes) in the Input Space

We consider here the usage of the FCM clustering algorithm which results in c prototypes $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c$.

Formation of Information Granules

The principle of justifiable granularity is considered to build an information granule A_i positioned around $\boldsymbol{v}_i$ by determining the radius ρ_i in such a way that the following product becomes maximized with respect to ρ_i

$$V(\rho_i) = \text{cov}(A_i)\text{sp}(A_i)\exp(-\beta\sigma_i) \tag{15.1}$$

The output values associated with A_i are characterized by the following mean

$$\bar{y}_i = \sum_{x_k \in A_i} y_k / card\{x_k \in A_i\} \tag{15.2}$$

where σ_i is the standard deviation of output data y_k s such that the corresponding $\boldsymbol{x}_k$ belongs to A_i that is

$$\sigma_i^2 = \sum_{x_k \in A_i} (y_k - \bar{y}_i)^2 / card\{x_k \in A_i\} \tag{15.3}$$

The parameter β, $\beta > 0$, controls an impact of the diversity of the output values contain in A_i. In this way, the information about the output is actively used in the formation of the granule. Furthermore we may monitor a condition of non-overlap of the granules requesting that for any $i \neq j$

$$A_i \cap A_j = \emptyset$$

In sum, the information granules A_i are concisely characterized by their geometric characteristics (viz. the location of the prototype $\boldsymbol{v}_i$ and the radius ρ_i) and associated information content, viz. the mean and the standard deviation of the output variable

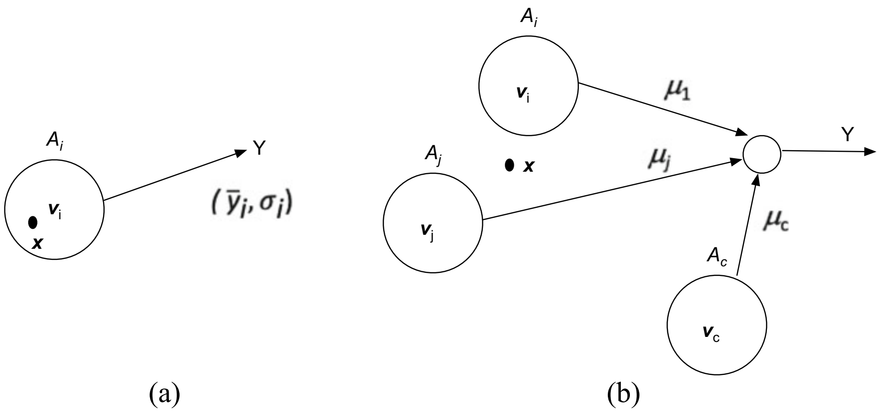


Fig. 15.1 A blueprint of the granular model along with inference mechanism for given $\boldsymbol{x}$: **a** $\boldsymbol{x}$ included in A_i, **b** $\boldsymbol{x}$ outside the union of A_is

associated with the input information granule. In other words, one has $A_i = A_i$ ($\boldsymbol{v}_i$, ρ_i; y_i, σ_i). Such A_is form the blueprint of the model by reflecting the structural content of the data.

The granular mapping completed by the above pairs is used to compute the granular output $Y = (\hat{y}, \hat{\sigma})$ for any input $\boldsymbol{x}$. There are two situations distinguished, refer also to Fig. 15.1, $\boldsymbol{x}$ belongs to one of A_i Fig. 15.1a. Y is an output information granule associated with A_i, namely $Y = (\hat{y}, \hat{\sigma}) = (\bar{y}_i, \sigma_i)$.

$\boldsymbol{x}$ is located outside the union of information granules. In this case, a certain approximation mechanism is invoked. We calculate a degree of matching of $\boldsymbol{x}$ with each A_i

$$\mu_i(x) = \frac{1}{\sum_{j=1}^{c} \left(\frac{||x-v_i||^2}{||x-v_j||^2} \right)^{1/(m-1)}} \tag{15.4}$$

(note that this formula is the same as we have already being used in the FCM algorithm). The weighted aggregation is now carried out which returns the center of the information granule and its standard deviation in the output space

$$\hat{y} = \sum_{i=1}^{c} \mu_i(x)\bar{y}_i \tag{15.5}$$

$$\hat{\sigma} = \sum_{i=1}^{c} \mu_i(x)\sigma_i \tag{15.6}$$

In both cases (i), (ii) the result is an information granule $Y = (\hat{y}, \hat{\sigma}) = [\max(0, \hat{y} - \hat{\sigma}, \min(1, \hat{y} + \hat{\sigma})]$. The clipping done by the maximum and minimum operations assure that the results are contained in the unit interval. Recall that we are concerned with the normalized output data assuming values in [0, 1]. The quality of the granular model (granular mapping) is assessed by engaging the criteria of coverage and specificity when looking at the obtained results. Considering input $\boldsymbol{x} = \boldsymbol{x}_k$, we obtain Y_k and $\hat{\sigma}_k$ produced by the model and the quality of this information granule is assessed with regard to y_k,

$$\mathrm{cov} = \frac{1}{N}\sum_k \mathrm{incl}(y_k, Y_k)$$
$$\mathrm{sp} = \frac{1}{N}\sum_k \max(0, 1 - \hat{\sigma}_k) \tag{15.7}$$

where

$$\mathrm{incl}(y_k, Y_k) = \begin{cases} 1, \text{if} y_k \in Y_k \\ 0, \text{otherwise} \end{cases}$$

Their product serves as the performance measure.

As usual, the evaluation of the granular model is realized for the training and testing data with the coverage expressed in the form $\mathrm{cov} = \frac{1}{N_{\mathrm{train}}}\sum_k \mathrm{incl}(y_k, Y_k)$, $\mathrm{cov} = \frac{1}{N_{\mathrm{test}}}\sum_k \mathrm{incl}(y_k, Y_k)$ and $\mathrm{sp} = \frac{1}{N_{\mathrm{train}}}\sum_k \max(0, 1-\hat{\sigma}_k)$, $\mathrm{sp} = \frac{1}{N_{\mathrm{test}}}\sum_k \max(0, 1-\hat{\sigma}_k)$ with N_{train} and N_{test} standing for the number of data in the training and testing data, respectively.

15.1.2 Determining Information Granules in the Input Space Through Mapping to One-Dimensional Spaces

In case of highly-dimensional input space (where the commonly used design method based on fuzzy clustering could exhibit a limited usefulness), the granular mapping can be formed by starting with an information granule defined in the output space and then building the associated information granules in the corresponding input spaces. Figure 15.2 illustrates the essence of the development:

- we define an information granule Z_i in the output space. For instance, one may require that its specificity is not lower than some predefined threshold.
- considering that Z_i is an interval information granule, it embraces a subset of all data, say $\boldsymbol{X}_i$. For each input variable of the data in $\boldsymbol{X}_i$., we construct an information granule (making use of the principle of justifiable granularity), say A_i, B_i, C_i, …. If Z_i is a fuzzy set then this form of information granule invokes a weighted version of the principle of justifiable granularity.

In this way, for each Z_i associated are input information granules in the corresponding spaces of input variables

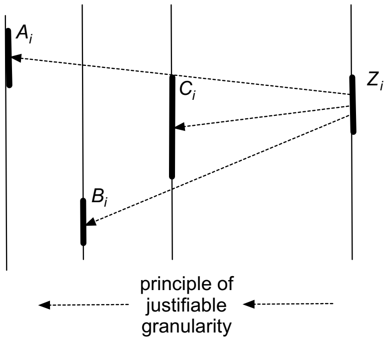


Fig. 15.2 Building information granules in one-dimensional input spaces

$$(A_i,\ B_i,\ C_i, \ldots)Z_i \tag{15.8}$$

By comparing this approach with the design presented in the previous section, we note that it constitutes a computationally sound version of the previous one as the associated information granules are built for individual input variables.

15.1.3 Development of Granular Fuzzy Models with the Use of Conditional FCM

The architecture of this model dwells upon the idea of conditional FCM; refer to Chap. 8.

Let us briefly recall that the term conditional FCM underlines a fact that clustering is carried out in a certain context by delivering conditioned process of clustering realized for data in the output space [1, 8]. We are interested in the design of information granules- fuzzy sets in the input space. As usual the training data are composed as input-output pairs ($\boldsymbol{x}_k$, $target_k$), $k = 1, 2, \ldots, N$.

In the output space we form a family of context fuzzy sets $B_1, B_2, \ldots, B_p$. Each of them implies some regions in the input space; see Fig. 15.3.

Considering some context B_l, the data are clustered, refer to Chap. 8, Sect. 8.9. Assuming that the number of clusters set is c_l, one forms fuzzy sets $A_{l1}, A_{l2}, \ldots, A_{lcl}$. Two points of this construct are worth stressing. First, context-based clustering is less computationally demanding as each context implies only a subset of the data to be clustered. Second, the resulting information granules in the input space are linked with the output information granule (fuzzy sets) used as the context in the clustering algorithm. In this way, we envision an emergence of the mapping from some regions of the input space to the region in the output space. In the same way context-based clustering is completed for all remaining subsets of the input data. Overall, we arrive at following fuzzy sets located in the input space

For context B_1: $A_{11}, A_{12}, \ldots, A_{1c1}$

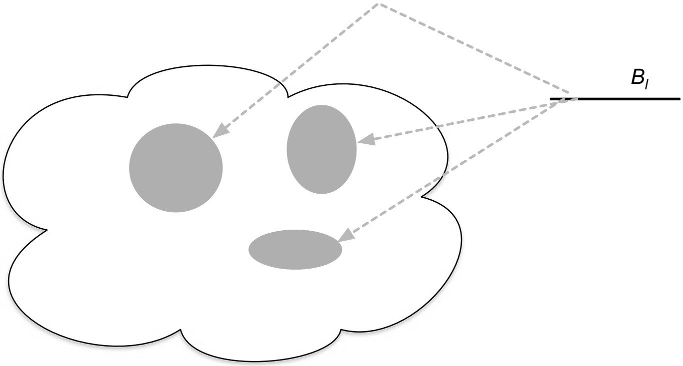


Fig. 15.3 Regions in input data space induced by the context fuzzy set B_l

For context B_2: $A_{21}, A_{22}, \ldots, A_{2c2}$
For context B_p: $A_{p1}, A_{p2}, \ldots, A_{pcp}$

Each A_{ij} comes with its prototype $\boldsymbol{v}_{ij}$. As before, these information granules form a blueprint of the granular model. A way in which they have been formed, gives rise to an architecture displayed in Fig. 15.4.

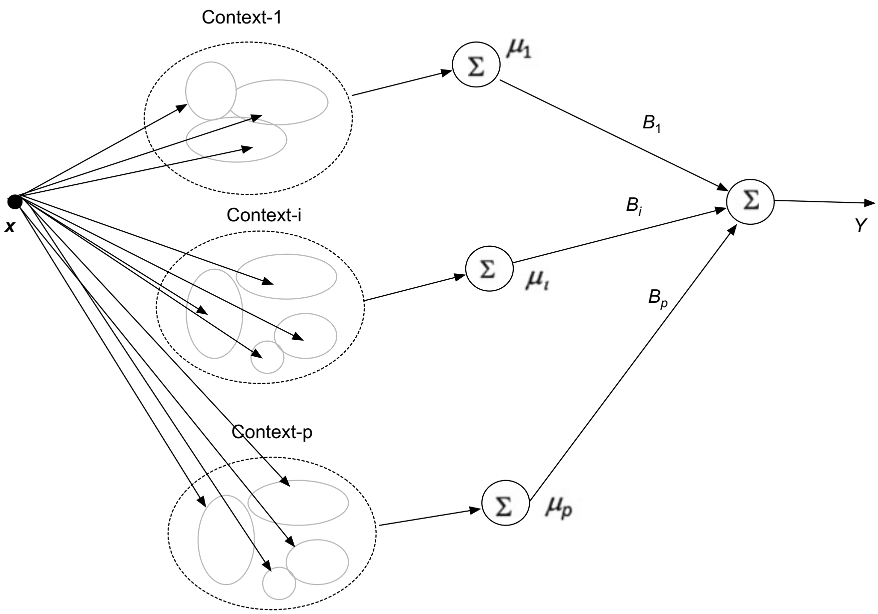


Fig. 15.4 Architecture of the granular fuzzy model

It is worth noting that once the fuzzy sets have been formed, the design process has been completed. Proceeding with the detailed computing for given input $\boldsymbol{x}$, the following phases are encountered:

The corresponding degrees of activation of the information granules μ_{ij} are computed in the following way (compare the expression below with the formula for the calculation of membership grades used in the FCM method); note that in total we have $c1 + c2 + \cdots + cp$ prototypes

$$\mu_{ij}(x) = \frac{1}{\sum_{l=1}^{p}\sum_{t=1}^{cl}\left(\frac{||x-v_{ij}||^2}{||x-v_{lt}||^2}\right)^{1/(m-1)}} \tag{15.9}$$

$i = 1, 2, \ldots, p$; $j = 1, \ldots, ci$. In the sequel the levels of activation of fuzzy sets generated by the given context are aggregated

$$\mu_i = \sum_{j=1}^{ci} \mu_{ij} \tag{15.10}$$

where the addition is realized at the summation nodes forming the second layer of the overall. The final layer is a granular neuron whose weights are granular being the context fuzzy sets B_j. This gives rise to the granular output Y

$$Y = (B_1 \otimes \mu_1) \oplus (B_2 \otimes \mu_2) \oplus \ldots \oplus (B_p \oplus \mu_p) \tag{15.11}$$

where $\oplus$ and $\otimes$ are the operations of addition and multiplication whose arguments are fuzzy sets.

If all Bis are fuzzy sets having the same type of membership functions of finite support, say triangular, trapezoidal, parabolic the obtained membership function has the same type. For instance, if B_i has a triangular membership function with the bounds and the modal value b_i^-, m_i, b_i^+, respectively, then Y has a triangular membership function with the parameters y^-, m_y, y^+

$$\begin{aligned} y^- &= \sum_{i=1}^{p} \mu_i b_i^- \\ m_y &= \sum_{i=1}^{p} \mu_i m_i \\ y^+ &= \sum_{i=1}^{p} \mu_i b_i^+ \end{aligned} \tag{15.12}$$

While the above design is predominantly based on the use of information granules produced through fuzzy clustering and no further optimization has been considered, one could envision some augmentations. First, the quality of the model is associated with the number of context fuzzy sets p and the number of clusters $c1$, $c2$, …, cp. All of these parameters could be subject to optimization leading to some structural enhancements of the model. The fuzzification coefficient m brings another component of flexibility. Second, instead of taking a sum in (15.10), various aggregation operators could be sought.

Given the granular nature of the data, the quality of the model is evaluated by evaluating the coverage and specificity of the granular results. We have $\text{cov} = \sum_{k=1}^{N} Y_k(y_k)$ and $\text{sp} = \sum_{k=1}^{N} sp(Y_k)$. The product $V = \text{cov} * \text{sp}$ of these two measures can serve as the performance index of the granular model; the higher this product is, the higher the quality of the resulting model. Further optimization can engage the maximization of V with respect to the parameters, say the fuzzification coefficient, the number of contexts, etc.

15.2 Granular Classifiers

Granular classifiers are classifiers whose construction and functioning are inherently associated with information granules and the concept of information granularity [12]. Granular classifiers feature two main properties:

(i) the classifier is interpretable and explainable: there are clearly visible classification rules behind the obtained results. One can also deliver an explanation why the pattern $\boldsymbol{x}$ has been classified in a particular way.
(ii) the results of classification are quantified by degrees of membership so some flagging mechanism as to the quality of classification results can be incorporated here.

In what follows, we consider a classification problem with p classes while the learning set is composed of a collection of pairs $\boldsymbol{D} = (\boldsymbol{x}_k, \boldsymbol{\phi}_k)$ where $\boldsymbol{x}_k$s are n-dimensional vectors and $\boldsymbol{\phi}_k \in \{0, 1\}^p$, $k = 1, 2, \ldots, N$. We assume that the data are normalized viz. located in the n-dimensional unit hypercube $[0, 1]^n$.

An overall architecture of the classifier is displayed in Fig. 15.5. There are two main phases. In the first one; built are information granules. In the second phase developed is a classification mechanism.

With our focus is on to the development of the granular classifier, the interest is a two-step buildup of information granules.

(i) The data are clustered with the use of some commonly used algorithms, in particular FCM and its generalizations. The result comes in the form of prototypes $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c$ and partition matrix U where c is the number of clusters, $c \geq p$. This structure delivers a preliminary blueprint for building information granules.

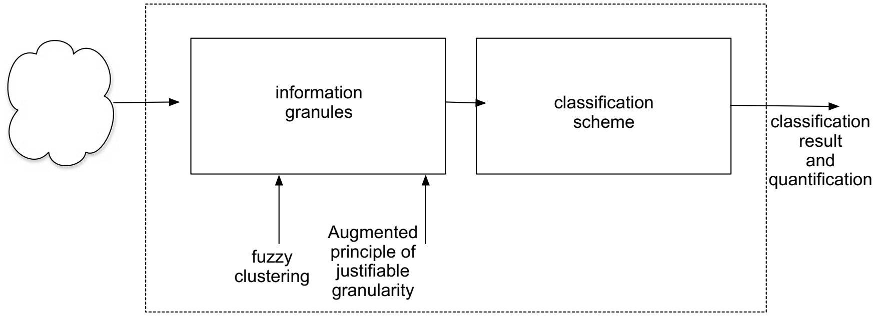


Fig. 15.5 A general architecture and functioning of granular classifiers

(ii) the augmented principle of justifiable granularity is used to build an information granule spread around the corresponding prototype. The refinement of the principle comes from the fact that in the design of the granule we take into account the requirement of homogeneity of the granule with respect to its class content of the patterns belonging to the granule. Let us recall that in addition to the "standard" requirements of coverage, cov(.,) and specificity sp(.), the entropy h regarded as a measure of homogeneity is considered. The coverage of the information granule Ω_i spanned over $\boldsymbol{v}_i$ is expressed in a usual way

$$\mathrm{cov}(\Omega_i) = \sum\nolimits_{x_k:||x_k - v_i||^2 \le n\varrho_i^2 u_{ik}} \tag{15.13}$$

The specificity is taken in its simple format by admitting a linearly decreasing function of the radius

$$\mathrm{sp}(\boldsymbol{\Omega}_i) = 1 - \rho_i \tag{15.14}$$

$\rho_i \in [0, 1]$. For given ρ_i, Ω_i contains data whose statistics of degrees of membership to classes are summarized in a class vector $\boldsymbol{\omega}_i$. The entries of this vector are determined as follows

$$\boldsymbol{\omega}_i = \left[n_{i1}/n_i n_{i2}/n_i \ldots n_{ip}/n_i\right] \tag{15.15}$$

where n_{i1}, n_{i2}, …, n_{ip} are the numbers of patterns contained in Ω_i and belonging to the corresponding classes and $n_i = \sum_{j=1}^{p} n_{ij}$

Ideally, an information granule Ω_i is fully homogeneous with respect to class content meaning that $\boldsymbol{\omega}_i$ has only a single nonzero entry (being equal to 1). The worst situation occurs when all entries of $\boldsymbol{\omega}_i$ are the same being equal to $1/p$. The entropy $h(\Omega_i)$ can be sought as a sound measure of quality of Ω_i with regard to its usefulness for classification purposes. In the first case, it attains the value equal to zero and in the second it has the highest value. The modified performance index is taken in the form of the following product of the individual criteria

$$V(\Omega_i) = \text{cov}(\Omega_i)\text{sp}(\Omega_i)(1 - h(\Omega_i)/h_{\max}) \tag{15.16}$$

Obviously the maximal value of entropy $h_{\max}$ occurs when all entries of $\boldsymbol{\omega}_i$ are the same and equal to $1/p$. The optimal information granule has the radius for which we attain the maximum of (15.16),

$$\rho_{i,}\text{max} = \ \arg\max_{\rho i} V(\Omega_i) \tag{15.17}$$

Once the optimization of $V(\Omega_i)$ has been completed, the obtained information granule becomes a building block of the classifier. Formally speaking Ω_i is concisely described as the four-element tuple

$$\Omega_i(x; v_i, \rho_i, \boldsymbol{\omega}_i) \tag{15.18}$$

with the associated entropy $h(\Omega_i)$. Noticeably, the granule is described by its geometry ($\boldsymbol{v}_i$, ρ_i) and the information (class) content $\boldsymbol{\omega}_i$. The entropy is a descriptor of the functional component of the ensuing classifier.

A collection of c information granules is used to realize classification of any new pattern $\boldsymbol{x}$.

In the realization of the inference process we distinguish three situations, refer to Fig. 15.6.

(i) The pattern $\boldsymbol{x}$ belongs to only single information granule Ω_i. The returned result is straightforward: the classification outcome $\hat{\omega}$ is equal to $\boldsymbol{\omega}_i$ associated with the ith information granule, $\hat{\omega} = \boldsymbol{\omega}_i$

In this case, the quality of classification is expressed by the entropy $h(\Omega_i)$. If we aim at the identification of a single class, the maximum rule is invoked: $\boldsymbol{x}$ belongs to class i_0 where

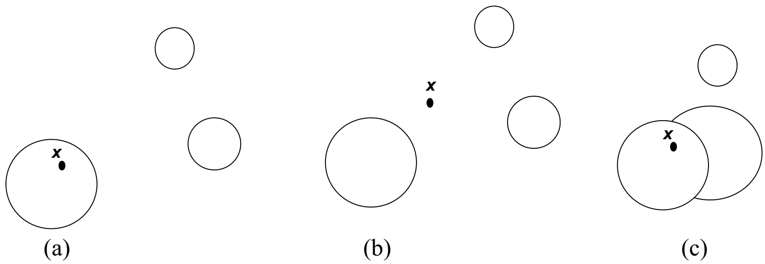


Fig. 15.6 Classification with the use of granular classifier: **a** $\boldsymbol{x}$ in Ω_i, **b** $\boldsymbol{x}$ outside information granules, and **c** overlapping Ω_is

$$i_0 = \arg\max_{i=1, 2, \ldots, p} \hat{\omega} \tag{15.19}$$

(ii) the pattern $\boldsymbol{x}$ does not belong to any Ω_i; refer to Fig. 15.6b. In this case, we determine a degree to which $\boldsymbol{x}$ matches (geometrically) each Ω_i and then view these levels of matching as weights attached to the information content of the corresponding information granules. Formally, we have the following linear combination

$$\hat{\omega} = \sum_{i=1}^{c} A_i(x)\omega_i \tag{15.20}$$

The degrees of matching $A_i(\boldsymbol{x})$ follow the membership formula which is the same as discussed in the FCM algorithm

$$A_i(x) = \frac{1}{\sum_{j=1}^{c} (\frac{x-v_i}{x-v_j})^{2/(m-1)}} \tag{15.21}$$

where m is the fuzzification coefficient used in the FCM clustering (as usual $m = 2$) and the distance ||.|| is the same as used in the FCM. Again, to arrive at the binary classification we use the maximum rule. Furthermore $h(\hat{\omega})$ serves as a measure quantifying the quality of the obtained classification result.

(iii) $\boldsymbol{x}$ belongs to several overlapping information granules, Fig. 15.6c. Here the calculations are the same as in (ii) but the determination of matching degrees involves only those information granules which embrace $\boldsymbol{x}$. We have the following expression

$$\hat{\omega} = \sum_{i \in I} A_i(x)\omega_i \tag{15.22}$$

$\boldsymbol{I}$ stands for the indices of A_is such that $\boldsymbol{x}$ is included in the corresponding information granules, $\boldsymbol{I} = \{i = 1, 2, \ldots, c \mid \Omega_i(\boldsymbol{x}) > 0\}$.

The classification error is determined by counting the number of situations when mismatch between $\hat{\omega}$ and $\boldsymbol{\phi}_k$ occurs. Denote by $i_{0,k} = \arg\max_{i=1, 2,\ldots,p} \phi_{k,i}$ and $\hat{i}_{0,k} = \arg\max_{i=1, 2,\ldots,p} \hat{\omega}_{k,i}$

$$\text{err} = 1/N \sum_{k=1}^{N} \delta(i_{0,k}, \hat{i}_{0,k}) \tag{15.23}$$

$\delta(i,j) = 0$ if $i = j$ and 1 otherwise. The non-Boolean nature of the classification results can be beneficial in flagging the quality of classification results delivered by the fuzzy set of class membership. A sound flagging criterion relies on the entropy of the result and comes with a classification metarule: if $h(\hat{\omega}) < \tau$ then accept classification outcome given by $\hat{\omega}$ otherwise additional analysis is invoked. Where τ is some threshold level whose value can be subject to a separate optimization process.

A sound value of τ is the one for which we achieve a small value of the classification error and the number of patterns to be inspected ds not exceed some predetermined limit.

Example 1 Consider that the three information granules are ideal in terms of information content, viz. $\boldsymbol{\omega}_1 = [1\ 0\ 0]$, $\boldsymbol{\omega}_2 = [0\ 1\ 0]$, and $\boldsymbol{\omega}_3 = [0\ 0\ 1]$. Determine the entropy of $\boldsymbol{x}$ if (i) $\boldsymbol{x}$ is included in one of the information granules, and (ii) $\boldsymbol{x}$ is outside all information granules.

(i) for $\boldsymbol{x}$ located in one of the information granules, the entropy is equal to zero. (ii) denote by λ_1, λ_2, and λ_3 the activation levels of the individual information granules. In virtue of (15.22) the vector of class membership is $[\lambda_1\ \lambda_2\ \lambda_3]$ and the entropy is $h(\lambda_1) + h(\lambda_2) + h(\lambda_3)$. It is apparent that the result of classification for $\boldsymbol{x}$ positioned outside information granules associates with the higher value of entropy.

Another option is to consider some other aggregation operators as studied in fuzzy sets. An alternative worth pursuing is to introduce a relation-based composition of $\boldsymbol{\omega}$ and R yielding $\boldsymbol{\zeta}$ such that

$$\hat{\xi} = f(W\hat{\omega}) \tag{15.24}$$

Where W is a matrix transforming original class membership grades. f is a nonlinear function resulting in the values in [0, 1]. The intent is to further optimize classification results by minimizing a classification error as wl as minimizing the entropy of the results. These requirements give rise to a two-objective optimization problem carried out with respect to the transformation matrix W.

15.3 Fuzzy Cognitive Maps

Fuzzy cognitive maps are representative examples of graph-oriented fuzzy models [6]. The objective is to model a problem by representing it through a collection of concepts and relationships among them. The map models the dynamics of the activation levels of the concepts observed over time and is formally represented as a directed graph, see Fig. 15.7. For detailed studies refer to [2–4, 7, 9, 11]

The concepts are nodes of the graph. The directed edges describe linkages between the concepts. The strength of linkages (weight) is quantified in the range $[-1, 1]$ The positive values associated with the edge state that a certain node positively (excites) another one. The higher the weight is, the stronger the relationship between the corresponding nodes becomes. Negative value of the connection states that one node

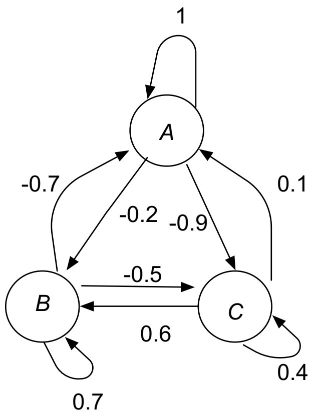


Fig. 15.7 Example of a fuzzy cognitive map with three concepts

is negatively impacted (inhibited) by another one; the values close to -1 indicate a high level of inhibition. Formally, the map is described in the following way, refer again to Fig. 15.7.

$$x(k+1) = f(x(k), W) = f(W\mathbf{x}(k)) \tag{15.25}$$

In terms of individual nodes of the map we have

$$x_i(k+1) = f(\sum_{j=1}^{c} w_{ij}x_i(k)) \tag{15.26}$$

$i = 1, 2, \ldots, c$. Here $\boldsymbol{x}(k) = [x_1(k)\, x_2(k) \ldots x_c(k)]$ is a vector of activations of the nodes in the k-th time instant, $W = [w_{ij}]$ stands for a matrix of linkages (associations). c denotes the number of nodes (concepts). The function f is a nonlinear mapping from $\boldsymbol{R}$ to [0, 1]. The commonly used functions include [6]

$f(u) = \mathbb{H}(u) = \begin{cases} 1\text{if } u \geq 0 \\ 0, \text{otherwise} \end{cases}$ (step function, Heaviside)
$f(u) = \tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$
$f(u) = 1/(1 + e^{-u}))$ (sigmoid).

As an illustration, consider a map with three nodes, Fig. 15.8.
The weight matrix W has the following entries

$$W = \begin{bmatrix} 0.0 & 0.8 & 0.7 \\ 0.0 & 0.0 & 0.5 \\ 0.0 & -0.7 & 0.0 \end{bmatrix}$$

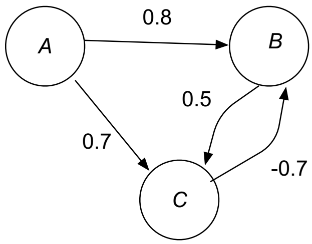


Fig. 15.8 Example of a fuzzy cognitive map with three nodes

By starting from some initial state $\boldsymbol{x}(0)$ and iterating the formula across discrete time moments, we have

$$\begin{aligned} \boldsymbol{x}(1) &= f(W\boldsymbol{x}(0)) \\ \boldsymbol{x}(2) &= f(W\boldsymbol{x}(1)) \\ &\ldots \\ \boldsymbol{x}(k+1) &= f(W\boldsymbol{x}(k)) \end{aligned} \tag{15.27}$$

The iterative process either converges or oscillates; the dynamics of the map predominantly depends upon the values of the weight matrix W.

Example 2 For the fuzzy cognitive map shown in Fig. 15.8, determine the vectors of activation in consecutive time moments when $\boldsymbol{x}(0) = [1\ 0\ 0]$. The nonlinear function is a step function.

Starting with $\boldsymbol{x}(0)$, we calculate

$\boldsymbol{x}(1) = [f(0)\, f(-0.7)\, f(0.5)] = [1\ 0\ 1]$
$\boldsymbol{x}(2) = [f(0)\, f(0.1)\, f(0.7)] = [1\ 1\ 1]$
$\boldsymbol{x}(3) = [f(0)\, f(0.1)\, f(1.2)] = [1\ 1\ 1]$
$\boldsymbol{x}(4) = [1\ 1\ 1]$
and the map converges to [1 1 1].

Fuzzy cognitive maps function at the abstract level of concepts. Real-world data are not concepts themselves so they have to be transformed (elevated) to the level of concepts. An overall framework of processing and design of the maps is presented in Fig. 15.9.

15.3.1 The Design of the Map

The design of the map involves three main phases: encoding, learning of the weights, and decoding.

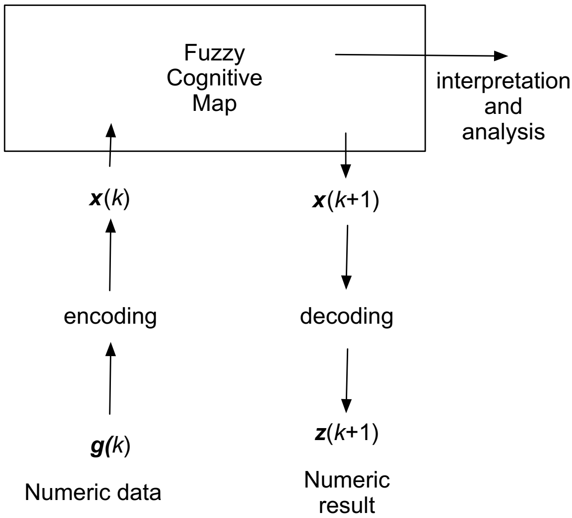


Fig. 15.9 Design, analysis and interpretation of fuzzy cognitive maps

Encoding

Maps function at the level of concepts that are formalized as information granules. Real-world phenomena generate numeric data. The essential design step is to transform data in the form of information granules. This process is realised through an encoding process. To elaborate on the details, let us consider the data coming in the form of multivariable time series where each variable is observed in discrete time moments, see Fig. 15.10.

Denote these data as n-dimensional vectors $\boldsymbol{g}(k)$, $k = 1, 2, \ldots, N$ which are to be used in the design of the map. Also there are some relationships among the variables that are present in the data and those have to be incorporated in the design of the fuzzy cognitive map.

Two ways of encoding are considered:

(i) formation of composite concepts built for all variables. In a formal way, the encoding mechanism, is sought as the mapping. $\boldsymbol{R}^n \to [0, 1]^c, \boldsymbol{g}(k) \to \boldsymbol{target}(k)$ where $\boldsymbol{target}(k)$ is a c-dimensional vector (concept) located in the c-dimensional hypercube. Referring to the map, each coordinate of $\boldsymbol{target}(k)$ is associated with the node of the map.
The data $\boldsymbol{g}(k)$ are clustered with Fuzzy C-Means being used as a vehicle of information granulation. $\boldsymbol{target}(k)$s are vectors of membership grades (activation levels of the nodes of the map) which are the results of transformation of $\boldsymbol{g}(k)$ with the use of the prototypes produced by the FCM algorithm.

(ii) encoding individual variables. The transformation is regarded as nonlinear (and linear mapping, in particular) from $\boldsymbol{R}$ to the [0, 1] interval.

Given n dimensional data, one has a series of mappings each for the corresponding coordinates of the variable. In this way one has $target_i = \Phi_i(g_i(k))$ where Φ_i is a

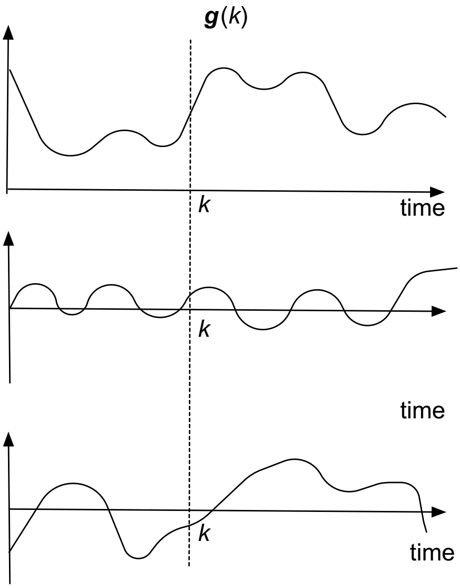


Fig. 15.10 3-dimensional data describing the system's variable in discrete time moment (k)

mapping for the individual variable. We assume that the mappings are monotonically increasing functions; some illustrative examples are included in Fig. 15.11. Note that in this encoding the map has n nodes.

Learning

The learning (which is about the determination of the matrix of weights W) is carried out by considering $\boldsymbol{g}(k)$s transformed through encoding and arranged in a collection of pairs (***target***(1), ***target*** (2)) (***target*** (2), ***target*** (3)) … (***target*** (k), $\boldsymbol{x}(k + 1)$), …, (***target*** ($N - 1$), ***target*** (N)).

Two modes of learning are considered here.

Hebbian learning [5] The learning is completed in an unsupervised way by determining the changes to the values of W, Δw_{ij} that are made proportional to the correlation between $target_i(k)$ and $target_j(k)$

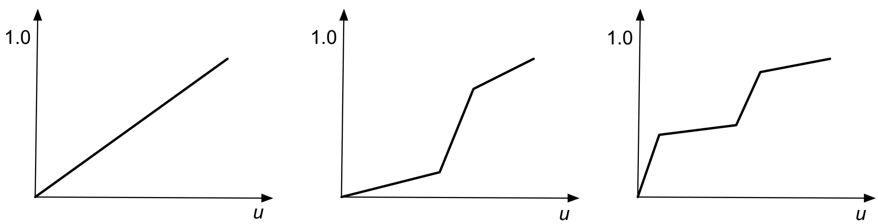


Fig. 15.11 Examples of nonlinear and nonlinear transformations from $\boldsymbol{R}$ to [0, 1]

$$\Delta w_{ij}(k) = \xi target_i(k) target_j(k) \tag{15.28}$$

and then using the data we carry out iterations

$$w_{ij}(iter + 1) = w_{ij}(iter) + \Delta w_{ij} \tag{15.29}$$

The new value of the weight w_{ij} is obtained by adding the increment $\Delta w_{ij}(k)$, $\xi > 0$. This learning is referred to as correlation-based learning.

Gradient-based learning [10] This learning is carried out in a supervised mode. We require that W is formed in such a way so that the predicted state $\boldsymbol{x}(k + 1)$ is made as close as possible to $\boldsymbol{target}(k + 1)$ for all $k = 1, 2, \ldots, N$. The minimized performance index Q is taken as the following sum of squared errors

$$Q = \sum_{k=1}^{N-1} ||\hat{\boldsymbol{x}}(k+1) - \boldsymbol{target}(k+1)||^2 \tag{15.30}$$

where

$$\hat{x}(k+1) = f(W\mathbf{target}(k))$$

The minimization is completed by modifying the values of the weights. The gradient-based minimization is realized in a way so that the results are truncated to the $[-1, 1]$ interval.

Decoding the Results Formed by the Fuzzy Cognitive Map

The transformation of the activation vector to the numeric values of the variables present in the original data is completed through the decoding process producing the vector of numeric outcomes.

Depending upon the encoding scheme, we consider the following decoding formulas

For encoding (i)

$$\hat{z}(k+1) = \sum_{i=1}^{c} v_i \hat{x}_i(k+1) \tag{15.31}$$

For encoding (ii)

$$\hat{z}_i(k+1) = \Phi^{-1}(\hat{x}_i(k+1)) \tag{15.32}$$

$i = 1, 2, \ldots, n$. An essence of the overall processing scheme is presented in Fig. 15.12.

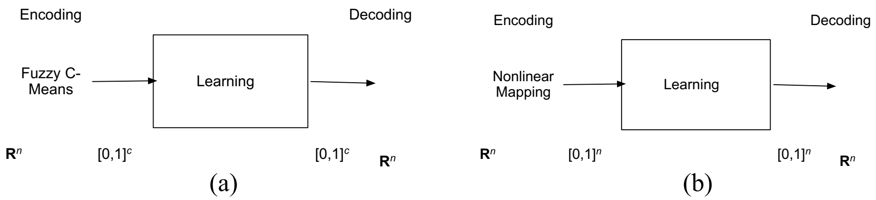


Fig. 15.12 Design of fuzzy cognitive maps: **a** use of clustering, **b** use of one-dimensional linear or nonlinear mappings

Interpretation of the Relationships Revealed by the Map

Fuzzy cognitive maps quantify the relationships among the concepts and the concepts themselves.

The concepts are inherently implied by the prototypes produced by the FCM algorithm. As such they carry well-formulated semantics. Consider a two -dimensional case $n = 2$ and $c = 3$ clusters (concepts), Fig. 15.13. By projecting the prototypes on the coordinates and ordering them, the semantics of the concepts is

The weights can be further interpreted by retaining the weights of significant magnitude $|w_{ij}| > 1$ where 1 is a certain threshold. The thresholding facilitates interpretation of the map by pointing at the main relationships among the concepts. Instead of thresholding applied to the weights, one can modify the learning scheme by augmenting the performance index by the regularization term

$$Q = \sum_{k=1}^{N-1} ||\hat{\boldsymbol{x}}(k+1) - \mathbf{target}(k+1)||^2 + \beta \sum_{i,j=1}^{c} h(|w_{ij}|) \tag{15.33}$$

where $h(u)$ is an entropy function which penalizes the values of u that are close to ½.

Interpretation of Fuzzy Cognitive Maps

As the map is fully described by the weight matrix, we can assess an impact of relationships present in the map at a global level as illustrated in Fig. 15.14.

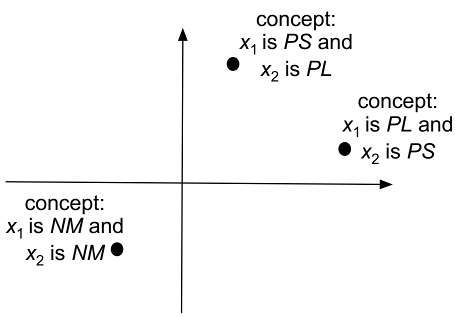


Fig. 15.13 Concepts (formed by the clusters) in a two-dimensional space

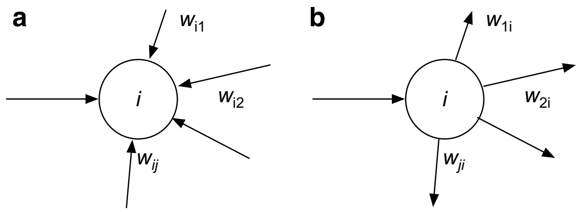


Fig. 15.14 Quantifying impact of the *i*-th concept: **a** the *i*-th concept impacted by other concepts, and **b** impact generated by the *i*-th concept

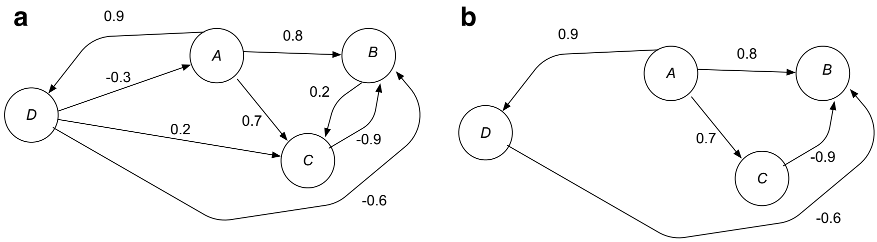


Fig. 15.15 From fuzzy cognitive map (**a**) to its core structure (**b**)

The impact of the concepts on the *i*-th one is quantified by taking the sum $\sum_{j=1}^{c} w_{ij}$. Likewise the impact the *i*-th node (concept) exhibits on all remaining concepts is quantified by the sum $\sum_{i=1}^{c} w_{ij}$

To facilitate the interpretation of the map, we remove the weakest weights, viz those whose absolute values are below the threshold λ. This pruning leaves a core structure of the map, see Fig. 15.15.

15.4 Conclusions

The granular constructs discussed here are representative examples of architectures demonstrating the use of information granules and fuzzy sets. It becomes apparent that information granularity of the obtained results can serve as a sound quantification of the quality of the models and confidence one can associate with the particular result produced by the model.

The design process shows that there is a systematic way of moving from data to more abstract and concise representative—information granules, which then play a pivotal role as a building blocks over which an overall model is being formed.

Problems

1. Consider the approximation of the fuzzy number $A = (f, g)$ by a triangular fuzzy number $A\tilde{}(x; L_1, L_2)$ where L_1 and L_2 are linear segments being the result of this

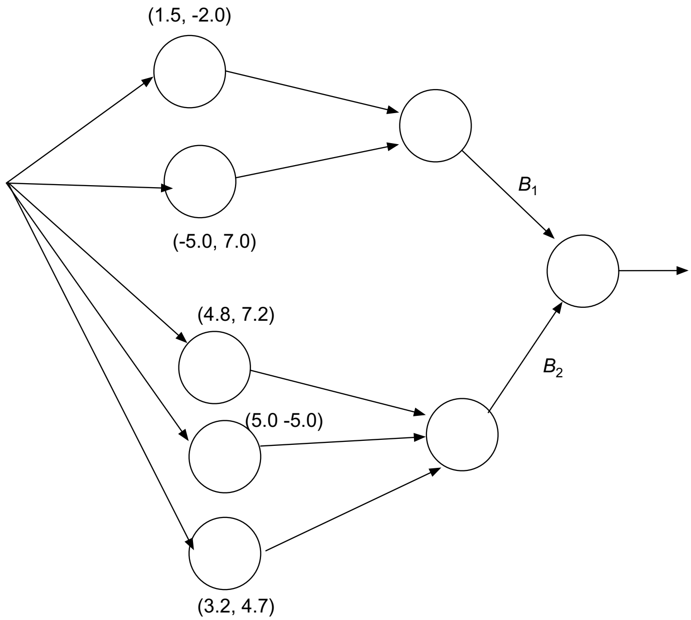


Fig. 15.16 Example of granular fuzzy model with two contexts

approximation. Augment this approximation by developing an interval triangular fuzzy number.

2. For some $\boldsymbol{x}$, determine the output of the model shown in Fig. 15.16 where there are two contexts described by triangular fuzzy sets $B_1(y; -5, 4, 7)$ and $B_2(y; 4, 7, 12)$.
3. Granular classifier in a two-dimensional feature space is based on three information granules whose geometry is built on the Tchebyshev distance. Recall that this distance is described as
$||x - v_i|| = \max_{j=1,2}\left|\left|x_j - v_{ij}\right|\right|$.

   (i) plot the information granules
   (ii) show the region in the feature space characterized by the highest value of entropy

4. Formulate the optimization problem (15.33) and discuss a way of determining its minimum.
5. Consider the multivariable time series appliances energy prediction at Machine Learning Repository https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction
Discuss a design of the fuzzy cognitive map.
6. The fuzzy cognitive map has two concepts and is characterized by the following weight matrix

$$W = \begin{bmatrix} -0.2 & 0.9 & 1.0 \\ 0.5 & 0.8 & 0.4 \\ 0.6 & -1.0 & 0.7 \end{bmatrix}$$

Analyze its dynamics for several initial condition $\boldsymbol{x}(0)$: [1 0 0 0], [0 1 0 0], [0 0 1 0], and [0 0 0 1]. In the analysis use a sigmoidal function in each node of the map.

References

1. Adhikari, S.K., Sing, J.K., Basu, D.K., Nasipuri, M.: Conditional spatial fuzzy C-means clustering algorithm for segmentation of MRI images. Appl. Soft Comput. **34**, 758–769 (2015)
2. Aguilar, J.: Different dynamic causal relationship approaches for cognitive maps. Appl. Soft Comput. **13**, 271–282 (2013)
3. Georgopoulosa, V., Malandrakia, G., Stylios, C.: A fuzzy cognitive map approach to differential diagnosis of specific language impairment. Artif. Intell. Med. **29**, 261–278 (2003)
4. Gonzalez, J.L., Aguilar, L.T., Castillo, O.: A cognitive map and fuzzy inference engine model for online design and self-fine-tuning of fuzzy logic controllers. Int. J. Intell. Syst. **24**, 1134–1173 (2009)
5. Hebb, D.O.: The Organization of Behavior. J Wiley, N. York (1949)
6. Kosko, B.: Fuzzy cognitive maps. Int. J. Man Mach. Stud. **24**, 65–75 (1986)
7. Papageorgiou, E.I., Froelich, W.: Application of evolutionary fuzzy cognitive maps for prediction of pulmonary infections. IEEE Trans. Inf Technol. Biomed. **16**, 143–149 (2012)
8. Pedrycz, W.: Conditional fuzzy clustering in the design of radial basis function neural networks. IEEE Trans. Neural Netw. **9**, 601–612 (1989)
9. Stach, W., Kurgan, L.A., Pedrycz, W., Reformat, M.: Genetic learning of fuzzy cognitive maps. Fuzzy Sets Syst. **153**, 371–401 (2005)
10. Stach, W., Kurgan, L.A., Pedrycz, W.: Numerical and linguistic prediction of time series with the use of fuzzy cognitive maps. IEEE Trans. Fuzzy Syst. **16**, 61–72 (2008)
11. Tchupo, D.E., Kim, H.H., Macht, G.A.: Fuzzy cognitive maps (FCMs) for the analysis of team communication. Appl. Ergon. **83** (2020)
12. Xia, S., Liu, Y., Ding, X., Wang, G., Luo, Y.: Granular ball computing classifiers for efficient, scalable and robust learning. Inf. Sci. **483**, 136–152 (2019)

Appendix A
Optimization

Gradient-Based Optimization Methods

Given is a differentiable function f: $\boldsymbol{R}^n \rightarrow \boldsymbol{R}$. There are several main categories of the gradient-based optimization methods (Chong and Zak 2013; Antoniou and Lu 2007) that are used to find a minimum of $f(\boldsymbol{x})$, $\boldsymbol{x}_0 = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$. Those are iterative algorithm where the sequence leading to the solution $\boldsymbol{x}_0$ is initialized from some random location $\boldsymbol{x}(0)$. ∇f denotes the gradient of f

$$\Delta f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \ldots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

H is a Hessian matrix of second derivatives (assuming that the function is twice differentiable) of f

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3

$$H = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Gradient-based method

It is the generic technique when the solution is computed with the used of the following scheme

$$\boldsymbol{x}(iter+1) = \boldsymbol{x}(iter) - \alpha f(\boldsymbol{x}(iter))$$

$\alpha > 0$ is a learning coefficient (rate).

Steep descent method

In this technique, the learning rate (step size) $\alpha(iter)$ used in the following learning scheme

$$\boldsymbol{x}(iter+1) = \boldsymbol{x}(iter) - \alpha(iter) f(\boldsymbol{x}(iter))$$

is optimized and coming as a result of the following minimization

$$\alpha(iter)_{opt} = \arg\ Min_{\alpha>0}[f(\boldsymbol{x}(iter) - \alpha f(\boldsymbol{x}(iter))]$$

The above learning schemes use only knowledge contained in the first derivative of f.

Newton method

The learning here includes knowledge about the second order derivative conveyed by the Hessian matrix

$$\boldsymbol{x}(iter+1) = \boldsymbol{x}(iter) - H^{-1} f(\boldsymbol{x}(iter))$$

Gradient-Based Optimization With Constraints

The minimization problem of $f(\boldsymbol{x})$ with constraints $g(\boldsymbol{x})$

$$\min_x f(\boldsymbol{x})$$
$$\text{subject to}$$
$$g(\boldsymbol{x}) = 0$$

is transformed into the minimization problem without constraint by introducing a Lagrange multiplier λ so that this reduces the original problem to the minimization problem without constraints

$$V(\boldsymbol{x}) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x}).$$

The necessary condition for the minimum of f results from the set of equations

$$\nabla_{\boldsymbol{x}} V = \boldsymbol{0} \quad \text{and} \quad \frac{\partial V}{\partial \lambda} = 0$$

Stochastic Optimization

The method of stochastic optimization ADAM (Kingma and Ba 2014) aimed at the minimization of $f(\boldsymbol{x})$ realizes the updates of the entries of $\boldsymbol{x}$ completed in the following iterative form ($\boldsymbol{m}$ and $\boldsymbol{v}$ have been initialized)

$$\begin{aligned}
&\boldsymbol{m}(iter+1) = b_1\boldsymbol{m}(iter) - (1-b_1)\boldsymbol{g}(iter)\\
&\hat{\boldsymbol{m}}(iter+1) = \boldsymbol{m}(iter)/(1-\beta_1^{iter})\\
&\boldsymbol{v}(iter+1) = b_2\boldsymbol{v}(iter) - (1-b_2)\boldsymbol{g}^2(iter)\\
&\hat{\boldsymbol{v}}(iter+1) = \boldsymbol{v}(iter)/(1-\beta_2^{\text{iter}})\\
&\boldsymbol{x}(iter+1) = \boldsymbol{x}(iter) - a\hat{\boldsymbol{m}}(iter+1)/\sqrt{\hat{\boldsymbol{v}}(iter+1)+\int}
\end{aligned}$$

$\alpha > 0$ while b_1 and b_2 are some additional parameters. In the above formula, ε is a small positive value that prevents division by zero while $\boldsymbol{g}^2$ stands for the element-wise square of the gradient vector $\boldsymbol{g}, \boldsymbol{g} = \nabla f$.

Typically, α=0.1; b_1 and b_2 are set as 0.9 and 0.999, respectively and $\varepsilon = 10\text{e}{-8}$.

Particle Swarm Optimization

Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) is a commonly used population-based optimization technique. The elements (particles) forming the population interact among themselves. In the search process each particle moves based on three components: its current velocity, its own experience (cognitive components) -the best position achieved so far and the best position produced by the entire swarm (social component). A quality of solution is assessed by a fitness function. The search is carried out in an n-dimensional search space. The generic formula guiding the trajectory of a particle located in the search space is described as follows

$$\begin{aligned}
\boldsymbol{v}(iter+1) = \xi\boldsymbol{v}(iter) + c_1 r\,(\boldsymbol{local_best} - \boldsymbol{x}(iter))\\
+ c_2\boldsymbol{g}(\boldsymbol{global_best} - \boldsymbol{x}(iter))
\end{aligned}$$

The symbol ·indicates that the multiplication is carried out coordinate-wise (Hadamard product). In other words, we have

$$v_i(iter + 1) = \xi v_i(iter) + c_1 r_i(local\_best_i - x_i(iter)) \\ + c_2 g_i(global\_best_i - x_i(iter))$$

$i = 1, 2, \ldots, n$. $\boldsymbol{r}$ and $\boldsymbol{g}$ are vectors of random numbers coming from the uniform distribution over [0, 1], ξ is an inertial weight, c_1 stands for the cognitive factor while c_2 is a social factor.

Once the velocity has been updated, a new position of the particle is determined as follows

$$\boldsymbol{x}(iter + 1) = \boldsymbol{x}(iter) + \boldsymbol{v}(iter + 1)$$

Eventual clipping of $\boldsymbol{x}(iter + 1)$ is done so that the particle stays within the search space.

An overall PSO scheme is composed of the following steps

Form an initial population (swarm)
Evaluate fitness values of individual particles
Determine the individual best position of each particle
Determine the global best position in the swarm
Update velocity and position of each particle
Check satisfaction of the stopping criterion, if the criterion has been met, stop else continue.

PSO is a certain metaheuristic which can be easily applied to a vast array of problems. The key problem-oriented aspects of the method involve

(i) representation of the search space. It is directly linked with the problem at hand and a suitable representation becomes crucial to the performance of the search process.
(ii) size of population N associates with the dimensionality of the search space; higher values of n imply higher values of N.
(iii) selection of values of the parameters (inertial weight, cognitive and social factors). There are a number of proposals present in the literature however they are problem-oriented. One can refer to (Clerc and Kennedy 2002) for the detailed discussion on this subject.

References

1. Antoniou, A., Lu, W.S.: Practical Optimization. Algorithms and Engineering Applications. Springer, N. York (2007).
2. Chong, E.K.P., Zak, S.K.: An Introduction to Optimization. Wiley, Hoboken, NJ (2013).
3. Clerc, M., Kennedy, J.: The particle swarm—explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evolut. Comput., **6** 58–73 (2002)

4. Kennedy, J., Eberhart, R.: Particle swarm optimization. In Proceedings of ICNN’95—International Conference on Neural Networks, 1942–1948 (1995).
5. Kingma, D.P., Ba, J.L.: A method for stochastic optimization, arXiv: 1412.6980 (2014, December).

Appendix B
Final Examinations

<u>Each problem is worth **10** points (90 points in total)</u>

Please note that some of the problems are of design nature; there could be several possible alternative solutions. In such cases, state clearly assumptions under which you are formulating the problem and developing its solution. Offer a motivation behind your design approach.

10 points

1. Develop an FCM clustering algorithm in which the Euclidean distance is replaced by its weighted version expressed in the following form

$$||\boldsymbol{x}-\boldsymbol{v}_i||2^{=}(\boldsymbol{x}-\boldsymbol{v}_i)^{\mathrm{T}}W(\boldsymbol{x}-\boldsymbol{v}_i)=\sum_{j=1}^{n}\sum_{j=1}^{n}(x_j-v_{ij})w_{jl}(x_l-v_{il})$$

W. Pedrycz, *An Introduction to Computing with Fuzzy Sets*, Intelligent Systems Reference Library 190, https://doi.org/10.1007/978-3-030-52800-3

where $W = [w_{jl}], j, l = 1, 2, \ldots, n$ is a certain weight matrix given in advance and $\boldsymbol{x}$ and $\boldsymbol{v}_i$ are n-dimensional vectors of real numbers.

10 points

2. Approximate the following membership function A by a normal triangular fuzzy number with the following membership function

$$A(x) = \begin{cases} \cos(x), \text{ifx} \in \left[0, \frac{\pi}{2}\right] \\ 0, \text{otherwise} \end{cases}$$

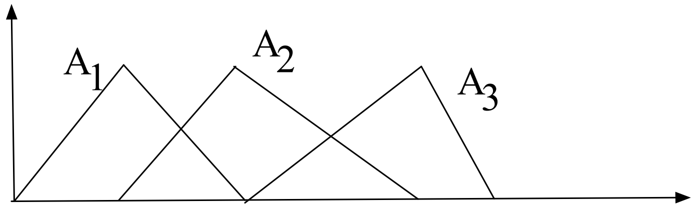
Hint: in this approximation, consider a quadratic approximation criterion to be minimized.

10 points

3. Given is a collection of reference fuzzy sets $A_1, A_2, \ldots, A_c$:••• $\boldsymbol{R} \to [0,1]$. For unknown fuzzy set X defined in the same space as the reference fuzzy sets, provided are possibility values, $\lambda_1 = \text{Poss}(X, A_1), \ldots\ \lambda_2 = \text{Poss}(X, A_2), \ldots, \lambda_c = \text{Poss}(X, A_c)$. Recall that the possibility is computed in the following form

$$Poss(X, A_i) = sup_x[min(X(x), A_i(x))]$$

Given A_i and the possibility values λ_i, determine X.
Plot the resulting X for the reference fuzzy sets shown below.



Determine fuzzy set X when

(i) $\lambda_1 = 0.2, \lambda_2 = 1, \lambda_3 = 1,$

and

(ii) $\lambda_1 = 0.7, \lambda_2 = 0.3, \lambda_3 = 0$

10 points

4. Given are two fuzzy sets A and B with the following membership functions defined in $\boldsymbol{R}$

(i) determine an intersection of A and B using the product of their membership functions. What is its support and height?
(ii) What is the possibility value Poss(A, B) and the possibility of their complements, $\bar{A}(x) = 1 - A$(x) and $\bar{B}(x) = 1 - B$(x)

10 points

5. A concept "*low* inflation" is provided by 7 experts who gave the following membership functions

[1.0 0.9 0.7 0.5 0.4 0.2 0.0 0.0]]
[1.0 1.0 0.5 0.3 0.2 0.2 0.1 0.0]
[1.0 0.4 0.2 0.1 0.1 0.0 0.0 0.0]
[1.0 0.8 0.7 0.6 0.5 0.4 0.2 0.2]
[1.0 0.6 0.5 0.5 0.4 0.0 0.0 0.0]
[1.0 1.0 0.8 0.4 0.3 0.2 0.1 0.1]
[1.0 0.8 0.7 0.4 0.4 0.1 0.1 0.0]

Using the principle of justifiable granularity, construct and draw an interval-valued fuzzy set. Interpret the obtained result.
10 points

6. An electric circuit is composed of two resistors R_1 and R_2 and a voltage source of 5 V. The resistance of these resistors are described by triangular membership functions, namely $R_1(x; 1, 3, 5)$ and $R_2(x; 0.3, 1, 3)$. What is the voltage drop V_1 on the first resistor? Determine the corresponding membership function of V_1 and plot it.

10 points

7. A classifier dealing with N data belonging to r classes, w_1, w_2, …, w_c can be realized through data clustering realized with the use of the FCM algorithm. Each cluster is composed of data belonging to it with the highest degrees of membership. Say, for the i-th cluster we have

$$\boldsymbol{X}_i = \left\{\boldsymbol{x}_k | i = \arg\max_{j=1,2,\ldots,c} u_{ik}\right\}$$

Typically, $\boldsymbol{X}_i$ consists of a mixture of data belonging to several classes with one dominant class (most frequently present class), say w_j. The pattern belonging to other classes are those incorrectly classified. The classification error is computed as a ratio of all misclassified data across all the clusters to N. Run an experiment (using the FCM algorithm) for the following datahttps://archive.ics.uci.edu/ml/datasets/banknote+authenticationSplit your data into its training (70%) and testing (30%) part. The fuzzification coefficient is set to 2, m=2.0.Report the obtained values of the classification error for the training and testing data. Discuss the "optimal" number of clusters—how does it compare to the number of classes.**10 points**

8. The principle of justifiable granularity produces an information granule by maximizing a performance index described as a product of coverage and specificity, $cov{*}sp$. A generalized version of the performance index comes in its parametrized version such as $cov{*}sp^{\alpha}$ where α assumes some nonnegative values. Sketch a possible relationship between coverage and specificity obtained for the maximized $cov{*}sp^{\alpha}$ in the coverage-specificity coordinates when varying values of α from zero to some maximal value α_{max}.

10 points

9. The rules are given in the following format

$$-\text{if } x \text{ is } A_i \text{ then } y \text{ is } \bar{y}_i$$

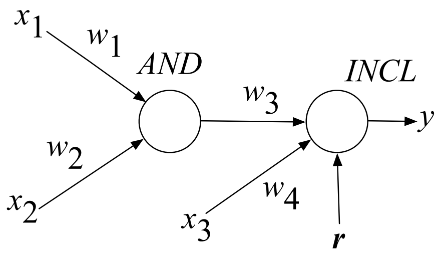
$i = 1, 2, \ldots, 4$. A_i are triangular fuzzy sets defined in $\boldsymbol{R}$ with an overlap of ½ between two consecutive fuzzy sets. The corresponding modal values $modal_i$ and the constant functions $\bar{y}_i$ are listed below

$Modal_i$	$\bar{y}_i$
2.0	5.0
3.5	2.0
7.0	−1.0
9.0	6.0

(i) Determine and draw the input-output relationship produced by this rule-based system.
(ii) What is the value of the output of this rule-based model for $x = 7.5$.
(iii) If the input is an information granule as an interval [2.6•••3.7], what is the corresponding output.

Each problem is worth **10** points (100 points in total)*Please note that some of the problems are of design nature; there could be several possible alternative solutions. In such cases, state clearly assumptions under which you are formulating the problem and developing its solution. Offer a motivation behind your design approach.*

1. The fuzzy neural network is shown below.



Elaborate on the learning scheme involving an optimization of the weights and the reference point $\boldsymbol{r}$. The data used for learning come in the form of input-output pairs ($\boldsymbol{x}(k)$, *target*(k)), $k = 1, 2, \ldots, N$. Assume that the t-norm and t-conorm are selected the product and the probabilistic sum, respectively.

2. Approximate the following membership function A

$$A(x) = \begin{cases} \cos(x), x \in [\frac{-\pi}{2} \frac{\pi}{2}] \\ 0, \text{ otherwise} \end{cases}$$

by a normal triangular fuzzy number.

3. Given is a fuzzy set $A = [0.7\ 0.2\ 0.9\ 1.0\ 0.5\ 0.6]$ and a result of its union with some fuzzy set B is $A \cup B = [0.9\ 0.4\ 0.9\ 0.5\ 0.6\ 0.7]$. The union is realized by the maximum operator. Determine the membership function B. Is the result unique; if not, discuss possible solutions.
4. In a rule-based model, the rules are given in the following form

$$-if\ \boldsymbol{x} \text{ is } A_i \text{ then } y \text{ is } w_i$$

$i = 1, 2, \ldots c$. Consider that clustering (Fuzzy C-Means) has been completed in the input space resulting in a collection of fuzzy sets (conditions of rules) $A_1, A_2, \ldots, A_c$. Discuss how to determine (optimize) conclusions w_i. The data are in the form of input-output pairs ($\boldsymbol{x}(k)$, *target*(k)), $k = 1, 2, \ldots, N$.

5. Fuzzy sets A and B are two triangular fuzzy numbers $A(x; -1, 0, 2)$ and $B(x; 1, 3, 5)$. What is the probability of $A \cap B$, $P(A \cap B)$, where the intersection operator is modeled by the minimum operation where $p(x) = \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{x^2}{2\sigma^2})$
6. The single input—single output relationship is described as $3\sin(x)$ for x in $[0,2\pi]$ and zero otherwise. Suggest a collection of rules modeling this relationship

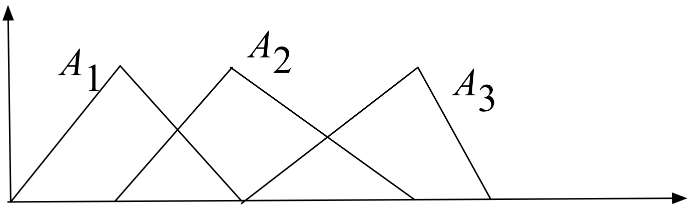
$$-\text{if } x \text{ is } A_i \text{ then} y \text{ is } a_{0i} + a_{1i}x$$

Elaborate on the number of required rules and a distribution of fuzzy sets in the space of input variable.

7. The FCM algorithm can be regarded as a classifier. Using the code developed for Assignment #3, use it as a classifier for the iris dataset http://archive.ics.uci.edu/ml/datasets/Iris/. Experiment with various values of the number of clusters c, $c = 2, 3, \ldots 6$ and the fuzzification coefficient m. Show results for the training and testing set; use the split of the data as 60% and 40%, respectively. Interpret the obtained results.
8. Given is a collection of reference fuzzy sets $A_1, A_2, \ldots, A_c$:••• $\boldsymbol{R} \rightarrow [0,1]$. For unknown fuzzy set X defined in the same space as the reference fuzzy sets, provided are possibility values, $\lambda_1 = \text{Poss}(X, A_1)$, $\lambda_2 = \text{Poss}(X, A_2)$, …, $\lambda_c = \text{Poss}(X, A_c)$. Recall that the possibility is computed in the following form

$$Poss(X, A_i) = sup_x[min(X(x), A_i(x))]$$

Given are fuzzy sets A_i and the possibility values λ_i, $i = 1, 2, \ldots, c$. Determine X. Plot the resulting X for the reference fuzzy sets shown below.



Determine X when

(i) $\lambda_1 = 0.2, \lambda_2 = 1, \lambda_3 = 1, \lambda_4 = 0.5,$

and

(ii) $\lambda_1 = 0.7, \lambda_2 = 0.3, \lambda_3 = 0, \lambda_4 = 0$

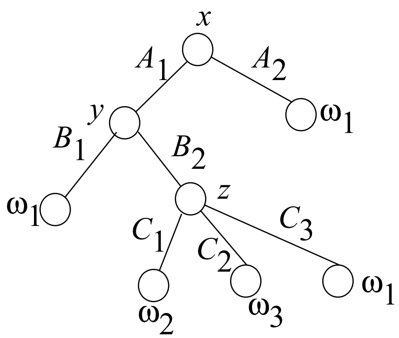
Plot the obtained X.

9. The decoding (defuzzification) method using the center of gravity method is governed by the following minimization criterion

$$Q = \int_X A(x)(x - x_0)^2 dx$$

Determine x_0 that minimizes Q.

10. The fuzzy decision tree is shown below



The membership functions defined in spaces $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$, namely A_1, A_2, B_1, B_2, C_1, C_2, and C_3 are given as Gaussian membership functions

$$A_1(x) = exp(-(x-2)^2/1.5) \quad A_2(x) = exp(-(x-3)^2/2.0)$$
$$B_1(y) = exp(-(y-2.4)^2/1.9) \quad B_2(y) = exp(-(y+1)^2/1.5)$$
$$C_1(z) = exp(-(z+3)^2/2) \quad C_2(z) = exp(-(z+1)^2/3)C_3(z) = exp(-(z-1)^2/1.5)$$

(i) Determine class membership of the input data $\boldsymbol{x}$ =[x_0, y_0, z_0].

(ii) Show conditions for which the entropy of the result is lower than some threshold h_0.

In other words, given are x_0 and y_0, what are the values of z for which the entropy is lower than h_0.