

Tu-Bao Ho
Zhi-Hua Zhou (Eds.)

LNAI 5351

PRICAI 2009

Trends in Artificial

10th Pacific Rim International Conference on Artificial Intelligence
Hanoi, Vietnam, December 14–18, 2009
Proceedings

Lecture Notes in Artificial Intelligence 5351

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Tu-Bao Ho Zhi-Hua Zhou (Eds.)

PRICAI 2008: Trends in Artificial Intelligence

10th Pacific Rim International Conference
on Artificial Intelligence
Hanoi, Vietnam, December 15-19, 2008
Proceedings



Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada

Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Tu-Bao Ho

Japan Advanced Institute of Science and Technology

Asahidai 1-1, Nomi 923-12292, Japan

E-mail: bao@jaist.ac.jp

Zhi-Hua Zhou

Nanjing University, Department of Computer Science & Technology

22 Hankou Road, Nanjing, 210093, China

E-mail: zhoush@nju.edu.cn

Library of Congress Control Number: 2008939435

CR Subject Classification (1998): I.2, F.1

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743

ISBN-10 3-540-89196-X Springer Berlin Heidelberg New York

ISBN-13 978-3-540-89196-3 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2008

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12558841 06/3180 5 4 3 2 1 0

Preface

The Pacific Rim International Conference on Artificial Intelligence (PRICAI) is one of the preeminent international conferences on artificial intelligence (AI). PRICAI 2008 (<http://www.jaist.ac.jp/PRICAI-08/>) was the tenth in this series of biennial international conferences highlighting the most significant contributions to the field of AI. The conference was held during December 15–19, 2008, in the beautiful city Hanoi, the capital of Vietnam.

As in previous years this year's technical program saw very high standards in both the submission and paper review process, resulting in an exciting program that reflects the great variety and depth of modern AI research. This year's contributions covered all traditional areas of AI, including AI foundations, knowledge representation, knowledge acquisition and ontologies, evolutionary computation, etc., as well as various exciting and innovative applications of AI to many different areas. There was particular emphasis in the areas of machine learning and data mining, intelligent agents, language and speech processing, information retrieval and extraction.

The technical papers in this volume were selected from a record of 234 submissions after a rigorous review process. Each submission was reviewed by at least three members and one Vice-Chair of the Program Committee. Decisions were reached following discussions among the reviewers of each paper, Vice Chairs and Chairs of the Program Committee, and finalized in a highly selective process that balanced many aspects of a paper, including the significance of the contribution and originality, technical quality and clarity of contributions, and relevance to the conference. Finally, we accepted 49 long papers and 33 regular papers for oral presentation (35%), and 32 short papers for poster presentation (13.6%) at the conference. In addition, we were honored to have one keynote and three invited speeches by leading researchers in the field. The PRICAI 2008 program also included four workshops (“Pacific Rim Knowledge Acquisition Workshop,” “Empirical Methods for Asian Language Processing,” “Soft Computing for Knowledge Technology”, and “Knowledge, Language, and Learning in Bioinformatics”) and three tutorials (“Empirical Methods for Artificial Intelligence,” “Agent and Data Mining: The Synergy to Empower Intelligent Information Processing Systems,” and “Writing and Presenting Scientific Papers”).

PRICAI 2008 would not have been possible without the work of many people and organizations. We wish to express our gratitude to:

- The Conference Chairs: Hiroshi Motoda and Bach Hung Khang
- The PRICAI Steering Committee
- The keynote and invited speakers: Paul Cohen, Hendrik Blockeel, An-Hai Doan and Yuji Matsumoto
- The Organizing Chairs: Nguyen Ngoc Binh, Pham Hoang Luong, and Luong Chi Mai as well as their staff and volunteer students
- The Workshop Chairs: Duc Nghia Pham and Takashi Washio
- The Tutorial Chairs: Aditya K. Ghose and Cao Hoang Tru

- The Industrial Chair: Minh B. Do
- The Publication Chair: Saori Kawasaki
- The Registration Chairs: Ngo Cao Son and Saori Kawasaki
- Web masters: Pham Ngoc Khanh and Tran Dang Hung
- The team of Microsoft's conference management tool for its support
- Springer for its continuing support in publishing the proceedings
- The workshop organizers: Debbie Rechards and Byeong Ho Kang; Akira Shimazu, Luong Chi Mai and Manabu Okumura; Hung Son Nguyen and Van Nam Huynh; Kenji Satou and Masanori Arita
- The tutorial presenters: Paul Cohen, Longbin Cao and Chengqi Zhang, and Tu Bao Ho
- The Program Committee members and Vice Chairs: Naoki Abe, Hung Bui, Peter Flach, Eibe Frank, Randy Goebel, Achim Hoffmann, James Kwok, Doheon Lee, Riichiro Mizoguchi, Wee Keong Ng, Satoshi Tojo, Abdul Sattar, Qiang Yang, Chengqi Zhang, and Limsoon Wong
- The external reviewers

We greatly appreciate the financial support from various sponsors: the Vietnamese Academy of Science and Technology (VAST), Ministry of Science and Technology of Vietnam (MoST), Hanoi University of Technology (HUT), Vietnam National Universities, Air Force Office of the Scientific Research/Asian Office of Aerospace Research and Development (AFOSR/AOARD).

Last but not the least we would like to thank all authors of the submitted papers, and all conference attendees for their contribution and participation. Without them we would not have had this conference.

December 2008

Tu-Bao Ho
Zhi-Hua Zhou

Organization

Conference Chairs

Hiroshi Motoda	AOARD/Osaka University, Japan
Bach Hung Khang	Vietnamese Academy of Science and Technology, Vietnam

Program Committee Chairs

Tu Bao Ho	Japan Advanced Institute of Science and Technology, Japan
Zhi-Hua Zhou	Nanjing University, China

Organizing Chairs

Tu Bao Ho	Japan Advanced Institute of Science and Technology, Japan
Nguyen Ngoc Binh	College of Technology, VNU-HN, Vietnam
Pham Hoang Luong	Hanoi University of Technology, Vietnam
Luong Chi Mai	Vietnamese Academy of Science and Technology, Vietnam

Workshop Chairs

Duc Nghia Pham	ICTA/Griffith University, Australia
Takashi Washio	Osaka University, Japan

Tutorial Chairs

Aditya K. Ghose	University of Wollongong, Australia
Tru Hoang Cao	Ho Chi Minh City University of Technology, Vietnam

Industrial Chair

Minh B. Do	Palo Alto Research Center, USA
------------	--------------------------------

Publication Chair

Saori Kawasaki	Japan Advanced Institute of Science and Technology, Japan
----------------	--

Registration Chairs

Ngo Cao Son	Vietnamese Academy of Science and Technology, Vietnam
Saori Kawasaki	Japan Advanced Institute of Science and Technology, Japan

PRICAI Steering Committee

Wai K. Yeap (Chair)	Inst. for Information Technology Research
Abdul Sattar (Secretary-Treasurer)	Griffith University
Tru Hoang Cao	Ho Chi Minh City University of Technology
Randy Goebel	University of Alberta
Mitsuru Ishizuka	University of Tokyo
Fangzhen Lin	Hong Kong Univ. of Science and Technology
Hiroshi Motoda	AOARD/Osaka University
Hideyuki Nakashima	Future University - Hakodate
Nancy Reed	University of Hawaii
R. Sadananda	Asian Institute of Technology
Mohd. Sapiyan	University of Malaya
Geoff Webb	Monash University
Chengqi Zhang	University of Technology Sydney

PRICAI 2008 Program Committee

Chairs

Ho Tu Bao	Japan Advanced Inst. of Science and Technology, Japan
Zhi-Hua Zhou	Nanjing University, China

Vice-Chairs

Naoki Abe	IBM T.J. Watson Research Center, USA
Hung H. Bui	SRI International
Peter Flach	University of Bristol, UK
Eibe Frank	University of Waikato, New Zealand
Randy Goebel	University of Alberta, Canada
Achim Hoffmann	University of New South Wales, Australia
James Kwok	Hong Kong University of Science and Technology, Hong Kong, China
Doheon Lee	Korean Advanced Institute of Science and Technology, Korea

Riichiro Mizoguchi	Osaka University, Japan
Wee Keong Ng	Nanyang Technological University, Singapore
Satoshi Tojo	Japan Advanced Institute of Science and Technology, Japan
Abdul Sattar	Griffith University, Australia
Qiang Yang	Hong Kong University of Science and Technology, Hong Kong, China
Chengqi Zhang	University of Technology, Sydney, Australia
Limsoon Wong	National University of Singapore, Singapore

Members

David Albrecht	Monash University, Australia
Rajendra Akerkar	Technomathematics Research Foundation, India
Aijun An	York University, UK
Mike Barley	University of Auckland, New Zealand
Laxmidhar Behera	Indian Institute of Technology, Kanpur, India
Hendrik Blockeel	Katholieke Universiteit Leuven, Belgium
Jean-Francois Boulicaut	Institut National des Sciences Appliquees de Lyon, France
Longbing Cao	University of Technology, Sydney, Australia
Tru Hoang Cao	Ho Chi Minh City University of Technology, Vietnam
Nicholas Cercone	Dalhousie University, Canada
Phoebe Y-P Chen	Deakin University, Australia
Songcan Chen	Nanjing University of Aeronautics and Astronautics , China
Zheng Chen	Microsoft Research Asia
Shu-Ching Chen	Florida International University, USA
David W-L Cheung	The University of Hong Kong, Hong Kong, China
Sung-Bae Cho	Yonsei University, Korea
Paul Compton	University of New South Wales, Australia
Jirapun Daengdej	Assumption University, Thailand
Raedt Luc De	Katholieke Universiteit Leuven, Belgium
Minh B. Do	Palo Alto Research Center, USA
AnHai Doan	University of Wisconsin-Madison, USA
Anh Duc Duong	Ho Chi Minh City University of Natural Sciences, Vietnam
Fazel Famili	National Research Council
Wei Fan	IBM T.J. Watson Research Center, USA
Hamido Fujita	Iwate Prefectural University, Japan
Peter A. Flach	University of Bristol, UK

X Organization

Joao Gama	University of Porto, Portugal
Dragan Gamberger	Rudjer Boskovic Institute, Croatia
Yang Gao	Nanjing University, China
Sharon XiaoYing Gao	Victoria University of Wellington, New Zealand
Fosca Giannotti	ISTI, CNR di Pisa, Italy
Xin Geng	Deakin University, Australia
Peter Haddawy	Asian Institute of Technology, Thailand
James Harland	RMIT University, Australia
Kazuo Hashimoto	Tohoku University, Japan
Takashi Hashimoto	Japan Advanced Institute of Science and Technology, Japan
Koichi Hori	University of Tokyo, Japan
Wynne Hsu	National University of Singapore, Singapore
Xiangji Huang	York University, UK
Joshua Huang	The University of Hong Kong, Hong Kong, China
Shell Ying Huang	Nanyang Technological University, Singapore
Van Nam Huynh	Japan Advanced Institute of Science and Technology, Japan
Mitsuru Ikeda	Japan Advanced Institute of Science and Technology, Japan
Rolly Intan	Petra Christian University, Indonesia
Sanjay Jain	National University of Singapore, Singapore
Zhi Jin	Chinese Academy of Science, China
Geun Sik Jo	Inha University, Korea
Jeffrey Junfeng	Google Inc.
Ken Kaneiwa	National Institute of Informatics
Byeong Ho Kang	University of Tasmania, Australia
Hiroyuki Kawano	Kyoto University, Japan
Masatsugu Kidode	Nara Institute of Science and Technology, Japan
Boonserm Kijsirikul	Chulalongkorn University, Thailand
Masahiro Kimura	Ryukoku University, Japan
Yasuhiko Kitamura	Kwansei Gakuin University, Japan
Peep Kungas	SOA Trader, Ltd.
Susumu Kunifugi	Japan Advanced Institute of Science and Technology, Japan
Satoshi Kurihara	Osaka University, Japan
Wai Lam	Chinese University of Hong Kong, Hong Kong, China
Nada Lavrac	Jozef Stefan Institute, Slovenia
Wee Sun Lee	National University of Singapore, Singapore
Tze Yun Leong	National University of Singapore, Singapore
Xue Li	The University of Queensland, Australia

Chun-Hung Li	Hong Kong Baptist University, Hong Kong, China
Zhoujun Li	Beihang University, China
Gerard Ligozat	University Paris-Sud, France
Ee-Peng Lim	Nanyang Technological University, Singapore
Jiming Liu	Hong Kong Baptist University, Hong Kong, China
Huan Liu	Arizona State University, USA
Xudong Luo	University of Southampton, UK
Jixin Ma	University of Greenwich, UK
Michael J. Maher	University of New South Wales, Australia
Donato Malerba	University of Bari, Italy
Yuji Matsumoto	Nara Institute of Science and Technology, Japan
Gordon McCalla	University of Saskatchewan, Canada
Chris Messon	Massey University, New Zealand
Antonija Mitrovic	University of Canterbury, New Zealand
Yohei Murakami	National Inst. of Information and Com. Technology
Le Minh Nguyen	Japan Advanced Institute of Science and Technology, Japan
Hung Son Nguyen	University of Warsaw, Poland
Ngoc Binh Nguyen	College of Technology
Thanh Thuy Nguyen	Hanoi University of Technology, Vietnam
Trong Dung Nguyen	Vietnam Academy of Science and Technology, Vietnam
Takashi Okada	Kwansei Gakuin University, Japan
Manabu Okumura	Tokyo Institute of Technology, Japan
Jeffrey Junfeng Pan	Google Inc.
Jeng-Shyang Pan	National Kaohsiung University, Taiwan, ROC
Hyeyoung Park	Kyungpook National University, Korea
Seong-Bae Park	Kyungpook National University, Korea
Jose M Pena	Universidad Politecnica de Madrid, Spain
Xuan Hieu Phan	Tohoku University, Japan
Tho Hoan Pham	Hanoi National University of Education, Vietnam
Fred Popowich	Simon Fraser University, Canada
Arun K. Pujari	University of Hyderabad, India
Hiok Chai Quek	Nanyang Technological University, Singapore
Joel Quinqueton	University Monpellier 3, France
Anca Luminita Ralescu	University of Cincinnati, USA
Debbie Richard	Macquarie University, Australia
Pat Riddle	University of Auckland, New Zealand
Fabio Roli	University of Cagliari, Italy
Kazumi Saito	University of Shizuoka, Japan
Kenji Satou	Kanazawa University, Japan

Rudy Setiono	National University of Singapore, Singapore
Yidong Shen	Chinese Academy of Science, China
Daming Shi	Nanyang Technological University, Singapore
Akira Shimazu	Japan Advanced Institute of Science and Technology, Japan
Kyoaki Shirai	Japan Advanced Institute of Science and Technology, Japan
Kate Smith-Miles	Deakin University, Australia
Von-Wun Soo	National Tsing-Hua University, China
Eiichiro Sumita	Adv. Telecommunications Research Inst. International, Japan
Wing Kin Sung	National University of Singapore, Singapore
Hideaki Takeda	National Institute of Informatics
An Hwee Tan	Nanyang Technological University, Singapore
Chew Lim Tan	National University of Singapore, Singapore
David Taniar	Monash University, Australia
Takao Terano	Tokyo Institute of Technology, Japan
Alexandre Termier	Université Joseph Fourier, France
Thanaruk Theeramunkong	Thammasat University, Thailand
John Thornton	Griffith University, Australia
Kai Ming Ting	Monash University, Australia
Cao Son Tran	New Mexico State University, USA
Shusaku Tsumoto	Shimane University, Japan
Toby Walsh	University of New South Wales, Australia
Lipo Wang	Nanyang Technological University, Singapore
Hui Wang	University of Ulster, UK
Takashi Washio	Osaka University, Japan
Ian Watson	University of Auckland, New Zealand
Graham Williams	Australian National University, Australia
Wayne Wobcke	University of New South Wales, Australia
Mingrui Wu	Max Planck Institute for Biological Cybernetics, Germany
Xintao Wu	University of North Carolina at Charlotte, USA
Hui Xiong	Rutgers University, USA
Xiangyang Xue	Fudan University, China
Seiji Yamada	National Institute of Informatics, Japan
Ying Yang	Monash University, Australia
Hyun Seung Yang	Korea Advanced Institute of Science and Technology, Korea
Yiyu Yao	University of Regina, Canada
Roland H.C Yap	National University of Singapore, Singapore
Jieping Ye	Arizona State University, USA

Dit-Yan Yeung	Hong Kong University of Science and Technology, Hong Kong, China
Jeffrey Xu Yu	Chinese University of Hong Kong, Hong Kong, China
Kai Yu	NEC Labs America, USA
Lei Yu	Binghamton University, USA
Philip Yu	University of Illinois at Chicago, USA
Shipeng Yu	Siemens Medical Solutions USA, USA
Pong Chi Yuen	Hong Kong Baptist University, Hong Kong, China
Yifeng Zeng	Aalborg University, Denmark
Hongbin Zha	Peking University, China
Dongmo Zhang	University of Western Sydney, Australia
Shichao Zhang	University of Technology, Sydney, Australia
Zili Zhang	Deakin University, Australia
Benyu Zhang	Microsoft Research Asia
Bo Zhang	Tsinghua University, China
Changshui Zhang	Tsinghua University, China
Daoqiang Zhang	Nanjing University of Aeronautics and Astronautics, China
Junping Zhang	Fudan University, China
Liqing Zhang	Shanghai Jiaotong University, China
Min-Ling Zhang	Hohai University, China
Byoung-Tak Zhang	Seoul National University, Korea
Du Zhang	California State University at Sacramento, USA
Jian Zhang	Carnegie Mellon University, USA
Weixiong Zhang	Washington University in St. Louis, USA
Yanqing Zhang	Georgia State University, USA
Zhongfei (Mark) Zhang	Binghamton University, USA
Alice Zheng	Carnegie Mellon University, USA
Ning Zhong	Maebashi Institute of Technology, Japan
Aoying Zhou	Fudan University, China
Shuigeng Zhou	Fudan University, China
Yan Zhou	University of South Alabama, USA
Jerry Zhu	University of Wisconsin-Madison, USA
Xinquan Zhu	Florida Atlantic University, USA
Jean-Daniel Zucker	LIP6 Paris, France

PRICAI 2008 External Reviewers

Annalisa Appice	Anja Austermann	Sebastian Brand	Michelangelo Ceci
Mafruz Ashrafi	Ivan Bindoff	Yundong Cai	Feng Chen

XIV Organization

Yann Chevaleyre	Rolly Intan	Motohiro Mase	Yasufumi Takama
Hai Leong Chieu	Xing Jiang	Makoto Nakamura	Li Tao
Anne Cregan	Tom Johnsten	Nina Narodytska	Milan Tofiloski
Luca Didaci	Daisuke Katagami	Nguyen Canh Hao	Gervase Tuxworth
Kurt Driessens	Yoshikiyo Kato	Masayuki Okabe	Siba Udgata
Naoki Fukuta	Yiping Ke	Jialin Pan	Mike Qiang Wang
Giorgio Fumera	Sankalp Khanna	Rong Pan	Qin Wang
Masabumi Furuhata	Yasuo Kudo	Maxim Roy	Wenchen
Guido Governatori	Yan Li	Dou Shen	Shanshan Wu
Baohua Gu	C. Likitvivatanavong	Zhiyong Shen	Pengyi Yang
Vijaya K. Gunta	Bo Liu	Zhongmin Shi	A. Zimmermann
Corneliu Henegar	Tony Fei Liu	G.R. Simari	
Martin Henz	Yang Liu	Alok Singh	
Shoji Hirano	Gian Luca Marcialis	Jun Sun	

Table of Contents

Keynotes

What Shall We Do Next? The Challenges of AI Midway through Its First Century.....	1
<i>Paul R. Cohen</i>	
Exposing the Causal Structure of Processes by Learning CP-Logic Programs	2
<i>Hendrik Blockeel</i>	
Building Structured Web Community Portals Via Extraction, Integration, and Mass Collaboration	3
<i>An-Hai Doan</i>	
Large Scale Corpus Analysis and Recent Applications	4
<i>Yuji Matsumoto</i>	
On the Computability and Complexity Issues of Extended RDF	5
<i>Anastasia Analyti, Grigoris Antoniou, Carlos Viegas Damásio, and Gerd Wagner</i>	
Toward Formalizing Common-Sense Psychology: An Analysis of the False-Belief Task	17
<i>Konstantine Arkoudas and Selmer Bringsjord</i>	
Computing Stable Skeletons with Particle Filters.....	30
<i>Xiang Bai, Xingwei Yang, Longin Jan Latecki, Yanbo Xu, and Wenyu Liu</i>	
Using Semantic Web Technologies for the Assessment of Open Questions	42
<i>Dagoberto Castellanos-Nieves, Jesualdo Tomas Fernandez-Breis, Rafael Valencia-Garcia, Carlos Cruz, Maria Paz Prendes-Espinosa, and Rodrigo Martinez-Bejar</i>	
Quantifying Commitment	54
<i>Timothy William Cleaver and Abdul Sattar</i>	
Temporal Data Mining for Educational Applications.....	66
<i>Carole R. Beal and Paul R. Cohen</i>	
Dual Properties of the Relative Belief of Singletons	78
<i>Fabio Cuzzolin</i>	

Alternative Formulations of the Theory of Evidence Based on Basic Plausibility and Commonality Assignments	91
<i>Fabio Cuzzolin</i>	
Non-negative Sparse Principal Component Analysis for Multidimensional Constrained Optimization	103
<i>Thanh D.X. Duong and Vu N. Duong</i>	
Sentence Compression by Removing Recursive Structure from Parse Tree	115
<i>Seiji Egawa, Yoshihide Kato, and Shigeki Matsubara</i>	
An ATP of a Relational Proof System for Order of Magnitude Reasoning with Negligibility, Non-closeness and Distance	128
<i>Joanna Golińska-Pilarek, Angel Mora, and Emilio Muñoz-Velasco</i>	
A Heuristic Data Reduction Approach for Associative Classification Rule Hiding	140
<i>Juggapong Natwichai, Xingzhi Sun, and Xue Li</i>	
Evolutionary Computation Using Interaction among Genetic Evolution, Individual Learning and Social Learning	152
<i>Takashi Hashimoto and Katsuhide Warashina</i>	
Behavior Learning Based on a Policy Gradient Method: Separation of Environmental Dynamics and State Values in Policies	164
<i>Seiji Ishihara and Harukazu Igarashi</i>	
Developing Evaluation Model of Topical Term for Document-Level Sentiment Classification	175
<i>Yi Hu, Wenjie Li, and Qin Lu</i>	
Learning to Identify Comparative Sentences in Chinese Text	187
<i>Xiaojiang Huang, Xiaojun Wan, Jianwu Yang, and Jianguo Xiao</i>	
Efficient Exhaustive Generation of Functional Programs Using Monte-Carlo Search with Iterative Deepening	199
<i>Susumu Katayama</i>	
Identification of Subject Shareness for Korean-English Machine Translation	211
<i>Kye-Sung Kim, Seong-Bae Park, Hyun-Je Song, Se-Young Park, and Sang-Jo Lee</i>	
Agent for Predicting Online Auction Closing Price in a Simulated Auction Environment	223
<i>Deborah Lim, Patricia Anthony, and Chong Mun Ho</i>	
Feature Selection Using Mutual Information: An Experimental Study	235
<i>Huawen Liu, Lei Liu, and Huijie Zhang</i>	

Finding Orthogonal Arrays Using Satisfiability Checkers and Symmetry Breaking Constraints	247
<i>Feifei Ma and Jian Zhang</i>	
Statistical Model for Japanese Abbreviations	260
<i>Norifumi Murayama and Manabu Okumura</i>	
A Novel Heuristic Algorithm for Privacy Preserving of Associative Classification	273
<i>Nattapon Harnsamut and Juggapong Natwichai</i>	
Time-Frequency Analysis of Vietnamese Speech Inspired on Chirp Auditory Selectivity	284
<i>Ha Nguyen and Luis Weruaga</i>	
Meta-level Control of Multiagent Learning in Dynamic Repeated Resource Sharing Problems	296
<i>Itsuki Noda and Masayuki Ohta</i>	
Ontology-Based Natural Query Retrieval Using Conceptual Graphs	309
<i>Tho Thanh Quan and Siu Cheung Hui</i>	
Optimal Multi-issue Negotiation in Open and Dynamic Environments	321
<i>Fenghui Ren and Minjie Zhang</i>	
The Density-Based Agglomerative Information Bottleneck	333
<i>Yongli Ren, Yangdong Ye, and Gang Li</i>	
State-Based Regression with Sensing and Knowledge	345
<i>Richard Scherl, Cao Son Tran, and Chitta Baral</i>	
Some Results on the Completeness of Approximation Based Reasoning	358
<i>Cao Son Tran and Enrico Pontelli</i>	
KT and S4 Satisfiability in a Constraint Logic Environment	370
<i>Lynn Stevenson, Katarina Britz, and Tertia Hörne</i>	
Clustering with Feature Order Preferences	382
<i>Jun Sun, Wenbo Zhao, Jiangwei Xue, Zhiyong Shen, and Yidong Shen</i>	
Distributed Memory Bounded Path Search Algorithms for Pervasive Computing Environments	394
<i>Anoj Ramasamy Sundar and Colin Keng-Yan Tan</i>	
Using Cost Distributions to Guide Weight Decay in Local Search for SAT	405
<i>John Thornton and Duc Nghia Pham</i>	

XVIII Table of Contents

Fault Resolution in Case-Based Reasoning	417
<i>Ha Manh Tran and Jürgen Schönwälder</i>	
Constrained Sequence Classification for Lexical Disambiguation	430
<i>Tran The Truyen, Dinh Q. Phung, and Svetha Venkatesh</i>	
Map Building by Sequential Estimation of Inter-feature Distances	442
<i>Atsushi Ueta, Takehisa Yairi, Hirofumi Kanazaki, and Kazuo Machida</i>	
Document-Based HITS Model for Multi-document Summarization	454
<i>Xiaojun Wan</i>	
External Force for Active Contours: Gradient Vector Convolution	466
<i>Yuanquan Wang and Yunde Jia</i>	
Representation = Grounded Information	473
<i>Mary-Anne Williams</i>	
Learning from the Past with Experiment Databases	485
<i>Joaquin Vanschoren, Bernhard Pfahringer, and Geoffrey Holmes</i>	
An Argumentation Framework Based on Conditional Priorities	497
<i>Quoc Bao Vo</i>	
Knowledge Supervised Text Classification with No Labeled Documents	509
<i>Congle Zhang, Gui-Rong Xue, and Yong Yu</i>	
Constrained Local Regularized Transducer for Multi-Component Category Classification	521
<i>Congle Zhang and Yong Yu</i>	
Low Resolution Gait Recognition with High Frequency Super Resolution	533
<i>Junping Zhang, Yuan Cheng, and Changyou Chen</i>	
NIIA: Nonparametric Iterative Imputation Algorithm.....	544
<i>Shichao Zhang, Zhi Jin, and Xiaofeng Zhu</i>	
Mining Multidimensional Data through Element Oriented Analysis	556
<i>Yihao Zhang, Mehmet A. Orgun, Weiqiang Lin, and Rohan Baxter</i>	
Evolutionary Feature Selections for Face Detection System	568
<i>Zalhan Mohd Zin, Marzuki Khalid, and Rubiyah Yusof</i>	
A Probabilistic Approach to the Interpretation of Spoken Utterances ...	581
<i>Ingrid Zukerman, Enes Makalic, Michael Niemann, and Sarah George</i>	

Regular Papers

Towards Autonomous Robot Operation: Path Map Generation of an Unknown Area by a New Trapezoidal Approximation Method Using a Self Guided Vehicle and Shortest Path Calculation by a Proposed SRS Algorithm.....	593
<i>K. Ahmed, M.S. Munir, A.S.M Shihavuddin, M.A. Hoque, and K.K. Islam</i>	
Exploring Combinations of Ontological Features and Keywords for Text Retrieval	603
<i>Tru H. Cao, Khanh C. Le, and Vuong M. Ngo</i>	
Instance Management Problems in the Role Model of Hozo	614
<i>Kouji Kozaki, Satoshi Endo, and Riichiro Mizoguchi</i>	
Advancing Topic Ontology Learning through Term Extraction	626
<i>Blaž Fortuna, Nada Lavrač, and Paola Velardi</i>	
Handling Unknown and Imprecise Attribute Values in Propositional Rule Learning: A Feature-Based Approach	636
<i>Dragan Gamberger, Nada Lavrač, and Johannes Fürnkranz</i>	
Fuzzy Knowledge Discovery from Time Series Data for Events Prediction	646
<i>Ehsanolah Gholami and Mohammadreza Matash Borujerdi</i>	
Evolution of Migration Behavior with Multi-agent Simulation.....	658
<i>Hideki Hashizume, Atsuko Mutoh, Shohei Kato, and Hidenori Itoh</i>	
Constraint Relaxation Approach for Over-Constrained Agent Interaction	668
<i>Mohd Fadzil Hassan and Dave Robertson</i>	
Structure Extraction from Presentation Slide Information	678
<i>Tessai Hayama, Hidetsugu Nanba, and Susumu Kunifugi</i>	
Combining Local and Global Resources for Constructing an Error-Minimized Opinion Word Dictionary	688
<i>Linh Hoang, Jung-Tae Lee, Young-In Song, and Hae-Chang Rim</i>	
An Improvement of PAA for Dimensionality Reduction in Large Time Series Databases	698
<i>Nguyen Quoc Viet Hung and Duong Tuan Anh</i>	
Stability Margin for Linear Systems with Fuzzy Parametric Uncertainty	708
<i>Petr Hušek</i>	

An Imperative Account of Actions	718
<i>Victor Jauregui and Son Bao Pham</i>	
Natural Language Interface Construction Using Semantic Grammars	728
<i>Anh Kim Nguyen and Huong Thanh Le</i>	
Exploiting the Role of Named Entities in Query-Oriented Document Summarization	740
<i>Wenjie Li, Furu Wei, Ouyang You, Qin Lu, and Yanxiang He</i>	
A Probabilistic Model for Understanding Composite Spoken Descriptions	750
<i>Enes Makalic, Ingrid Zukerman, Michael Niemann, and Daniel Schmidt</i>	
Fuzzy Communication Reaching Consensus under Acyclic Condition	760
<i>Takashi Matsuhisa</i>	
Probabilistic Nogood Store as a Heuristic	768
<i>Andrei Missine and William S. Havens</i>	
Semantic Filtering for DDL-Based Service Composition	778
<i>Wenjia Niu, Zhongzhi Shi, Peng Cao, Hui Peng, and Liang Chang</i>	
Prediction of Protein Functions from Protein Interaction Networks: A Naïve Bayes Approach	788
<i>Cao D. Nguyen, Kathleen J. Gardiner, Duong Nguyen, and Krzysztof J. Cios</i>	
Multi-class Support Vector Machine Simplification	799
<i>DucDung Nguyen, Kazunori Matsumoto, Kazuo Hashimoto, Yasuhiro Takishima, Daichi Takatori, and Masahiro Terabe</i>	
A Syntactic-based Word Re-ordering for English-Vietnamese Statistical Machine Translation System	809
<i>Hong-Nhung Nguyen Thi and Dien Dinh</i>	
A Multi-modal Particle Filter Based Motorcycle Tracking System	819
<i>Phi-Vu Nguyen and Hoai-Bac Le</i>	
Bayesian Inference on Hidden Knowledge in High-Throughput Molecular Biology Data	829
<i>Viet-Anh Nguyen, Zdena Koukolíková-Nicola, Franco Bagnoli, and Pietro Lió</i>	
Personalized Search Using ODP-based User Profiles Created from User Bookmark	839
<i>Tetsuya Oishi, Yoshiaki Kambara, Tsunenori Mine, Ryuzo Hasegawa, Hiroshi Fujita, and Miyuki Koshimura</i>	

Domain-Driven Local Exceptional Pattern Mining for Detecting Stock Price Manipulation	849
<i>Yuming Ou, Longbing Cao, Chao Luo, and Chengqi Zhang</i>	
A Graph-Based Method for Combining Collaborative and Content-Based Filtering	859
<i>Nguyen Duy Phuong, Le Quang Thang, and Tu Minh Phuong</i>	
Hierarchical Differential Evolution for Parameter Estimation in Chemical Kinetics	870
<i>Yuan Shi and Xing Zhong</i>	
Differential Evolution Based on Improved Learning Strategy	880
<i>Yuan Shi, Zhen-zhong Lan, and Xiang-hu Feng</i>	
SalienceGraph: Visualizing Salience Dynamics of Written Discourse by Using Reference Probability and PLSA	890
<i>Shun Shiramatsu, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno</i>	
Learning Discriminative Sequence Models from Partially Labelled Data for Activity Recognition	903
<i>Tran The Truyen, Hung H. Bui, Dinh Q. Phung, and Svetha Venkatesh</i>	
Feature Selection for Clustering on High Dimensional Data	913
<i>Hong Zeng and Yiu-ming Cheung</i>	
Availability of Web Information for Intercultural Communication	923
<i>Takashi Yoshino, Kunikazu Fujii, and Tomohiro Shigenobu</i>	
Short Papers	
Mining Weighted Frequent Patterns in Incremental Databases	933
<i>Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young-Koo Lee</i>	
Revision of Spatial Information by Containment	939
<i>Omar Doukari, Robert Jeansoulin, and Eric Würbel</i>	
Joint Power Control and Subcarrier Allocation in MC - CDMA Systems - An Intelligent Search Approach	945
<i>Le Xuan Dung</i>	
Domain-Independent Error-Based Simulation for Error-Awareness and Its Preliminary Evaluation	951
<i>Tomoya Horiguchi and Tsukasa Hirashima</i>	

A Characterization of Sensitivity Communication Robots Based on Mood Transition	959
<i>Chika Itoh, Shohei Kato, and Hidenori Itoh</i>	
Recommendation Algorithm for Learning Materials That Maximizes Expected Test Scores	965
<i>Tomoharu Iwata, Tomoko Kojiri, Takeshi Yamada, and Toyohide Watanabe</i>	
A Hybrid Kansei Design Expert System Using Artificial Intelligence	971
<i>Jyun-Sing Chen, Kun-Chieh Wang, and Jung-Chin Liang</i>	
Solving the Contamination Minimization Problem on Networks for the Linear Threshold Model	977
<i>Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda</i>	
A Data-Driven Approach for Finding the Threshold Relevant to the Temporal Data Context of an Alarm of Interest	985
<i>Savo Kordic, Peng Lam, Jitian Xiao, and Huaizhong Li</i>	
Branch and Bound Algorithms to Solve Semiring Constraint Satisfaction Problems	991
<i>Louise Leenen and Aditya Ghose</i>	
Image Analysis of the Relationship between Changes of Cornea and Postmortem Interval	998
<i>Fang Liu, Shaohua Zhu, Yuxiao Fu, Fan Fan, Tianjiang Wang, and Songfeng Lu</i>	
Context-Based Term Frequency Assessment for Text Classification	1004
<i>Rey-Long Liu</i>	
Outlier Mining on Multiple Time Series Data in Stock Market	1010
<i>Chao Luo, Yanchang Zhao, Longbing Cao, Yuming Ou, and Li Liu</i>	
Generating Interactive Facial Expression of Communication Robots Using Simple Recurrent Network	1016
<i>Yuki Matsui, Masayoshi Kanoh, Shohei Kato, and Hidenori Itoh</i>	
Effects of Repair Support Agent for Accurate Multilingual Communication	1022
<i>Mai Miyabe, Takashi Yoshino, and Tomohiro Shigenobu</i>	
Towards Adapting XCS for Imbalance Problems	1028
<i>Thach Huy Nguyen, Sombut Foitong, Phaitoon Srinil, and Ouen Pinngern</i>	
Personalized Summarization Agent Using Non-negative Matrix Factorization	1034
<i>Sun Park</i>	

Interactive Knowledge Acquisition and Scenario Authoring	1039
<i>Debbie Richards</i>	
Reconstructing Hard Problems in a Human-Readable and Machine-Processable Way	1046
<i>Rolf Schwitter</i>	
Evolving Intrusion Detection Rules on Mobile Ad Hoc Networks	1053
<i>Sevil Sen and John A. Clark</i>	
On the Usefulness of Interactive Computer Game Logs for Agent Modelling	1059
<i>Matthew Sheehan and Ian Watson</i>	
An Empirical Study on the Effect of Different Similarity Measures on User-Based Collaborative Filtering Algorithms	1065
<i>Ashish Sureka and Pranav Prabhakar Mirajkar</i>	
Using Self-Organizing Maps with Learning Classifier System for Intrusion Detection	1071
<i>Kreangsak Tamee, Pornthep Rojanavasu, Sonchai Udomthanapong, and Ouen Pinngern</i>	
New Particle Swarm Optimization Algorithm for Solving Degree Constrained Minimum Spanning Tree Problem	1077
<i>Huynh Thi Thanh Binh and Truong Binh Nguyen</i>	
Continuous Pitch Contour as an Improvement Feature for Music Information Retrieval by Humming/Singing	1086
<i>Tri Nguyen Truong Duc, Minh Le Nhat, Ha Nguyen Duc Hoang, and Quan Vu Hai</i>	
Classification Using Improved Hybrid Wavelet Neural Networks	1092
<i>Nhu Khue Vuong, Yi Zhi Zhao, and Xiang Li</i>	
Online Classifier Considering the Importance of Attributes	1098
<i>Hiroaki Ueda, Yo Nasu, Yuki Mikura, and Kenichi Takahashi</i>	
An Improved Tabu Search Algorithm for 3D Protein Folding Problem	1104
<i>Xiaolong Zhang and Wen Cheng</i>	
Transferring Knowledge from Another Domain for Learning Action Models	1110
<i>Hankui Zhuo, Qiang Yang, Derek Hao Hu, and Lei Li</i>	
Texture and Target Orientation Estimation from Phase Congruency	1116
<i>Qingbo Yin, Liran Shen, and Jong Nam Kim</i>	

XXIV Table of Contents

Query Classification and Expansion for Translation Mining Via Search Engines	1121
<i>Jian-Min Yao, Jun Sun, Lei Guo, and Qiao-Ming Zhu</i>	
Author Index	1127

What Shall We Do Next?

The Challenges of AI Midway through Its First Century

Paul R. Cohen

University of Arizona, USA

Abstract. After half a century of productive work, let us pause to consider what to do next. Looking back we see that Turing's Test was a destination without a map, a goal without a methodology. We see three major, gradual retreats from AI's original goals—general intelligence, knowledge-based intelligence, and problem solving. We see the fragmentation of AI into sub-disciplines and growing uncertainty about who we are and what we want to accomplish. Yet I am optimistic: With an informed understanding of our past we can design and run large-scale, goal-directed research programs—as other organized sciences do—and if we do so with vision and discipline we may yet see Turing's Test passed—and our understanding of intelligence dramatically increased – before the end of our first century. This is already happening in several areas of AI, as I will illustrate in my talk.

Exposing the Causal Structure of Processes by Learning CP-Logic Programs

Hendrik Blockeel

K.U. Leuven, Belgium and Leiden University, the Netherlands

Abstract. Since the late nineties there has been an increased interested in probabilistic logic learning, an area within AI that combines machine learning with logic-based knowledge representation and uncertainty reasoning. Several different formalisms for combining first-order logic with probability reasoning have been proposed, and it has been studied how models in these formalisms can be automatically learned from data.

This talk starts with a brief introduction to probabilistic logic learning, after which we will focus on a relatively new formalism known as CP-logic. CP-logic stands for “causal probabilistic logic”. It is a knowledge representation formalism that allows us to write down rules that indicate that a certain combination of conditions may cause certain effects with a particular probability (e.g., tossing a coin may cause a result of heads or tails, each with 50% probability). Besides the fact that this formalism is interesting for knowledge representation in itself, it also offers interesting opportunities from the machine learning point of view. Indeed, given the semantics of these CP-logic programs, learning them from data amounts to extracting probabilistic causal influences from data. We will discuss recent research on learning CP-logic programs, including: algorithms for learning them; how they relate to graphical models; and applications of learning CP-logic programs.

Building Structured Web Community Portals Via Extraction, Integration, and Mass Collaboration

An-Hai Doan

University Wisconsin Madison, USA

Abstract. The World-Wide Web hosts numerous communities, each focusing on a particular topic. As such communities proliferate, so do efforts to build community portals. Most current portals are organized according to topic taxonomies. Recently, however, there has been a growing effort to build structured data portals (e.g., IMDB, Citeseer) that present a unified view of entities and relationships in the community. Such portals can prove extremely valuable in a wide range of domains. But how can we build them efficiently?

In this talk, I will present a new research vision that addresses this question. The goal is to develop a system that a small team (or ideally just one person) can quickly deploy to build an initial (but already useful) structured portal, then leverage the entire community in a mass collaboration fashion to improve and expand this portal. As such, the research agenda requires combining and extending research in information extraction, information integration, and Web 2.0 technologies, among others. This agenda is actively being pursued in the Cimple project, a joint effort between the University of Wisconsin and Yahoo Research. In the talk I will describe recent progress in Cimple, portal prototypes, lessons learned, and future directions. I will focus in particular on how Cimple raises interesting and novel challenges for both AI and database research. More information about Cimple can be found at www.cs.wisc.edu/~anhai/projects/cimple.

Large Scale Corpus Analysis and Recent Applications

Yuji Matsumoto

Nara Institute of Science and Technology, Japan

Abstract. Recent progress of corpus and machine learning-based natural language processing methodologies have made it possible to handle large scale corpus with a quite high accuracy. The speaker is now involved in a project for constructing a large scale contemporary Japanese balanced corpus, aiming at constructing automatic annotation tools on various levels of natural language analyses. I will first introduce our activities on corpus based natural language analyzers for word dependency parsing and anaphora resolution and annotated corpus management environment. Then, I will explain recent natural language applications such as sentiment/opinion mining and knowledge extraction from a large scale text data like Weblogs.

On the Computability and Complexity Issues of Extended RDF

Anastasia Analyti¹, Grigoris Antoniou^{1,2},
Carlos Viegas Damásio³, and Gerd Wagner⁴

¹ Institute of Computer Science, FORTH-ICS, Greece
{analyti,antoniou}@ics.forth.gr

² Department of Computer Science, University of Crete, Greece
³ CENTRIA, Departamento de Informatica, Faculdade de Ciencias e Tecnologia,
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
cd@di.fct.unl.pt

⁴ Inst. of Informatics, Brandenburg Univ. of Technology at Cottbus, Germany
G.Wagner@tu-cottbus.de

Abstract. ERDF stable model semantics is a recently proposed semantics for ERDF ontologies and a faithful extension of RDFS semantics on RDF graphs. In this paper, we elaborate on the computability and complexity issues of the ERDF stable model semantics. We show that decidability under this semantics cannot be achieved, unless ERDF ontologies of restricted syntax are considered. Therefore, we propose a slightly modified semantics for ERDF ontologies, called *ERDF #n-stable model semantics*. We show that entailment under this semantics is in general decidable and it also extends RDFS entailment. An equivalence statement between the two semantics and various complexity results are provided.

Keywords: Extended RDF ontologies, Semantic Web, negation, rules, complexity.

1 Introduction

Rules constitute the next layer over the ontology languages of the Semantic Web, allowing arbitrary interaction of variables in the head and body of the rules. Berners-Lee [3] identifies the following fundamental theoretical problems: negation and contradictions, open-world versus closed-world assumptions, and rule systems for the Semantic Web. In [1], the Semantic Web language RDFS [8] is extended to accommodate the two negations of Partial Logic [9], namely *weak negation* \sim (expressing negation-as-failure or non-truth) and *strong negation* \neg (expressing explicit negative information or falsity), as well as derivation rules. The new language is called *Extended RDF* (*ERDF*). In [1], the *stable model semantics* of ERDF ontologies is developed, based on Partial Logic, extending the model-theoretic semantics of RDFS [8].

ERDF enables the combination of closed-world (non-monotonic) and open-world (monotonic) reasoning, in the same framework, through the presence of

weak negation (in the body of the rules) and the new metaclasses *erdf:TotalClass* and *erdf:TotalProperty*, respectively. In particular, relating strong and weak negation at the interpretation level, ERDF distinguishes two categories of properties and classes. *Partial properties* are properties p that may have truth-value gaps, that is $p(x, y)$ is possibly neither true nor false. *Total properties* are properties p that satisfy *totalness*, that is $p(x, y)$ is either true or false. Partial and total classes c are defined similarly, by replacing $p(x, y)$ by *rdf:type*(x, c). ERDF also distinguishes between properties (and classes) that are completely represented in a knowledge base and those that are not. Clearly, in the case of a completely represented (*closed*) property p , entailment of $\sim p(x, y)$ allows to derive $\neg p(x, y)$, and the underlying *completeness assumption* has also been called *Closed-World Assumption (CWA)* in the AI literature.

Such a completeness assumption for *closing* a partial property p by default may be expressed in ERDF by means of the rule $\neg p(?x, ?y) \leftarrow \sim p(?x, ?y)$ and for a partial class c , by means of the rule $\neg \textit{rdf:type} (?x, c) \leftarrow \sim \textit{rdf:type} (?x, c)$. These derivation rules are called *default closure rules*. In the case of a total property p , default closure rules are not applicable. This is because, some of the considered interpretations will satisfy $p(x, y)$ and the rest $\neg p(x, y)$ ¹, preventing the preferential entailment of $\sim p(x, y)$. Thus, on total properties, an *Open-World Assumption (OWA)* applies. Similarly to first-order-logic, in order to infer negated statements about total properties, explicit negative information has to be supplied, along with ordinary (positive) information.

Intuitively, an ERDF ontology is the combination of (i) an ERDF graph G containing (implicitly existentially quantified) positive and negative information, and (ii) an ERDF program P containing derivation rules, with possibly all connectives $\sim, \neg, \supset, \wedge, \vee, \forall, \exists$ in the body of a rule, and strong negation \neg in the head of a rule.

Example 1. We want to select wines for a dinner such that for each adult guest that (we know that) likes wine, there is on the table exactly one wine that he/she likes. Further, we want guests who are neither adults nor children to be served *Coca-Cola*. Additionally, we want adult guests, for whom we do not know if they like wine, also to be served *Coca-Cola*. Assume that in contrast to a child, we cannot decide if guest is an adult or not. For this drink selection problem, we use the classes: (i) *ex:Guest*, whose instances are the persons that will be invited to the dinner, (ii) *ex:Wine*, whose instances are wines, (iii) *ex:SelectedWine* whose instances the wines *chosen* to be served, (iv) *ex:Adult*, whose instances are persons, 18 years of age or older, and (v) *ex:Child*, whose instances are persons, 10 years of age or younger. Additionally, we use the properties: (i) *ex:likes*(X, Y) indicating that *we know that* person X likes wine Y , and (ii) *ex:serveSoftDrink*(X, Y) indicating that person X will be served soft drink Y . An ERDF program P that describes this drink selection problem is the following^{2,3}:

```

id(?x, ?x) ← true.
rdf:type(?y, SelectedWine) ← rdf:type(?x, Guest), rdf:type(?x, Adult),

```

¹ On total properties p , the *Law of Excluded Middle* $p(x, y) \vee \neg p(x, y)$ applies.

² To improve readability, we ignore the example namespace *ex*:

³ Commas “,” in the body of the rules indicate conjunction \wedge .

$$\begin{aligned}
& \text{rdf:type(?y, Wine), likes(?x, ?y),} \\
& \forall?z \ (\text{rdf:type(?z, SelectedWine), } \sim\text{id}(\text{?y, ?z}) \supset \sim\text{likes}(\text{?x, ?z})). \\
\text{rdf:type(Adult, erdf:TotalClass)} & \leftarrow \text{true}. \\
\neg\text{rdf:type(?x, Child)} & \leftarrow \neg\text{rdf:type(?x, Child)}. \\
\text{serveSoftDrink(?x, Coca-Cola)} & \leftarrow \text{rdf:type(?x, Guest), } \neg\text{rdf:type(?x, Adult)}, \\
& \quad \neg\text{rdf:type(?x, Child)}. \\
\text{serveSoftDrink(?x, Coca-Cola)} & \leftarrow \text{rdf:type(?x, Guest), rdf:type(?x, Adult),} \\
& \quad \forall?y \ (\text{rdf:type(?y, Wine)} \supset \sim\text{likes}(\text{?x, ?y})).
\end{aligned}$$

Consider now the ERDF graph G , containing the factual information: $G = \{\text{rdf:type(Carlos, Guest), rdf:type(Gerd, Guest), rdf:type>Anne, Guest), rdf:type(Riesling, Wine), rdf:type(Retsina, Wine), likes(Gerd, Riesling), likes(Gerd, Retsina), likes(Carlos, Retsina), rdf:type(Gerd, Adult), rdf:type(Carlos, Adult)\}$.

Then, $O = \langle G, P \rangle$ is an ERDF ontology. Note that *Adult* is declared in P as total class⁴. Thus, on this class the OWA applies and case-based reasoning on the truth value of $\text{rdf:type}(Anne, Adult)$ is performed. On the other hand, *likes(X, Y)* is a partial property and *Child* is a partial class. In particular, on *Child* a CWA applies, expressed by a default closure rule. \square

In [1], it is shown that stable model entailment conservatively extends RDFS entailment from RDF graphs to ERDF ontologies. Unfortunately, satisfiability and entailment under the ERDF stable model semantics are in general undecidable. In this work, we further elaborate on the undecidability result of the ERDF stable model semantics. We show that decidability cannot be achieved under this semantics, unless ERDF ontologies of restricted syntax are considered. This is due to the fact that the RDF vocabulary is infinite. Therefore, to achieve decidability of reasoning in the general case, we propose a modified semantics, called *ERDF #n-stable model semantics* (for $n \in \mathbb{N}$). The new semantics also extends RDFS entailment from RDF graphs to ERDF ontologies. Moreover, if O is a simple ERDF ontology (i.e., the bodies of the rules of O contain only the logical factors \sim, \neg, \wedge) then query answering under the ERDF #n-stable model semantics (for $n \in \mathbb{N}$) reduces to query answering under the answer set semantics [7]. An equivalence statement between the ERDF stable and #n-stable model semantics is provided. Moreover, we provide complexity results for (i) the ERDF #n-stable model semantics on simple ERDF ontologies and objective ERDF ontologies (i.e., ERDF ontologies whose rules contain only the logical factors \neg, \wedge) and (ii) the ERDF stable model semantics on objective ERDF ontologies.

The rest of the paper is organized as follows: Section 2 reviews the stable model semantics of ERDF ontologies. In Section 3, we propose the #n-stable model semantics of ERDF ontologies that extends RDFS entailment on RDF graphs and guarantees decidability of reasoning. Additionally, we provide an equivalence statement between the ERDF #n-stable and stable model semantics. Section 4 provides various complexity results for ERDF #n-stable and stable model semantics. Finally, Section 4 concludes the paper and reviews related work.

⁴ Of course, this declaration could have been included (equivalently) in G , instead of P .

2 Stable Model Semantics of ERDF Ontologies

In this Section, we briefly review the stable model semantics of ERDF ontologies. Details and examples can be found in [1].

A (Web) *vocabulary* V is a set of URI references and/or literals (plain or typed). We denote the set of all URI references by \mathcal{URI} , the set of all plain literals by \mathcal{PL} , the set of all typed literals by \mathcal{TL} , and the set of all literals by \mathcal{LIT} . We consider a set Var of variable symbols, such that the sets Var , \mathcal{URI} , \mathcal{LIT} are pairwise disjoint. In our examples, variable symbols are prefixed by “?”.

Let V be a vocabulary. An *ERDF triple* over V is an expression of the form $p(s, o)$ or $\neg p(s, o)$, where $s, o \in V \cup Var$ are called *subject* and *object*, respectively, and $p \in V \cap \mathcal{URI}$ is called *property*. An *ERDF graph* G is a set of ERDF triples over some vocabulary V . We denote the variables appearing in G by $Var(G)$, and the set of URI references and literals appearing in G by V_G .

Let V be a vocabulary. We denote by $L(V)$ the smallest set that contains the ERDF triples over V and is closed with respect to the following conditions: if $F, G \in L(V)$ then $\{\sim F, F \wedge G, F \vee G, F \supset G, \exists x F, \forall x F\} \subseteq L(V)$, where $x \in Var$. An *ERDF formula* over V is an element of $L(V)$. We denote the set of variables appearing in F by $Var(F)$, and the set of free variables appearing in F by $FVar(F)$. Moreover, we denote the set of URI references and literals appearing in F by V_F .

Intuitively, an ERDF graph G represents an existentially quantified conjunction of ERDF triples. Specifically, let $G = \{t_1, \dots, t_m\}$ be an ERDF graph, and let $Var(G) = \{x_1, \dots, x_k\}$. Then, G represents the ERDF formula $formula(G) = \exists?x_1, \dots, \exists?x_k t_1 \wedge \dots \wedge t_m$. Existentially quantified variables in ERDF graphs are handled by *skolemization*. Let G be an ERDF graph. The *skolemization function* of G is an 1:1 mapping $sk_G : Var(G) \rightarrow \mathcal{URI}$, where for each $x \in Var(G)$, $sk_G(x)$ is an artificial URI, denoted by $G:x$. The *skolemization* of G , denoted by $sk(G)$, is the ground ERDF graph derived from G after replacing each $x \in Var(G)$ by $sk_G(x)$.

An *ERDF rule* r over a vocabulary V is an expression of the form: $Concl(r) \leftarrow Cond(r)$, where $Cond(r) \in L(V) \cup \{true\}$ and $Concl(r)$ is an ERDF triple or *false*. We denote the set of variables and the set of free variables of r by $Var(r)$ and $FVar(r)$ ⁵, respectively. An *ERDF program* P is a set of ERDF rules. We denote the set of URI references and literals appearing in P by V_P .

An *ERDF ontology* is a pair $O = \langle G, P \rangle$, where G is an ERDF graph and P is an ERDF program.

A partial interpretation is an extension of a simple interpretation of RDF semantics [8], where each property is associated not only with a truth extension but also with a falsity extension.

Definition 1 (Partial interpretation). A *partial interpretation* I of a vocabulary V consists of:

- A non-empty set of resources Res_I , a set of properties $Prop_I$, and a set of literal values $LV_I \subseteq Res_I$, which contains $V \cap \mathcal{PL}$.

⁵ $FVar(r) = FVar(F) \cup FVar(G)$.

- A vocabulary interpretation mapping: $I_V : V \cap \mathcal{URI} \rightarrow Res_I \cup Prop_I$.
- A property-truth extension mapping⁶: $PT_I : Prop_I \rightarrow \mathcal{P}(Res_I \times Res_I)$.
- A property-falsity extension mapping: $PF_I : Prop_I \rightarrow \mathcal{P}(Res_I \times Res_I)$.
- A mapping $IL_I : V \cap \mathcal{TL} \rightarrow Res_I$.

We define the mapping: $I : V \rightarrow Res_I \cup Prop_I$, called *denotation*, such that: (i) $I(x) = I_V(x)$, $\forall x \in V \cap \mathcal{URI}$, (ii) $I(x) = x$, $\forall x \in V \cap \mathcal{PL}$, and (iii) $I(x) = IL_I(x)$, $\forall x \in V \cap \mathcal{TL}$. \square

A partial interpretation I of a vocabulary V is *coherent* iff for all $x \in Prop_I$, $PT_I(x) \cap PF_I(x) = \emptyset$.

Let I be a partial interpretation of a vocabulary V and let v be a partial function $v : Var \rightarrow Res_I$ (called *valuation*). If $x \in Var$, we define $[I + v](x) = v(x)$. If $x \in V$, we define $[I + v](x) = I(x)$.

Definition 2. (Satisfaction of an ERDF formula w.r.t. a partial interpretation and a valuation) Let F, G be ERDF formulas and let I be a partial interpretation of a vocabulary V . Additionally, let v be a mapping $v : Var(F) \rightarrow Res_I$.

- If $F = p(s, o)$ then $I, v \models F$ iff $p \in V \cap \mathcal{URI}$, $s, o \in V \cup Var$, $I(p) \in Prop_I$, and $\langle [I + v](s), [I + v](o) \rangle \in PT_I(I(p))$.
- If $F = \neg p(s, o)$ then $I, v \models F$ iff $p \in V \cap \mathcal{URI}$, $s, o \in V \cup Var$, $I(p) \in Prop_I$, and $\langle [I + v](s), [I + v](o) \rangle \in PF_I(I(p))$.
- If $F = \sim G$ then $I, v \models F$ iff $V_G \subseteq V$ and $I, v \not\models G$.
- If $F = F_1 \wedge F_2$ then $I, v \models F$ iff $I, v \models F_1$ and $I, v \models F_2$.
- If $F = F_1 \vee F_2$ then $I, v \models F$ iff $I, v \models F_1$ or $I, v \models F_2$.
- If $F = F_1 \supset F_2$ then $I, v \models F$ iff $I, v \models \sim F_1 \vee F_2$.
- If $F = \exists x G$ then $I, v \models F$ iff there exists mapping $u : Var(G) \rightarrow Res_I$ such that $u(y) = v(y)$, $\forall y \in Var(G) - \{x\}$, and $I, u \models G$.
- If $F = \forall x G$ then $I, v \models F$ iff for all mappings $u : Var(G) \rightarrow Res_I$ such that $u(y) = v(y)$, $\forall y \in Var(G) - \{x\}$, it holds $I, u \models G$. \square

Let F be an ERDF formula, let G be an ERDF graph, and let I be a partial interpretation of a vocabulary V . We define: $I \models F$ iff for each mapping $v : Var(F) \rightarrow Res_I$, it holds that $I, v \models F$. Additionally, we define: $I \models G$ iff $I \models formula(G)$.

We assume that for every partial interpretation I , it holds that $I \models true$ and $I \not\models false$.

The vocabulary of RDF, \mathcal{V}_{RDF} , is a set of \mathcal{URI} references in the *rdf*: namespace [8]. The vocabulary of RDFS, \mathcal{V}_{RDFS} , is a set of \mathcal{URI} references in the *rdfs*: namespace [8]. The *vocabulary of ERDF* is defined as $\mathcal{V}_{ERDF} = \{erdf:TotalClass, erdf:TotalProperty\}$. Intuitively, instances of the metaclass *erdf:TotalClass* are classes c that satisfy totalness, meaning that each resource x belongs either to the truth or falsity extension of c (i.e., the statement “ x is of type c ” is either true or explicitly false). Similarly, instances of the metaclass *erdf:TotalProperty*

⁶ The notation $\mathcal{P}(S)$, where S is a set, denotes the powerset of S .

are properties p that satisfy totalness, meaning that each pair of resources $\langle x, y \rangle$ belongs either to the truth or falsity extension of p (i.e., the statement “ $\langle x, y \rangle$ satisfies property p ” is either true or explicitly false).

Definition 3 (ERDF interpretation). An *ERDF interpretation* I of a vocabulary V is a coherent, partial interpretation of $V \cup \mathcal{V}_{RDF} \cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF}$, extended by the new ontological categories $Cls_I \subseteq Res_I$ for classes, $TCls_I \subseteq Cls_I$ for total classes, and $TProp_I \subseteq Prop_I$ for total properties, as well as the class-truth extension mapping $CT_I : Cls_I \rightarrow \mathcal{P}(Res_I)$, and the class-falsity extension mapping $CF_I : Cls_I \rightarrow \mathcal{P}(Res_I)$, such that:

1. $x \in CT_I(y)$ iff $\langle x, y \rangle \in PT_I(I(rdf:type))$, and
 $x \in CF_I(y)$ iff $\langle x, y \rangle \in PF_I(I(rdf:type))$.
2. The ontological categories are defined as follows:
 $Prop_I = CT_I(I(rdf:Property))$ $Cls_I = CT_I(I(rdfs:Class))$
 $Res_I = CT_I(I(rdfs:Resource))$ $LV_I = CT_I(I(rdfs:Literal))$
 $TCls_I = CT_I(I(erd़:TotalClass))$ $TProp_I = CT_I(I(erd़:TotalProperty))$.
3. If $\langle x, y \rangle \in PT_I(I(rdfs:domain))$ and $\langle z, w \rangle \in PT_I(x)$ then $z \in CT_I(y)$.
4. If $\langle x, y \rangle \in PT_I(I(rdfs:range))$ and $\langle z, w \rangle \in PT_I(x)$ then $w \in CT_I(y)$.
5. If $x \in Cls_I$ then $\langle x, I(rdfs:Resource) \rangle \in PT_I(I(rdfs:subClassOf))$.
6. If $\langle x, y \rangle \in PT_I(I(rdfs:subClassOf))$ then
 $x, y \in Cls_I$, $CT_I(x) \subseteq CT_I(y)$, and $CF_I(y) \subseteq CF_I(x)$.
7. $PT_I(I(rdfs:subClassOf))$ is a reflexive and transitive relation on Cls_I .
8. If $\langle x, y \rangle \in PT_I(I(rdfs:subPropertyOf))$ then
 $x, y \in Prop_I$, $PT_I(x) \subseteq PT_I(y)$, and $PF_I(y) \subseteq PF_I(x)$.
9. $PT_I(I(rdfs:subPropertyOf))$ is a reflexive and transitive relation on $Prop_I$.
10. If $x \in CT_I(I(rdfs:Datatype))$ then
 $\langle x, I(rdfs:Literal) \rangle \in PT_I(I(rdfs:subClassOf))$.
11. If $x \in CT_I(I(rdfs:ContainerMembershipProperty))$ then
 $\langle x, I(rdfs:member) \rangle \in PT_I(I(rdfs:subPropertyOf))$.
12. If $x \in TCls_I$ then $CT_I(x) \cup CF_I(x) = Res_I$.
13. If $x \in TProp_I$ then $PT_I(x) \cup PF_I(x) = Res_I \times Res_I$.
14. If “ s ” \sim $rdf:XMLLiteral$ $\in V$ and s is a well-typed XML literal string, then
 $IL_I(\text{“}s\text{”}\sim rdf:XMLLiteral)$ is the XML value of s , and
 $IL_I(\text{“}s\text{”}\sim rdf:XMLLiteral) \in CT_I(I(rdf:XMLLiteral))$.
15. If “ s ” \sim $rdf:XMLLiteral$ $\in V$ and s is an ill-typed XML literal string then
 $IL_I(\text{“}s\text{”}\sim rdf:XMLLiteral) \in Res_I - LV_I$, and
 $IL_I(\text{“}s\text{”}\sim rdf:XMLLiteral) \in CF_I(I(rdfs:Literal))$.
16. I satisfies the RDF and RDFS axiomatic triples [8], respectively.
17. I satisfies the following triples, called *ERDF axiomatic triples*:
 $rdfs:subClassOf(erd़:TotalClass, rdfs:Class)$.
 $rdfs:subClassOf(erd़:TotalProperty, rdfs:Class)$. □

The *vocabulary* of an ERDF ontology O is defined as $V_O = V_{sk(G)} \cup V_P \cup \mathcal{V}_{RDF} \cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF}$. Additionally, we denote by Res_O^H the union of V_O and the set of XML values of the well-typed XML literals in V_O minus the well-typed XML literals.

Definition 4 (Herbrand interpretation of an ERDF ontology). Let $O = \langle G, P \rangle$ be an ERDF ontology and let I be an ERDF interpretation of V_O . We say that I is a *Herbrand interpretation* of O iff: (i) $\text{Res}_I = \text{Res}_O^H$, (ii) $I_V(x) = x$, for all $x \in V_O \cap \mathcal{URI}$, (iii) $IL_I(x) = x$, if x is a typed literal in V_O other than a well-typed XML literal, and $IL_I(x)$ is the XML value of x , if x is a well-typed XML literal in V_O . We denote the set of Herbrand interpretations of O by $\mathcal{I}^H(O)$. \square

Let $O = \langle G, P \rangle$ be an ERDF ontology and let $I, J \in \mathcal{I}^H(O)$. We say that J extends I , denoted by $I \leq J$, iff $\text{Prop}_I \subseteq \text{Prop}_J$, and $\forall p \in \text{Prop}_I, \text{PT}_I(p) \subseteq \text{PT}_J(p)$ and $\text{PF}_I(p) \subseteq \text{PF}_J(p)$.

Let V be a vocabulary and let r be an ERDF rule. We denote by $[r]_V$ the set of rules that result from r if we replace each variable $x \in FVar(r)$ by $v(x)$, for all mappings $v : FVar(r) \rightarrow V$. Let P be an ERDF program. We define $[P]_V = \bigcup_{r \in P} [r]_V$.

Below, we define the stable models of an ERDF ontology, based on the coherent stable models of Partial Logic [9].

Definition 5 (ERDF stable model). Let $O = \langle G, P \rangle$ be an ERDF ontology and let $M \in \mathcal{I}^H(O)$. We say that M is an *(ERDF) stable model* of O iff there is a chain of Herbrand interpretations of O , $I_0 \leq \dots \leq I_{k+1}$ such that $I_k = I_{k+1} = M$ and:

1. $I_0 \in \text{minimal}(\{I \in \mathcal{I}^H(O) \mid I \models sk(G)\})$.
2. For successor ordinals α with $0 < \alpha \leq k + 1$:
 $I_\alpha \in \text{minimal}(\{I \in \mathcal{I}^H(O) \mid I \geq I_{\alpha-1} \text{ and } I \models \text{Concl}(r), \forall r \in P_{[I_{\alpha-1}, M]}\})$, where
 $P_{[I_{\alpha-1}, M]} = \{r \in [P]_{V_O} \mid I \models \text{Cond}(r), \forall I \in \mathcal{I}^H(O) \text{ s.t. } I_{\alpha-1} \leq I \leq M\}$.

The set of stable models of O is denoted by $\mathcal{M}^{st}(O)$. \square

Note that I_0 is a minimal Herbrand interpretation of $O = \langle G, P \rangle$ that satisfies $sk(G)$, while Herbrand interpretations I_1, \dots, I_{k+1} correspond to a stratified sequence of rule applications, where all applied rules remain applicable throughout the generation of stable model M .

Let $O = \langle G, P \rangle$ be an ERDF ontology and let F be an ERDF formula or ERDF graph. We say that O entails F under the *(ERDF) stable model semantics*, denoted by $O \models^{st} F$, iff for all $M \in \mathcal{M}^{st}(O)$, $M \models F$.

Example 2. Consider the ERDF ontology O of Example 1. Then, O has two stable models M_1, M_2 , where $M_1 \models \neg \text{rdf:type}(Anne, \text{Adult})$ and $M_2 \models \text{rdf:type}(Anne, \text{Adult})$ ⁷. For both $M \in \{M_1, M_2\}$, it holds $M \models \text{serveSoftDrink}(Anne, \text{Coca-Cola})$. This is because, if $Anne$ is not an adult then, since she is not a child, it is decided to drink *Coca-Cola*. If $Anne$ is an adult then, since it is not known if she likes wine, it is also decided to drink *Coca-Cola*. Thus, it holds $O \models^{st} \text{serveSoftDrink}(Anne, \text{Coca-Cola})$. Additionally, for both $M \in \{M_1,$

⁷ Note that *ex:Adult* is a total class and that we do not know if *Anne* is an adult.

$M_2\}$, it holds $M \models \text{rdf:type}(Retsina, SelectedWine) \wedge \neg \text{rdf:type}(Riesling, SelectedWine)$. This is because (i) both *Gerd* and *Carlos* like *Retsina* and (ii) *Carlos* likes *only Retsina*. Thus, it holds $O \models^{st} \text{rdf:type}(Retsina, SelectedWine) \wedge \neg \text{rdf:type}(Riesling, SelectedWine)$. \square

In [1], it is shown that stable model entailment conservatively extends RDFS entailment from RDF graphs to ERDF ontologies.

Proposition 1. Let G, G' be RDF graphs such that $V_G \cap \mathcal{V}_{ERDF} = \emptyset$, $V_{G'} \cap \mathcal{V}_{ERDF} = \emptyset$, and $V_{G'} \cap sk_G(\text{Var}(G)) = \emptyset$. It holds: $G \models^{\text{RDFS}} G'$ iff $\langle G, \emptyset \rangle \models^{st} G'$.

3 Undecidability of ERDF Stable Model Semantics Leads to $\#n$ -Stable Model Semantics

Unfortunately, satisfiability and entailment under the ERDF stable model semantics are in general undecidable [1]. The proof of undecidability exploits a reduction from the *unbounded tiling problem*, whose existence of a solution is known to be undecidable [2]. Note that since each constraint $false \leftarrow F$ that appears in an ERDF ontology O can be replaced by the rule $\neg t \leftarrow F$, where t is an RDF, RDFS, or ERDF axiomatic triple, the presence of constraints in O does not affect decidability.

Definition 6 (Simple, Objective ERDF ontology). An ERDF formula F is called *simple* if it has the form $t_1 \wedge \dots \wedge t_k \wedge \neg t_{k+1} \wedge \dots \wedge \neg t_m$, where each t_i , $i = 1, \dots, m$, is an ERDF triple. An ERDF program P is called *simple* if for all $r \in P$, $\text{Cond}(r)$ is a simple ERDF formula or *true*. An ERDF ontology $O = \langle G, P \rangle$ is called *simple*, if P is a simple ERDF program. A simple ERDF ontology O (resp. ERDF program P) is called *objective*, if no weak negation appears in O (resp. P). \square

Reduction in [1] shows that ERDF stable model satisfiability and entailment remain undecidable, even if (i) $O = \langle G, P \rangle$ is a simple ERDF ontology, (ii) the terms *erdf:TotalClass* and *erdf:TotalProperty* do not appear in O (i.e., $(V_G \cup V_P) \cap \mathcal{V}_{ERDF} = \emptyset$), and (iii) the entailed formula has the form $\exists \bar{x} F$, where F is a simple ERDF formula and \bar{x} are the variables appearing in F . Moreover, we can prove by a reduction from the unbounded tiling problem [2] that even if $O = \langle G, P \rangle$ is an objective ERDF ontology, entailment of a *general* ERDF formula F under the ERDF stable model semantics is still undecidable.

Let O be a *general* ERDF ontology. The source of undecidability of the ERDF stable model semantics of O is the fact that \mathcal{V}_{RDF} is infinite. Thus, the vocabulary of O is also infinite (note that $\{ \text{rdf}:i \mid i \geq 1 \} \subseteq \mathcal{V}_{RDF} \subseteq V_O$). In this Section, we slightly modify the definition of the ERDF stable model semantics, based on a redefinition of the vocabulary of an ERDF ontology, which now becomes finite. We call the modified semantics, the *ERDF $\#n$ -stable model semantics* (for $n \in \mathbb{N}$).

In order to define the ERDF $\#n$ -stable model semantics, we need to modify several of the definitions on which the ERDF stable model semantics is based. Specifically:

- We define: $\mathcal{V}_{RDF}^{\#n} = \mathcal{V}_{RDF} - \{rdf:_i \mid i > n\}$.
- An *ERDF $\#n$ -interpretation* is defined exactly as an ERDF interpretation in Def. 3 except that \mathcal{V}_{RDF} is replaced by $\mathcal{V}_{RDF}^{\#n}$ and in semantic condition 16, only the RDF and RDFS axiomatic triples that contain \mathcal{URI} references in $\mathcal{V}_{RDF}^{\#n}$ are considered.
- Let $O = \langle G, P \rangle$ be an ERDF ontology. We define: $V_O^{\#n} = V_O - \{rdf:_i \mid i > n\}$, and $Res_O^{H^{\#n}} = Res_O^H - \{rdf:_i \mid i > n\}$.
- Let $O = \langle G, P \rangle$ be an ERDF ontology. An $\#n$ -Herbrand interpretation I of O is an ERDF $\#n$ -interpretation of $V_O^{\#n}$ such that: (i) $Res_I = Res_O^{H^{\#n}}$, (ii) $I_V(x) = x$, for all $x \in V_O^{\#n} \cap \mathcal{URI}$, (iii) $IL_I(x) = x$, if x is a typed literal in $V_O^{\#n}$ other than a well-typed XML literal, and $IL_I(x)$ is the XML value of x , if x is a well-typed XML literal in $V_O^{\#n}$. We denote the set of $\#n$ -Herbrand interpretations of O by $\mathcal{I}^{H^{\#n}}(O)$.
- Let $O = \langle G, P \rangle$ be an ERDF ontology. An (*ERDF*) $\#n$ -stable model of O is defined as a stable model of O in Def. 5, except that $\mathcal{I}^H(O)$ is replaced by $\mathcal{I}^{H^{\#n}}(O)$ and V_O is replaced by $V_O^{\#n}$. The set of $\#n$ -stable models of O is denoted by $\mathcal{M}^{st\#n}(O)$.

Let $O = \langle G, P \rangle$ be an ERDF ontology and let F be an ERDF formula or ERDF graph. Let $n \in \mathbb{N}$. We say that O entails F under the (*ERDF*) $\#n$ -stable model semantics, denoted by $O \models^{st\#n} F$ iff for all $M \in \mathcal{M}^{st\#n}(O)$, $M \models F$.

Let $O = \langle G, P \rangle$ be an ERDF ontology and let F be an ERDF formula. Let $n \in \mathbb{N}$. The (*ERDF*) $\#n$ -stable answers of F w.r.t. O are defined as follows⁸:

$$Ans_O^{st\#n}(F) = \begin{cases} \text{"yes"} & \text{if } FVar(F) = \emptyset \text{ and } \forall M \in \mathcal{M}^{st\#n}(O) : M \models F \\ \text{"no"} & \text{if } FVar(F) = \emptyset \text{ and } \exists M \in \mathcal{M}^{st\#n}(O) : M \not\models F \\ \{v : FVar(F) \rightarrow V_O^{\#n} \mid \forall M \in \mathcal{M}^{st\#n}(O) : M \models v(F)\} & \text{if } FVar(F) \neq \emptyset \end{cases}$$

Let $O = \langle G, P \rangle$ be an ERDF ontology. We define: (i) $n_O = 0$, if $(V_G \cup V_P) \cap \{rdf:_i \mid i \geq 1\} = \emptyset$, and (ii) $n_O = \max(\{i \in \mathbb{N} \mid rdf:_i \in V_G \cup V_P\})$, otherwise.

For example, if O is the ERDF ontology of Example 1 then $n_O = 0$.

Proposition 2 below relates stable model entailment and $\#n$ -stable model entailment. First, we provide a definition. Let F be an ERDF formula. We say that F is an *ERDF d-formula* iff (i) F is the disjunction of existentially quantified conjunctions of ERDF triples, and (ii) $FVar(F) = \emptyset$. For example, let $F = (\exists ?x rdf:type(?x, Vertex) \wedge rdf:type(?x, Red)) \vee (\exists ?x rdf:type(?x, Vertex) \wedge \neg rdf:type(?x, Blue))$. Then, F is an ERDF d-formula. It is easy to see that if G is an ERDF graph then $formula(G)$ is an ERDF d-formula.

Proposition 2. Let $O = \langle G, P \rangle$ be an objective ERDF ontology and let $n \geq \max(n_O, 1)$. Let F^d be an ERDF d-formula s.t. $\max(\{i \in \mathbb{N} \mid rdf:_i \in V_{F^d}\}) \leq n$. It holds: $O \models^{st} F^d$ iff $O \models^{st\#n} F^d$.

⁸ $v(F)$ results from F after replacing all the free variables x in F by $v(x)$.

Since $V_O^{\#n}$ (for $n \in \mathbb{N}$) is finite, query answering under the ERDF $\#n$ -stable model semantics is decidable. Now, since satisfiability under the ERDF stable model semantics is in general undecidable, Proposition 2 does not hold in the case that $O = \langle G, P \rangle$ is a general ERDF ontology. Moreover, Proposition 2 does not hold in the case that F is a general ERDF formula. For example, consider the ERDF graph G :

$$G = \{ \text{rdf:type}(x, c1) \mid x \in \{c1, c2, id\} \cup \mathcal{V}_{RDF}^{\#0} \cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF} \}$$

Additionally, consider the ERDF program $P = \{id(?x, ?x) \leftarrow \text{true.}\}$ and the ERDF formula F (which is not an ERDF d -formula):

$$F = \exists?x, \exists?y \sim \text{rdf:type} (?x, c1) \wedge \sim \text{rdf:type} (?y, c1) \wedge \sim id (?x, ?y).$$

Let $O = \langle G, P \rangle$. It holds, $n_O = 0$. Note that $O \models^{st} F$, while $O \not\models^{st\#1} F$.

The following proposition is a direct consequence of Propositions 1 and 2, and shows that $\#n$ -stable model entailment also extends RDFS entailment from RDF graphs to ERDF ontologies.

Proposition 3. Let G, G' be RDF graphs such that $V_G \cap \mathcal{V}_{ERDF} = \emptyset$, $V_{G'} \cap \mathcal{V}_{ERDF} = \emptyset$, and $V_{G'} \cap sk_G(\text{Var}(G)) = \emptyset$. Let $O = \langle G, \emptyset \rangle$ and $n \geq \max(n_O, 1)$. If $\max(\{i \in \mathbb{N} \mid \text{rdf:-}i \in V_{G'}\}) \leq n$ then: $G \models^{RDFS} G'$ iff $O \models^{st\#n} G'$.

4 Complexity Results

In this section, we provide complexity results for (i) the ERDF $\#n$ -stable model semantics on simple and objective ERDF ontologies, and (ii) the ERDF stable model semantics on objective ERDF ontologies. Additionally, for $n \in \mathbb{N}$, we show that the $\#n$ -stable answers of a simple ERDF formula F w.r.t. a simple ERDF ontology $O = \langle G, P \rangle$ can be computed through Answer Set Programming [7] on an extended logic program (ELP) $\Pi_O^{\#n}$ (not given here due to space limitations).

Let Π be an extended logic program (ELP) and let F be a query of the form $L_1 \wedge \dots \wedge L_k \wedge \sim L_{k+1} \wedge \dots \wedge \sim L_m$, where $L_i, i = 1, \dots, m$, is an ELP literal. We will denote by $\text{Ans}_{\Pi}^{AS}(F)$ the (skeptical) answers of F w.r.t. Π according to answer set semantics [7].

Proposition 4. Let $O = \langle G, P \rangle$ be a simple ERDF ontology and let F be a simple ERDF formula. Let $n \in \mathbb{N}$. It exists an ELP $\Pi_O^{\#n}$ generated in polynomial time w.r.t. the size of O and n s.t. $\text{Ans}_O^{st\#n}(F) = \text{Ans}_{\Pi_O^{\#n}}^{AS}(F')$, where F' is the query that results after replacing each ERDF triple $p(s, o)$ appearing in F by the ELP literal $\text{Holds}(s, p, o)$.

Based on Proposition 4 and complexity results for answer set semantics (see [5]), we can state the following Corollary.

Corollary 1. Let $O = \langle G, P \rangle$ be a simple ERDF ontology and let F be an ERDF formula. Additionally, let v be (i) one of {“yes”, “no”}, if $F\text{Var}(F) = \emptyset$, or (ii) a mapping $v : F\text{Var}(F) \rightarrow V_O^{\#n}$, if $F\text{Var}(F) \neq \emptyset$. Let $n \in \mathbb{N}$.

1. The problem of establishing whether O has an $\#n$ -stable model is NP-complete w.r.t. size of $sk(G) \cup [P]_{V_O^{\#n}}$.
2. The problem of establishing whether $v \in Ans_O^{st\#n}(F)$ is co-NP-complete w.r.t. size of $sk(G) \cup [P]_{V_O^{\#n}}$.

Below, we state complexity results for the $\#n$ -stable model semantics of objective ERDF ontologies. We see that even though no weak negation appears in the rules of objective ERDF ontologies, complexity of reasoning w.r.t. simple ERDF ontologies remains the same. This is due to the ERDF metaclasses *erdf:TotalClass* and *erdf:TotalProperty* on the instances of which, the OWA applies.

Proposition 5. Let $O = \langle G, P \rangle$ be an objective ERDF ontology. Let G' be an ERDF graph and let F be an ERDF formula. Additionally, let v be (i) one of {“yes”, “no”}, if $FVar(F) = \emptyset$, or (ii) a mapping $v : FVar(F) \rightarrow V_O^{\#n}$, if $FVar(F) \neq \emptyset$. Let $n \in \mathbb{N}$.

1. The problem of establishing whether O has an $\#n$ -stable model is NP-complete w.r.t. size of $sk(G) \cup [P]_{V_O^{\#n}}$.
2. The problems of establishing whether: (i) $O \models^{st\#n} G'$ and (ii) $O \models^{st\#n} F$ are co-NP-complete w.r.t. size of $sk(G) \cup [P]_{V_O^{\#n}}$.

The hardness part of the above complexity results can be proved by a reduction from the *Graph 3-Colorability* problem, which is a classical NP-complete problem.

Based on Proposition 2 and Proposition 5, it follows:

Corollary 2. Let $O = \langle G, P \rangle$ be an objective ERDF ontology. Let G' be an ERDF graph and let F^d be an ERDF d -formula s.t. $\max(\{i \in \mathbb{N} \mid \text{rdf}:_i \in V_X\}) \leq n_O$, where $X \in \{G', F_d\}$.

1. The problem of establishing whether O has a stable model is NP-complete w.r.t. size of $sk(G) \cup [P]_{V_O^{\#n_O}}$.
2. The problems of establishing whether: (i) $O \models^{st} G'$ and (ii) $O \models^{st} F^d$ are co-NP-complete w.r.t. size of $sk(G) \cup [P]_{V_O^{\#n_O}}$.

Yet, as mentioned in Section 3, satisfiability and entailment of *simple* (and of course, general) ERDF ontologies under the ERDF stable model semantics are undecidable.

5 Conclusions and Related Work

In this paper, we elaborated on the computability and complexity issues of the stable model semantics of ERDF ontologies. We show that decidability under this semantics cannot be achieved, unless ERDF ontologies of restricted syntax are considered. We propose the $\#n$ -stable model semantics of ERDF ontologies

(for $n \in \mathbb{N}$) and show that entailment under this semantics extends RDFS entailment. Moreover, query answering under the ERDF $\#n$ -stable model semantics is decidable. An equivalence statement between the ERDF stable and $\#n$ -stable model semantics, as well as various complexity results are provided. Future work concerns the implementation of the $\#n$ -stable model semantics on ERDF ontologies, as well as the extension of our complexity results to other syntax restricted ERDF ontologies and general ERDF ontologies.

Notation 3 (N3) [4] provides a more human readable syntax for RDF and also extends RDF by adding numerous pre-defined constructs for being able to express rules conveniently. In particular, N3 contains a built-in (`log:notIncludes`) for expressing simple negation-as-failure tests and another built-in (`log:definitiveDocument`) for making restricted completeness assumptions. However, N3 does not provide strong negation and closed-world reasoning is not fully supported. In [11], RDF graphs are extended with a set of rules R and R -entailment is defined, extending RDFS entailment. However, in this work, weak and strong negation are not considered. In [6], RDFS is extended with rules and/or general axioms, using embeddings in F-Logic [10]. However, such extensions are not entirely faithful to the model-theoretic semantics of RDF.

References

1. Analyti, A., Antoniou, G., Damásio, C.V., Wagner, G.: Extended RDF as a Semantic Foundation of Rule Markup Languages. *Journal of Artificial Intelligence Research (JAIR)* 32, 37–94 (2008)
2. Berger, R.: The Undecidability of the Dominoe Problem. *Memoirs of the American Mathematical Society* 66, 1–72 (1966)
3. Berners-Lee, T.: Design Issues - Architectural and Philosophical Points. Personal notes (1998), <http://www.w3.org/DesignIssues>
4. Berners-Lee, T., Connolly, D., Kagal, L., Scharf, Y., Hendler, J.: N3Logic: A Logical Framework For the World Wide Web. *Theory and Practice of Logic Programming (TPLP)* 8(3), 249–269 (2008)
5. Dantsin, E., Eiter, T., Gottlob, G., Voronkov, A.: Complexity and expressive power of logic programming. *ACM Computing Surveys* 33(3), 374–425 (2001)
6. de Bruijn, J., Heymans, S.: RDF and Logic: Reasoning and Extension. In: 6th International Workshop on Web Semantics (WebS 2007), co-located with DEXA 2007, pp. 460–464 (2007)
7. Gelfond, M., Lifschitz, V.: Logic programs with Classical Negation. In: 7th International Conference on Logic Programming, pp. 579–597 (1990)
8. Hayes, P.: RDF Semantics. W3C Recommendation, February 10 (2004), <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/>
9. Herre, H., Jaspars, J., Wagner, G.: Partial Logics with Two Kinds of Negation as a Foundation of Knowledge-Based Reasoning. In: Gabbay, D.M., Wansing, H. (eds.) *What Is Negation?*. Kluwer Academic Publishers, Dordrecht (1999)
10. Kifer, M., Lausen, G., Wu, J.: Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM* 42(4), 741–843 (1995)
11. ter Horst, H.J.: Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005, vol. 3729, pp. 668–684. Springer, Heidelberg (2005)

Toward Formalizing Common-Sense Psychology: An Analysis of the False-Belief Task

Konstantine Arkoudas and Selmer Bringsjord

Cognitive Science and Computer Science Departments, RPI
`arkouk@rpi.edu, brings@rpi.edu`

Abstract. Predicting and explaining the behavior of others in terms of mental states is indispensable for everyday life. It will be equally important for artificial agents. We present an inference system for representing and reasoning about certain types of mental states, and use it to provide a formal analysis of the false-belief task. The system allows for the representation of information about events, causation, and perceptual, doxastic, and epistemic states (vision, belief, and knowledge), incorporating ideas from the event calculus and multi-agent epistemic logic. Unlike previous AI formalisms, our focus here is on mechanized proofs and proof programmability, not on metamathematical results. Reasoning is performed via cognitively plausible inference rules, and automation is achieved by general-purpose inference *methods*. The system has been implemented as an interactive theorem prover and is available for experimentation.¹

1 Introduction

Predicting and explaining the behavior of other people is indispensable for everyday life. The ability to ascribe mental states to others and to reason about such mental states is pervasive and invaluable. All social transactions—from engaging in commerce and negotiating to making jokes and empathizing with other people’s pain or joy—require at least a rudimentary grasp of common-sense psychology (CSP). Artificial agents without an ability of this sort would be severely handicapped in their interactions with humans. This could present problems not only for artificial agents trying to interpret human behavior, but also for artificial agents trying to interpret the behavior of one another. When a system exhibits a complex but rational behavior, and detailed knowledge of its internal structure is not available, the best strategy for predicting and explaining its actions might be to analyze its behavior in intentional terms, i.e., in terms of mental states such as beliefs and desires (regardless of whether the system *actually* has genuine mental states; we are not interested here in whether artificial agents are

¹ The prover, along with code that makes it possible to engineer autonomous synthetic characters (residing on servers in our lab) that have avatars in *Second Life*, has also been used to allow such characters to pass the false-belief task. For demonstrations, visit www.cogsci.rpi.edu/research/rair/asc_rca/SLDemos

capable of having bona fide mental states). Mentalistic models are likely to be particularly apt for agents trying to manipulate the behavior of other agents.

Any computational treatment of CSP will have to integrate action and cognition. Agents must be able to reason about the causes and effects of various events, whether they are non-intentional physical events or intentional events brought about by their own agency. More importantly, they must be able to reason about what others believe or know about such events. To that end, our system combines and adapts ideas drawn from the event calculus and from multi-agent epistemic logics. It is based on multi-sorted first-order logic extended with subsorting, epistemic operators for perception, belief, and knowledge, and mechanisms for reasoning about causation and action. Using subsorting, we formally model agent actions as types of events, which enables us to use the resources of the event calculus to represent and reason about agent actions. The usual axioms of the event calculus are encoded as common knowledge, suggesting that people have an understanding of the basic folk laws of causality (innate or acquired), and are indeed aware that others have such understanding.

It is important to be clear about what we hope to accomplish through the present work. In general, any logical system or methodology capable of representing and reasoning about intentional notions such as knowledge can have at least three different uses. First, it can serve as a tool for the specification, analysis, and verification of rational agents. Second, in tandem with some appropriate reasoning mechanism, it can serve as a knowledge representation framework, i.e., it can be used *by* artificial agents to represent their own “mental states”—and those of other agents—and to deliberate and act in accordance with those states and their environment. Finally, it can be used to provide formal models of certain interesting cognitive phenomena. One intended contribution of our present work is of the third sort, namely, to provide a formal model of false-belief attributions, and, in particular, a description of the logical competence of an agent capable of passing a false-belief task. It addresses questions such as the following: What sort of principles is it plausible to assume that an agent has to deploy in order to be able to succeed on a false-belief task? What is the depth and complexity of the required reasoning? Can such reasoning be automated, and if so, how? These questions have not been taken up in detail in the relevant discussions in cognitive science and the philosophy of mind, which have been couched in overly abstract and rather vague terms. Formal computational models such as the one we present here can help to ground such discussions, to clarify conceptual issues, and to begin to answer important questions in a concrete setting.

Although the import of such a model is primarily scientific, there can be interesting engineering implications. For instance, if the formalism is sufficiently expressive and versatile, and the posited computational mechanisms can be automated with reasonable efficiency, then the system can make contributions to the first two areas mentioned above. We believe that our system has such potential for two reasons. First, the combination of epistemic constructs such as common knowledge with the conceptual resources of the event calculus for dealing with causation appears to afford great expressive power, as demonstrated by

our formalization. A key technical insight behind this combination is the modelling of agent actions as events via subsorting. Second, procedural abstraction mechanisms appear to hold significant promise for automation; we discuss this issue later in more detail.

The remainder of this paper is structured as follows. The next section gives the formal definition of our system. Section 3 represents the false-belief task in this system, and section 4 presents a model of the reasoning that is required to succeed in such a task, carried out in a modular fashion by collaborating methods. Section 5 discusses some related work and concludes.

2 A Calculus for Representing and Reasoning about Mental States

The syntactic and semantic problems that arise when one tries to use classical logic to represent and reason about intentional notions are well-known. Syntactically, modelling belief or knowledge relationally is problematic because one believes or knows arbitrarily complex propositions, whereas the arguments of relation symbols are terms built from constants, variables, and function symbols. (The objects of belief could be encoded as strings, but such representations are too low-level for most purposes.) Semantically, the main issue is the referential opacity (or intensionality) exhibited by propositional-attitude operators. In intensional contexts one cannot freely substitute one coreferential term for another. Broadly speaking, there are two ways of addressing these issues. One is to use a modal logic, with built-in syntactic operators for intentional notions. The other is to retain classical logic but distinguish between an object-language and a meta-language, representing intentional discourse at the object level. Each approach has its advantages and drawbacks. Retaining classical logic has the important advantage of efficiency, in that (semi-)automated deduction systems for classical logic, such as resolution provers—which have made impressive strides over the last decade—can be used for reasoning. This is the option we have chosen in some previous work [3]. One disadvantage of this approach is that when the object language is first-order (includes quantification), then notions such as substitutions and alphabetic equivalence must be explicitly encoded. Depending on the facilities provided by the meta-language, this does not need to be overly onerous, but it does require extra effort. The modal-logic approach has the advantage of solving the syntactic and referential-opacity problems directly, without the need to distinguish an object-language and a meta-language. That is the approach we have taken in this work.

The specification of the syntax of our system appears in figure 1, which describes the various sorts of our universe (S), the signatures of certain built-in function symbols (f), and the abstract syntax of terms (t) and propositions (P). The symbol \sqsubseteq denotes subsorting. Propositions of the form $\mathbf{S}(a, P)$, $\mathbf{B}(a, P)$, and $\mathbf{K}(a, P)$ should be understood as saying that agent a sees that P is the case, believes that P , and knows that P , respectively. Propositions of the form $\mathbf{C}(P)$ assert that P is commonly known. Sort annotations will generally be omitted,

```

 $S ::= \text{Object} \mid \text{Agent} \mid \text{ActionType} \mid \text{Action} \sqsubseteq \text{Event} \mid \text{Moment} \mid \text{Boolean} \mid \text{Fluent}$ 
 $\quad \text{action} : \text{Agent} \times \text{ActionType} \rightarrow \text{Action}$ 
 $\quad \text{initially} : \text{Fluent} \rightarrow \text{Boolean}$ 
 $\quad \text{holds} : \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean}$ 
 $f ::= \text{happens} : \text{Event} \times \text{Moment} \rightarrow \text{Boolean}$ 
 $\quad \text{clipped} : \text{Moment} \times \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean}$ 
 $\quad \text{initiates} : \text{Event} \times \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean}$ 
 $\quad \text{terminates} : \text{Event} \times \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean}$ 
 $\quad \text{prior} : \text{Moment} \times \text{Moment} \rightarrow \text{Boolean}$ 
 $t ::= x : S \mid c : S \mid f(t_1, \dots, t_n)$ 
 $P ::= t : \text{Boolean} \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q \mid P \Leftrightarrow Q \mid$ 
 $\quad \forall x : S . P \mid \exists x : S . P \mid \mathbf{S}(a, P) \mid \mathbf{K}(a, P) \mid \mathbf{B}(a, P) \mid \mathbf{C}(P)$ 

```

Fig. 1. The specification of sorts, function symbols, terms, and propositions

as they are easily deducible from the context. We write $P[x \mapsto t]$ for the proposition obtained from P by replacing every free occurrence of x by t , assuming that t is of a sort compatible with the sort of the free occurrences in question, and taking care to rename P as necessary to avoid variable capture. We use the infix notation $t_1 < t_2$ instead of $\text{prior}(t_1, t_2)$.

We express the following standard axioms of the event calculus as common knowledge:

- [A₁] $\mathbf{C}(\forall f, t . \text{initially}(f) \wedge \neg \text{clipped}(0, f, t) \Rightarrow \text{holds}(f, t))$
- [A₂] $\mathbf{C}(\forall e, f, t_1, t_2 . \text{happens}(e, t_1) \wedge \text{initiates}(e, f, t_1) \wedge t_1 < t_2 \wedge \neg \text{clipped}(t_1, f, t_2) \Rightarrow \text{holds}(f, t_2))$
- [A₃] $\mathbf{C}(\forall t_1, f, t_2 . \text{clipped}(t_1, f, t_2) \Leftrightarrow [\exists e, t . \text{happens}(e, t) \wedge t_1 < t < t_2 \wedge \text{terminates}(e, f, t)])$

suggesting that people have a (possibly innate) understanding of basic causality principles, and are indeed aware that everybody has such an understanding. In addition to [A₁]—[A₃], we postulate a few more axioms pertaining to what people know or believe about causality. First, agents know the events that they intentionally bring about themselves—that is part of what “action” means. In fact, this is common knowledge. The following axiom expresses this:

$$[A_4] \mathbf{C}(\forall a, d, t . \text{happens}(\text{action}(a, d), t) \Rightarrow \mathbf{K}(a, \text{happens}(\text{action}(a, d), t)))$$

The next axiom states that it is common knowledge that if an agent a believes that a certain fluent f holds at t and he does not believe that f has been clipped between t and t' , then he will also believe that f holds at t' :

$$[A_5] \mathbf{C}(\forall a, f, t, t' . \mathbf{B}(a, \text{holds}(f, t)) \wedge \mathbf{B}(a, t < t') \wedge \neg \mathbf{B}(a, \text{clipped}(t, f, t')) \Rightarrow \mathbf{B}(a, \text{holds}(f, t')))$$

The final axiom states that if a believes that b believes that f holds at t_1 and a believes that nothing has happened between t_1 and t_2 to change b 's mind, then a will believe that b will not think that f has been clipped between t_1 and t_2 :

$$[A_6] \forall a, b, t_1, t_2, f . [\mathbf{B}(a, \mathbf{B}(b, \text{holds}(f, t_1))) \wedge \mathbf{B}(a, \neg \exists e, t . \mathbf{B}(b, \text{happens}(e, t)) \wedge \mathbf{B}(b, t_1 < t < t_2) \wedge \mathbf{B}(b, \text{terminates}(e, f, t))] \Rightarrow \mathbf{B}(a, \neg \mathbf{B}(b, \text{clipped}(t_1, f, t_2)))$$

This captures a form of closed-world reasoning, for it could well be the case that, in fact, b has come to believe that something has happened between t and t' that

terminated f , and therefore no longer believes that f holds. But if a believes that there have been no such events, then it is reasonable for a to assume that b will not believe that f has been clipped.

In addition to the usual introduction and elimination rules for first-order predicate logic with equality, we will make use of the following inference rules:

$$\frac{\mathbf{C}(\mathbf{S}(a, P) \Rightarrow \mathbf{K}(a, P)) \quad [R_1]}{\mathbf{C}(\mathbf{K}(a, P) \Rightarrow \mathbf{B}(a, P)) \quad [R_2]}$$

$$\frac{\mathbf{C}(P) \quad [R_3] \quad \frac{\mathbf{K}(a, P)}{P} \quad [R_4]}{\mathbf{K}(a_1, \mathbf{K}(a_2, \mathbf{K}(a_3, P)))}$$

[R_1] says that it is common knowledge that visual perception is a justified source of knowledge. In other words, it is commonly known that if I see that P , I know P .² [R_2] says that it is commonly known that knowledge requires belief, while [R_3] captures an essential property of common knowledge. Usually common knowledge of a proposition P is taken to mean that everybody knows that P , everybody knows that everybody knows that P , and so on ad infinitum. This is captured by recursive rules that allow us to “unfold” the common-knowledge operator arbitrarily many times. However, this viewpoint is quite problematic for finite knowers of limited cognitive capacity. After three or four levels of nesting, iterated knowledge claims become unintelligible. Because in the present setting we are concerned with cognitive plausibility, we refrain from characterizing common knowledge in the customary strong form, imposing instead limit of three levels of iteration, as indicated in [R_3 .]³ [R_4] is a veracity rule for knowledge.

The following rules can now be readily derived:

$$\frac{\mathbf{C}(P) \quad [DR_1] \quad \frac{\mathbf{C}(P)}{\mathbf{K}(a, P)} \quad [DR_2]}{\mathbf{K}(a_1, \mathbf{K}(a_2, P))}$$

$$\frac{\mathbf{C}(P) \quad [DR_3] \quad \frac{\mathbf{S}(a, P)}{\mathbf{K}(a, P)} \quad [DR_4] \quad \frac{\mathbf{K}(a, P)}{\mathbf{B}(a, P)} \quad [DR_5]}{P}$$

We next have the following three rules:

$$\frac{\mathbf{C}(\mathbf{K}(a, P_1 \Rightarrow P_2) \Rightarrow \mathbf{K}(a, P_1) \Rightarrow \mathbf{K}(a, P_2)) \quad [R_5]}{\mathbf{C}(\mathbf{K}(a, P_1 \Rightarrow P_2) \Rightarrow \mathbf{B}(a, P_1) \Rightarrow \mathbf{B}(a, P_2)) \quad [R_6] \quad \frac{\mathbf{C}(\mathbf{C}(P_1 \Rightarrow P_2) \Rightarrow \mathbf{C}(P_1) \Rightarrow \mathbf{C}(P_2))}{\mathbf{C}(\mathbf{C}(P_1 \Rightarrow P_2) \Rightarrow \mathbf{B}(a, P_1) \Rightarrow \mathbf{B}(a, P_2))} \quad [R_7]}$$

From these we can easily derive the so-called Kripke (“ K ”) rules for knowledge, belief, and common knowledge:

$$\frac{\mathbf{K}(a, P_1 \Rightarrow P_2) \quad \mathbf{K}(a, P_1) \quad [DR_6]}{\mathbf{K}(a, P_2)}$$

We likewise have derived rules [DR_7] and [DR_8] for belief and common knowledge, respectively (omitted here). We also assume that a few straightforward tautologies are common knowledge, and the self-explanatory [R_{11}]:

² We currently ignore the issue of perceptual illusions.

³ Although there is not enough space here for a full discussion, we point out that third-order epistemic and doxastic states (as opposed to n -order for $n > 3$) are often held to be at a level of iteration sufficient for general accounts of human thinking, e.g., see Dennett (1978). This is not to say that fairly realistic scenarios involving iteration of 4 or even 5 levels cannot be devised, but in the present paper we have used 3 for the purpose of modeling the false-belief task.

$$\frac{}{\mathbf{C}((\forall x . P) \Rightarrow P[x \mapsto t])} [R_8] \quad \frac{}{\mathbf{C}([P_1 \Leftrightarrow P_2] \Rightarrow \neg P_2 \Rightarrow \neg P_1)} [R_9]$$

$$\frac{}{\mathbf{C}([P_1 \wedge \dots \wedge P_n \Rightarrow P] \Rightarrow [P_1 \Rightarrow \dots \Rightarrow P_n \Rightarrow P])} [R_{10}] \quad \frac{\mathbf{B}(a, P_1) \quad \mathbf{B}(a, P_2)}{\mathbf{B}(a, P_1 \wedge P_2)} [R_{11}]$$

Note that usually it is postulated that *every* tautology is common knowledge. If we took that as a principle, the presentation of the system could be somewhat simplified. However, such a principle (and other “logical omniscience” principles like it) is wildly implausible, as has often been pointed out. Since we do not accept such unrestricted principles, we only posit certain specific tautologies that are intuitively deemed as obvious. While this is not a general solution, it nevertheless averts the cognitive implausibility of the unrestricted rules, and also serves to isolate the logical knowledge that we need to attribute to agents for a specific reasoning problem.

We can now go on to derive several useful rules, a sample of which is shown below:⁴

$$\frac{\mathbf{K}(a, \forall x . P) \quad [DR_9]}{\mathbf{K}(a, P[x \mapsto t])} \quad \frac{\mathbf{B}(a, \forall x . P) \quad [DR_{10}]}{\mathbf{B}(a, P[x \mapsto t])}$$

$$\frac{\mathbf{C}(\forall x . P) \quad [DR_{11}]}{\mathbf{C}(P[x \mapsto t])} \quad \frac{\mathbf{B}(a_1, \mathbf{K}(a_2, P)) \quad [DR_{12}]}{\mathbf{B}(a_1, \mathbf{B}(a_2, P))}$$

$$\frac{\mathbf{K}(a_1, \mathbf{K}(a_2, [P_1 \wedge \dots \wedge P_n] \Rightarrow P)) \quad \dots \quad \mathbf{K}(a_1, \mathbf{K}(a_2, P_n)) \quad [DR_{17}]}{\mathbf{K}(a_1, \mathbf{K}(a_2, P))}$$

$$\frac{\mathbf{B}(a_1, \mathbf{B}(a_2, [P_1 \wedge \dots \wedge P_n] \Rightarrow P)) \quad \dots \quad \mathbf{B}(a_1, \mathbf{B}(a_2, P_n)) \quad [DR_{18}]}{\mathbf{B}(a_1, \mathbf{B}(a_2, P))}$$

$$\frac{\mathbf{B}(a, P_1 \wedge P_2 \wedge P_3 \Rightarrow P_4) \quad \mathbf{B}(a, P_1) \quad \mathbf{B}(a, P_2) \quad \mathbf{B}(a, P_3) \quad [DR_{19}]}{\mathbf{B}(a, P_4)}$$

The system presented in this section has been implemented in the form of a denotational proof language, as a language similar to the Athena system [2].

3 Encoding the False-Belief Task

False-belief scenarios can be regarded as the drosophila of computational theories of mind. Experiments with false beliefs were first carried out by Wimmer and Perner [12]. In a typical scenario, a child (we will call her Alice) witnesses a character (we will call him Bob) placing an object (say, a cookie) in a certain location l_1 , say in a particular kitchen cabinet. Then Bob leaves, and during his absence someone else (say, Charlie) removes the object from its original location l_1 and puts it in a different location l_2 (say, a kitchen drawer). Alice watches all this transpire, and she is then asked to predict where Bob will look for the object when he gets back, the right answer, of course, being the original location—the cabinet. In this section we show how to formalize this scenario in our calculus. In the next section we will present a formal explanation as to how Alice can come to acquire the correct belief about Bob’s false belief.

⁴ Derivation proofs are omitted, but can be obtained (along with the implementation of the system) by contacting the authors.

We introduce the sort **Location** and the following function symbols specifically for reasoning about the false-belief task:

$$\begin{aligned} places : \text{Object} \times \text{Location} &\rightarrow \text{ActionType} \\ moves : \text{Object} \times \text{Location} \times \text{Location} &\rightarrow \text{ActionType} \\ located : \text{Object} \times \text{Location} &\rightarrow \text{Fluent} \end{aligned}$$

Intuitively, $action(a, places(o, l))$ signifies a 's action of placing object o in location l , while $action(a, moves(o, l_1, l_2))$ is a 's action of moving object o from location l_1 to location l_2 . It is common knowledge that placing o in l initiates the fluent $located(o, l)$:

$$[D_1] \quad \mathbf{C}(\forall a, t, o, l . \text{initiates}(action(a, places(o, l)), located(o, l), t))$$

It is likewise known that if an object o is located at l_1 at a time t , then the act of moving o from l_1 to l_2 results in o being located at l_2 :

$$[D_2] \quad \mathbf{C}(\forall a, t, o, l_1, l_2 . \text{hold}(\text{located}(o, l_1), t) \Rightarrow \text{initiates}(action(a, moves(o, l_1, l_2)), located(o, l_2), t))$$

If, in addition, the new location is different from the old one, the move terminates the fluent $located(o, l_1)$:

$$[D_3] \quad \mathbf{C}(\forall a, t, o, l_1, l_2 . \text{holds}(\text{located}(o, l_1), t) \wedge l_1 \neq l_2 \Rightarrow \text{terminates}(action(a, moves(o, l_1, l_2)), located(o, l_1), t))$$

The following axiom captures the constraint that an object cannot be in more than one place at one time; this is also common knowledge:

$$[D_4] \quad \mathbf{C}(\forall o, t, l_1, l_2 . \text{holds}(\text{located}(o, l_1), t) \wedge \text{holds}(\text{located}(o, l_2), t) \Rightarrow l_1 = l_2)$$

We introduce three time moments that are central to the narrative of the false-belief task: *beginning*, *departure*, and *return*. The first signifies the time point when Bob places the cookie in the cabinet, while *departure* and *return* mark the points when he leaves and comes back, respectively. We assume that it's common knowledge that these three time points are linearly ordered in the obvious manner:

$$[D_5] \quad \mathbf{C}(\text{beginning} < \text{departure} < \text{return}).$$

We also introduce two distinct locations, *cabinet* and *drawer*:

$$[D_6] \quad \mathbf{C}(\text{cabinet} \neq \text{drawer}).$$

Finally, we introduce a domain **Cookie** as a subsort of **Object**, and declare a single element of it, *cookie*. It is a given premise that, in the beginning, Alice sees Bob place the cookie in the cabinet:

$$[D_7] \quad \mathbf{S}(\text{Alice}, \text{happens}(action(\text{Bob}, places(cookie, cabinet)), \text{beginning})).$$

4 Modeling the Reasoning Underlying False-Belief Tasks, and Automating It Via Abstraction

At this point we have enough representational and reasoning machinery in place to infer the correct conclusion from a couple of obvious premises. However, a monolithic derivation of the conclusion from the premises would be unsatisfactory, as

it would not give us a story about how such reasoning can be dynamically put together. Agents must be able to reason about the behavior of other agents efficiently (humans certainly do so). It is not at all obvious how efficiency can be achieved in the absence of mechanisms for abstraction, modularity, and reusability.

We can begin to address both issues by pursuing further the idea of derived inference rules, and by borrowing a page from classic work in cognitive science and production systems. Suppose that we had a mechanism which enabled the derivation of not only *schematic* inference rules, such as the ones that we presented in section 2, but derived inference rules allowing for arbitrary computation and search. We could then formulate *generic* inference rules, capable of being applied to an unbounded (potentially infinite) number of arbitrarily complex concrete situations.

Our system has a notion of *method* that allows for that type of abstraction and encapsulation. Methods are derived inference rules, not just of the schematic kind, but incorporating arbitrary computation and search. They are thus more general than the simple if-then rules of production systems, and more akin to the knowledge sources (or “demons”) of blackboard systems [8]. They can be viewed as encapsulating specialized expertise in deriving certain types of conclusions from certain given information. They can be parameterized over any variables, e.g., arbitrary agents or time points.

A key role in our system is played by an associative data structure (shared by all methods) known as the *assumption base*, which is an efficiently indexed collection of propositions that represent the collective knowledge state at any given moment, including perceptual knowledge. The assumption base is capable of serving as a communication buffer for the various methods. Finally, the control executive is itself a method, which directs the reasoning process incrementally by invoking various methods triggered by the contents of the assumption base.

We describe below three general-purpose methods for reasoning in the calculus we have presented. With these methods, the reasoning for the false-belief task can be performed in a handful of lines—essentially with one invocation of each of these methods. We stress that these methods are not ad hoc or hardwired to false-belief tasks. They are generic, and can be reused in any context that requires reasoning about other minds and satisfies the relevant preconditions. In particular, the methods do not contain or require any information specific to false-belief tasks.

- *Method 1:* This method, which we call M_1 , shows that when an agent a_1 sees an agent a_2 perform some action-type α at some time point t , a_1 knows that a_2 knows that a_2 has carried out α at t . M_1 is parameterized over a_1 , a_2 , α , and t :

1. The starting premise is that a_1 sees a_2 perform α at t :

$$\mathbf{S}(a_1, \text{happens}(\text{action}(a_2, \alpha), t)) \quad (1)$$

2. Therefore, a_1 knows that the corresponding event has occurred at t :

$$\mathbf{K}(a_1, \text{happens}(\text{action}(a_2, \alpha), t)) \quad (2)$$

This follows from the preceding premise and $[DR_4]$.

3. From $[A_4]$ and $[DR_2]$ we obtain:

$$\mathbf{K}(a_1, \forall a, \alpha, t . \text{happens}(\text{action}(a, \alpha), t) \Rightarrow \mathbf{K}(a, \text{happens}(\text{action}(a, \alpha), t))) \quad (3)$$

4. From (3) and $[DR_9]$ we get:

$$\mathbf{K}(a_1, \text{happens}(\text{action}(a_2, \alpha), t) \Rightarrow \mathbf{K}(a_2, \text{happens}(\text{action}(a_2, \alpha), t))) \quad (4)$$

5. From (4), (2), and $[DR_6]$ we get:

$$\mathbf{K}(a_1, \mathbf{K}(a_2, \text{happens}(\text{action}(a_2, \alpha), t))) \quad (5)$$

- *Method 2*: The second method, M_2 , shows that when (1) it is common knowledge that a certain event e initiates a fluent f ; (2) an agent a_1 knows that an agent a_2 knows that e has happened at a time t_1 ; (3) it is commonly known that $t_1 < t_2$; and (4) a_1 knows that a_2 knows that nothing happens between t_1 and t_2 to terminate the fluent f ; then a_1 knows that a_2 knows that f holds at t_2 . M_2 is parameterized over a_1, a_2, e, f, t_1 , and t_2 . We omit the definition due to space limitations (it can be found in the source code).
- *Method 3*: The last method, M_3 , shows that when (1) it is common knowledge that t_1 is prior to t_2 ; (2) an agent a_1 knows that an agent a_2 knows that a fluent f holds at t_1 ; and (3) a_1 believes that nothing happened between t_1 and t_2 that would cause a_2 to believe that f no longer holds; then a_1 believes that a_2 believes that f holds at t_2 :

1. The starting premises are:

- $P_1 : \mathbf{C}(t_1 < t_2); P_2 : \mathbf{K}(a_1, \mathbf{K}(a_2, \text{holds}(f, t_1)))$;

$$P_3 : \mathbf{B}(a_1, \neg \exists e, t. \mathbf{B}(a_2, \text{happens}(e, t)) \wedge \mathbf{B}(a_2, t_1 < t < t_2) \wedge \mathbf{B}(a_2, \text{terminates}(e, f, t)))$$
.

2. From premise P_2 , $[DR_5]$, and $[DR_{12}]$, we get:

$$\mathbf{B}(a_1, \mathbf{B}(a_2, \text{holds}(f, t_1))) \quad (6)$$

3. From $[A_6]$, $[DR_3]$, and universal specialization we get:

$$[\mathbf{B}(a_1, \mathbf{B}(a_2, \text{holds}(f, t_1))) \wedge \mathbf{B}(a_1, \neg \exists e, t . \mathbf{B}(a_2, \text{happens}(e, t)) \wedge \mathbf{B}(a_2, t_1 < t < t_2) \wedge \mathbf{B}(a_2, \text{terminates}(e, f, t))] \Rightarrow \mathbf{B}(a_1, \neg \mathbf{B}(a_2, \text{clipped}(t_1, f, t_2))) \quad (7)$$

4. By P_3 , (7), (6), conjunction introduction, and modus ponens, we get:

$$\mathbf{B}(a_1, \neg \mathbf{B}(a_2, \text{clipped}(t_1, f, t_2))) \quad (8)$$

5. From $[A_5]$, $[DR_{11}]$, and $[DR_2]$ we get:

$$\mathbf{K}(a_1, [\mathbf{B}(a_2, \text{holds}(f, t_1)) \wedge \mathbf{B}(a_2, t_1 < t_2) \wedge \neg \mathbf{B}(a_2, \text{clipped}(t_1, f, t_2))] \Rightarrow \mathbf{B}(a_2, f)) \quad (9)$$

6. From (9) and $[DR_5]$ we get:

$$\mathbf{B}(a_1, [\mathbf{B}(a_2, \text{holds}(f, t_1)) \wedge \mathbf{B}(a_2, t_1 < t_2) \wedge \neg \mathbf{B}(a_2, \text{clipped}(t_1, f, t_2))] \Rightarrow \mathbf{B}(a_2, \text{holds}(f, t_2))) \quad (10)$$

7. From P_1 , $[DR_1]$, $[DR_5]$, and $[DR_{12}]$ we get:

$$\mathbf{B}(a_1, \mathbf{B}(a_2, t_1 < t_2)) \quad (11)$$

8. From (10), (6), (11), (8), and $[DR_{19}]$ we get:

$$\mathbf{B}(a_1, \mathbf{B}(a_2, \text{holds}(f, t_2))) \quad (12)$$

The correct conclusion for the false-belief task, produced by our implementation in a fraction of a second, is now obtained in the following manner:

1. Method M_1 fires, invoked with Alice, Bob, the action type $\text{places(cookie, cabinet)}$, and time point *beginning*.
2. Axiom $[D_1]$ is repeatedly instantiated (via $[DR_{11}]$) with *Bob*, *cookie*, and *cabinet*.
3. Method M_2 fires, invoked with *Alice*, *Bob*, the action that *Bob* has placed the cookie in the cabinet, the fluent that the cookie is located in the cabinet, and the two time points *beginning* and *departure*.
4. Method M_3 fires, invoked with *Alice*, *Bob*, the fluent that the cookie is located in the cabinet, and the two time points *departure* and *return*.

Therefore, the final conclusion is the following: Alice believes that, upon his return, Bob believes that the cookie is located in the cabinet.

5 Related Work and Conclusions

We have presented a formal system for representing and reasoning about certain important types of mental states, and used it to provide a formal analysis of false-belief tasks. Such tasks have been extensively discussed, particularly in the debate between theory-theory and simulation [6], but there are few rigorous models to be found. The only computational treatments of which we are aware are by Bello and Cassimatis [4] and by Watt [14]. Neither is based on a formal inference system. Goodman et al. [10] present a rational analysis of false belief reasoning based on causal Bayesian models.

Technically, our system is a multi-sorted multi-modal first-order logic. There is a growing recognition of the importance of quantification in epistemic contexts. Propositional multi-modal logics are just not sufficiently expressive. For instance, they cannot capture the difference between de dicto and de re knowledge. The versatility of first-order logic is necessary, alongside constructs such as common knowledge.

Our approach has been thoroughly proof-theoretic; we have not given a model-theoretic semantics for our logic. Coming up with an appropriate formal semantics for propositional attitudes is exceedingly difficult, and should not hold back experimentation with and implementation of various proof systems. The usual possible-world semantics [9] are mathematically elegant and well-understood, and they can be a useful tool in certain situations (e.g., in security protocol

analysis). But they are notoriously implausible from a cognitive viewpoint.⁵ The element of justification, for instance, which is central in our intuitive conception of knowledge, is entirely lacking from the formal semantics of epistemic logic. Indeed, knowledge, belief, desire, intention, provability, etc., all receive the exact same formal analysis in possible-world semantics. That is simply not tenable.

At any rate, even in the standard Kripke framework, the question of how to combine quantification with epistemic constructs (particularly with common knowledge) is a difficult open problem: There have been no complete recursive axiomatizations, and indeed such logics are not even recursively enumerable [16]. Some decidable fragments have been investigated, such as the space of monodic formulas [13], but such restrictions limit expressivity, which in our view is a more important consideration. Indeed, we see no reason to insist on a computationally tractable—or even decidable—formalism, or on a complete logic, at the expense of expressivity. First-order logic is undecidable, but it is routinely used for the analysis and verification of a wide variety of extensional systems, by deploying interactive theorem-proving systems. Higher-order logic is both undecidable and incomplete, but it too is used widely for similar purposes. Things need not be different when it comes to the representation, analysis, and verification of rational agents. Our concern here has been to design and implement an expressive logic that can be readily used for such purposes, and to gain experience with constructing machine-checkable proofs in that logic, particularly with writing proof tactics in it.

LORA [17] is a multi-sorted language that extends first-order branching-time temporal logic with modal constructs for beliefs, desires, and intentions (drawing on the seminal work of Cohen and Levesque [5], and particularly on the BDI paradigm that followed it [11]), as well as a dynamic logic for representing and reasoning about actions. It does not have any constructs for perception or for common knowledge, and does not allow for the representation of events that are not actions. Its semantics for the propositional attitudes are standard Kripke semantics, with the possible worlds being themselves branching time structures. We are not aware of any implementations of LORA.

CASL (Cognitive Agents Specification Language) [12] is another system which combines an action theory, defined in terms of the situation calculus, with modal operators for belief, desire, and intention. Like LORA, CASL does not have any constructs for perception or for group knowledge (shared, distributed, or common). Also like LORA, the semantics of all intensional operators in CASL are given in terms of standard possible worlds. They are, in fact, explicitly defined in the higher-order logic PVS by quantifying over states. Insofar as both LORA and CASL base their treatment of intensional operators on Kripke structures, they inherit all the conceptual difficulties associated with them. An advantage of CASL from our viewpoint is that it is implemented and allows for mechanized

⁵ In an apt assessment of the situation, Anderson [1] wrote that epistemic logic “has been a pretty bleak affair.” Fagin et al. [9] describe various attempts to deal with some of the problems arising in a possible-worlds setting, none of which has been widely accepted as satisfactory.

proofs, given in PVS. However, the natural deduction style of our framework is more conducive to defining arbitrary proof tactics.

While preliminary experience with our implementation is encouraging, more work is needed to determine to what extent methods can facilitate reasoning in such a multi-modal first-order system. One issue related to efficiency is parallelism. The current implementation of our system is single-threaded, and is thus unable to realize one of the chief advantages of blackboard systems, namely, the independence of the specialists from one another and the fact that they can run concurrently. (Of course, even with concurrency, the top control mechanism still needs to coordinate the writing into the blackboard, and that can be expected to become a bottleneck in some cases.) We plan to implement a multi-threaded version of the system in the future.

References

1. Anderson, C.A.: The Paradox of the Knower. *Journal of Philosophy* 80(6), 338–355 (1983)
2. Arkoudas, K.: Athena, <http://www.pac.csail.mit.edu/athena>
3. Arkoudas, K., Bringsjord, S.: Metareasoning for multi-agent epistemic logics. In: Leite, J., Torroni, P. (eds.) CLIMA 2004. LNCS (LNAI), vol. 3487, pp. 111–125. Springer, Heidelberg (2005)
4. Bello, P., Bignoli, P., Cassimatis, N.: Attention and Association Explain the Emergence of Reasoning about False Beliefs in Young Children. In: Proceedings of the 8th International Conference on Cognitive Modeling, pp. 169–174 (2007)
5. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. *Artificial Intelligence* 42, 213–261 (1990)
6. Davies, M., Stone, T. (eds.): Folk Psychology: The Theory of Mind Debate. Blackwell Publishers, Malden (1995)
7. Dennett, D.: Conditions of personhood. In: Brainstorms: Philosophical Essays on Mind and Psychology, pp. 267–285. Bradford Books, Montgomery (1978)
8. Engelmore, R., Morgan, T. (eds.): Blackboard Systems. Addison-Wesley, Reading (1988)
9. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about knowledge. MIT Press, Cambridge (1995)
10. Goodman, N.D., Bonawitz, E.B., Baker, C.L., Mansinghka, V.K., Gopnik, A., Wellman, H., Schulz, L., Tenenbaum, J.B.: Intuitive theories of mind: A rational approach to false belief. In: Proceedings of the Twenty-Eight Annual Conference of the Cognitive Science Society (2006)
11. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: Proceedings of Knowledge Representation and Reasoning (KR&R 1991), pp. 473–484 (1999)
12. Shapiro, S., Lespérance, Y., Levesque, H.J.: The cognitive agents specification language and verification environment for multiagent systems. In: The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, pp. 19–26 (2002)
13. Sturm, H., Wolter, F., Zakharyaschev, M.: Common Knowledge and Quantification. *Economic Theory* 19, 157–186 (2002)
14. Watt, S.N.K.: Seeing things as people. PhD thesis, Knowledge Media Institute, Open University, UK (1997)

15. Wimmer, H., Perner, J.: Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. *Cognition* 13, 103–128 (1983)
16. Wolter, F.: First Order Common Knowledge Logics. *Studia Logica* 65, 249–271 (2000)
17. Wooldridge, M.: Rational Agents. MIT Press, Cambridge (2000)

Computing Stable Skeletons with Particle Filters

Xiang Bai¹, Xingwei Yang², Longin Jan Latecki², Yanbo Xu¹, and Wenyu Liu¹

¹ Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China

{xiang.bai,xuyanbohust}@gmail.com, liuwy@hust.edu.cn

² Department of Computer and Information Sciences, Temple University,
Philadelphia, PA 19022, USA

{xingwei.yang,latecki}@temple.edu

Abstract. We present a novel method to obtain high quality skeletons of binary shapes. The obtained skeletons are connected and one pixel thick. They do not require any pruning or any other post-processing. The computation is composed of two major parts. First, a small set of salient contour points is computed. We use Discrete Curve Evolution, but any other robust method could be used. Second, particle filters are used to obtain the skeleton. The main idea is that the particles walk along the skeletal paths between pairs of the salient points. We provide experimental results that clearly demonstrate that the proposed method significantly outperforms other well-known methods for skeleton computation.

Keywords: Skeleton, shape, pruning, skeletal paths, particle filters.

1 Introduction

The skeleton is important for object representation and recognition in different areas, such as image retrieval and computer graphics, character recognition, image processing, and the analysis of biomedical images [1]. The skeleton is an abstraction of objects that at the same time contains both shape features and topological structures of the original object. Therefore, many researchers have worked on matching skeleton structures represented by graphs or trees [2,3,4,5,6]. However, as the skeleton is sensitive to the noise and deformation of the boundary, which may seriously disturb the topology of the skeleton graph, these methods cannot work on complex shapes or shapes with obvious noise.

We list now properties of the ideal skeleton mentioned in [7].

- (1) it should preserve the topology of the original object
- (2) it should be stable under deformations
- (3) it should be invariant under Euclidean transformations such as rotations and translations
- (4) the position of the skeleton should be accurate
- (5) it should be composed of 1D arcs (i.e., one-pixel wide in digital images)
- (6) it should represent significant visual parts of objects

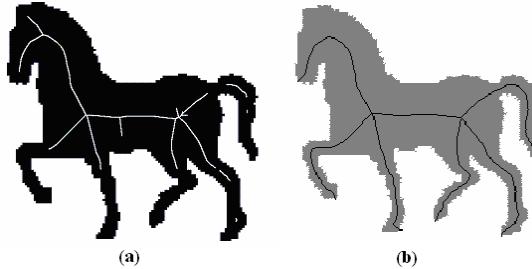


Fig. 1. (a) skeleton computed by the method in [11], (b) by the proposed method

Properties (4) and (5) mean that the skeleton should contain the centers of maximal disks, and nothing more than the centers of maximal disks. Property (6) means that there should be skeleton branches in every significant object part and that there should be no spurious branches that do not correspond to any object parts (which are usually due to noise).

Since most of the existing skeleton computation methods are not able to produce skeletons that satisfy property 6, skeleton pruning is applied. Its goal is to remove spurious branches. Clearly, a pruned skeleton should still have properties (1)-(6), which can be will useful for shape matching and recognition [6,8].

Ogniewicz and Kübler [9] presented a few significant measures for pruning complex Voronoi skeletons without disconnecting the skeletons, but it may lead to topology violation. The method in [10] has difficulty in distinguishing noise from low frequency shape information on boundaries. The skeleton generated by [11] cannot guarantee the property of the connectivity, as shown in the experimental results in Fig. 1(a). The skeleton computed by our method is shown in Fig. 1(b).

The method introduced by Bai et al. [7,12] can obtain excellent skeletons which contain most of the properties of ideal skeletons, but it cannot guarantee that the skeleton is one-pixel wide and it need the postprocessing. Compared to it, our method produces one-pixel thick skeletons without skeleton pruning.

Particle filters estimate the posterior probability density over the state space of a dynamic system. The key idea of this technique is to represent probability densities by sets of samples. By sampling in proportion to likelihood, particle filters focus the computational resources on regions with high likelihood, where good approximations are most important. Over the last few years, particle filters have been applied with great success to a variety of state estimation problems including visual tracking, speech recognition, mobile robot localization, robot map building, people tracking, and fault detection. Moreover, Adluru et al have used particle filters in contour grouping [13]. The proposed method is the first one that utilizes particle filters in computing skeletons.

The proposed method first utilizes the Discrete Curve Evolution (DCE) [14] to simplify the contour, and to obtain a small set of salient points as vertices of the simplified polygon, but other approaches which produce stable salient points could also be used. The basic idea of the DCE is simple. In every evolutional step of DCE, a pair of consecutive line segments s_1, s_2 is replaced by a single

Fig. 2. Hierarchical skeleton of elephant obtained by pruning the input skeleton (left) with respect to contour segments obtained by the Discrete Curve Evolution (DCE). The outer (red) polylines show the corresponding DCE simplified contours.

line segment joining the endpoints of $s_1 \cup s_2$. The order of the substitution is determined by the relevance measure K given by:

$$K(S_1, S_2) = \frac{\beta(S_1, S_2)l(S_1)l(S_2)}{l(S_1)l(S_2)} \quad (1)$$

where line segments s_1, s_2 are the polygon sides incident to a vertex v , $\beta(s_1, s_2)$ is the turn angle at the common vertex of segments s_1, s_2 , l is the length function normalized by the total length of a polygonal curve C . The higher value of $K(s_1, s_2)$, the larger is the contribution of the arc $s_1 \cup s_2$ to the shape. During the evolution, we will first remove the arcs with the smallest contribution. In Fig. 2, we show some results to illustrate that each convex vertex of the DCE simplified polygon is guaranteed to be a skeleton endpoint.

We benefit from a geometric relation between the skeletal path and the contour, which is a key observation that motivates our approach: the endpoints of significant skeleton branches coincide with convex salient contour points. We illustrate the main ideas of the proposed method in Fig. 3. Let a and b be two salient contour points. They divide the contour into two parts $C = C_1 \cup C_2$ marked with red and blue colors, respectively. The skeleton path $p(a, b)$ from a to b is composed of centers of maximal disks that are tangent both to C_1 and to C_2 . We use a particle filter to compute the path $p(a, b)$. The condition that the maximal disks are tangent to two contour parts makes our skeleton insensitive to noise and contour deformations. The computation with particle filters assures that the skeleton paths are connected, vary smoothly, and are one pixel thick.

The final skeleton consists of the skeleton paths between all pairs of salient points. For a given set of salient contour points, we obtain an excellent skeleton without any pruning process. We use DCE to generate salient points, since it is

Fig. 3. In green a single skeleton path $p(a, b)$ from a to b computed by our algorithm. The salient points a and b divide the contour into two parts $C = C_1 \cup C_2$ marked with red and blue colors, respectively. $p(a, b)$ is composed of centers of maximal disks that are tangent both to C_1 and to C_2 .

proved in Bai et al. [7], each DCE computed convex salient point is guaranteed to be a skeleton endpoint.

As in our case the target function is nonlinear, the Dynamic Programming (DP), which can only solve the linear function, will carry contour noise to the skeleton. Compared to DP, the particle filter can get rid of the noise and local solutions. It can allow branching and carrying multiple solutions. Therefore, We use a particle filter to find the skeleton path between any pair of the salient points instead of DP. Particle filters are also known as Sequential Monte-Carlo (SMC) methods, which have the ability to carry multiple hypotheses, and are widely used to track multiple targets with cluttered background in image sequences. The first application of particle filters in Computer Vision is in the tracking of object contours (Isard and Blake [15,16]). Tracking of tracking of motion boundaries is used for motion estimation in [17]. The first application of particle filters to static images is presented in Pérez et al. [18], where particle filters are applied to perform inference over a spatial chain of edge pixels rather than over a temporal chain. An extension of SMC that performs inferences on arbitrarily structured graphical models has been proposed in [19,20] and applied to an edge linking task in [19].

The rest of the paper is organized as follows: our approach to computing skeleton paths is introduced in Section 2. The construction of the whole skeleton is presented in Section 3. The experimental results are shown in Section 4. Finally, the conclusion is presented in Section 5.

2 Computing Skeleton Paths with a Particle Filter

Let a and b be two convex, salient contour points. As stated in the introduction, we use DCE polygon simplification to compute the salient points, since all convex

vertices of the DCE simplified polygon are guaranteed to be skeleton endpoints. Our goal is to obtain a skeleton path from a to b . We use $x_{1:t}^j$ to denote a sequence of skeleton points of particle j at time step t , i.e., $x_{1:t}^j = x_1^j, \dots, x_t^j$. Then x_t^j is the current endpoint of the particle j at the step t . Let $N(x_t^j)$ represent the set of 8-nearest neighbors of all of skeleton points of particle j .

We initialize with n particles, each equal to a , and the initial weights of the particles are $1/n$. At each iteration, we consider eight possible continuations of particle $x_{1:t-1}^j$ as the 8-nearest neighbors of x_{t-1}^j . (Here we benefit from the fact that a digital image is a discrete structure.) We obtain an eight extensions of particle $x_{1:t}^k = \{x_{1:t-1}^j, x_t^k\}$ for each of the eight neighbors $x_t^k \in N(x_{t-1}^j)$. The index k of particle $x_{1:t}^k$ may be different from j , since particle j has 8 extensions corresponding to the 8 neighbors $N(x_{t-1}^j)$ of x_{t-1}^j .

Now we derive a particle filter algorithm that is particularly suitable for computation in digital images. Our goal is to estimate the posterior $p(x_{1:t}|z_{1:t})$ over all potential skeleton paths in a given shape. Our observations $z_{1:t} = \{z_1, z_2, \dots, z_t\}$ represent distances to the shape contour (a detailed definition follows below). Each particle represents a particular skeleton path. We will follow the framework of a particle filter algorithm called sampling importance resampling (SIR) filter [21], which can be summarized as follows:

1) *Prediction by Sampling*: The next generation of particles $\{x_{1:t}^k\}_k$ is obtained from the generation $\{x_{1:t}^j\}_{j=1}^n$ by sampling from a proposal distribution π (defined below).

We use prior boosting in prediction by sampling (Gordon et al., [22]). It allows us to capture multi-modal likelihood regions in the posterior. In prior boosting we sample more than one follower for each particle so that different followers can capture different modes of the proposal. As described above, the fact that we work in digital images naturally suggests the eight followers be the eight neighbors of the latest pixel in each particle sequence. Thus, we increase the number of particles from N to $8N$, which is then reduced back to N in the resampling step (3).

2) *Importance Weighting*: An importance weight is assigned to each particle according to the importance sampling principle $w_t^k = \frac{p(x_{1:t}^k|z_{1:t})}{\pi(x_{1:t}^k|z_{1:t})}$. The weights account for the fact that the proposal distribution is usually not equal to the target distribution $p(x_{1:t}|z_{1:t})$.

3) *Resampling*: Particles are drawn with replacement proportional to their importance weights. The weight of each of the eight new particles is defined as:

$$w_t^k = \frac{p(x_{1:t}^k|z_{1:t})}{\pi(x_{1:t}^k|z_{1:t})} = \frac{\eta p(z_t|x_{1:t}^k, z_{1:t-1}) p(x_t^k|x_{t-1}^j) p(x_{1:t-1}^j|z_{1:t-1})}{\pi(x_t^j|x_{1:t-1}^j, z_{1:t})} \quad (2)$$

$$\propto \frac{p(z_t|x_t^k) p(x_t^k|x_{t-1}^j)}{\pi(x_t^j|x_{1:t-1}^j, z_{1:t})} w_{t-1}^j, \quad (3)$$

where w_{t-1}^j is the weight of particle x_{t-1}^j and $\eta = 1/p(z_t|z_{1:t-1})$ is a normalization factor resulting from Bayes rule that is equal for all particles. Now we make

an important assumption that the proposal distribution $\pi(x_t | x_{1:t-1}^j, z_{1:t})$ is uniform. This is justified in our context by the fact that each point is a pixel that has eight neighbors, and continuation to each of the eight neighbors is equally probable. Therefore, we obtain

$$w_t^k \propto p(z_t | x_t^k) p(x_t^k | x_{t-1}^j) w_{t-1}^j \quad (4)$$

The conditional probabilities in equation (3) are defined below based on digital topology of paths in digital images $p(x_t^k | x_{t-1}^j)$ and on geometric properties of skeletons $p(z_t | x_t^k)$.

The conditional probability of the new particle $x_{1:t}^k$ generated by extending the j th particle is given by:

$$p(x_t^k | x_{1:t-1}^j) = \begin{cases} 1, & \text{if } x_t^k \in N(x_{t-1}^j) - N(x_{1:t-1}^j) \\ 0.01, & \text{else} \end{cases} \quad (5)$$

The main contribution of this probability is to avoid visiting the same pixels again, since we do not want the particle path to go backward, which would create a loop in the skeleton path or perturb it. Hence we assign very low probability to the neighbors of x_{t-1}^j that already belong to the sequence of particle $x_{1:t-1}^j$.

In order to calculate $p(z_t | x_t^k)$, we recall that the contour is divided into two parts C_1 and C_2 . Let d_1, d_2 represent the minimum distance from the point x_t^k to each of the parts., which for a correct skeleton paths both should be equal to the radius of the maximal disk centered at x_t^k . In particular, we should have $d_1 = d_2$. Thus, our observation z_t is composed of two distances d_1, d_2 from the contour parts C_1 and C_2 . Fig. 4 illustrates our computation of $p(z_t | x_t^k)$. Consider two different points P_1 and P_2 as candidates for the skeleton point x_t^k . It is obvious that P_1 is more likely to be the center of a maximal disk with respect to the contour parts C_1 and C_2 than P_2 , since $D' = |d'_1 - d'_2|$ is smaller than $D = |d_1 - d_2|$.Therefore, we assume that the observation density is a Gaussian function of the difference $d_1 - d_2$:

$$p(z_t | x_t^k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(d_1 - d_2)^2}{2\sigma^2}} \quad (6)$$

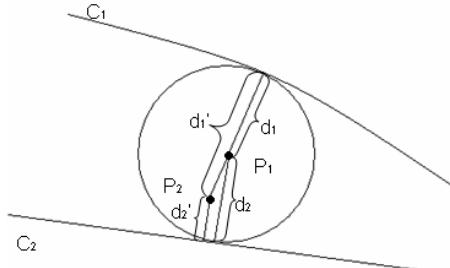


Fig. 4. Point P_1 is more likely to be a skeleton point than point P_2

The outline of the derived particle filter algorithm is as follows:

From the "old" sample set $\{(x_{t-1}^j, w_{t-1}^j) : j = 1, \dots, N\}$ at the time step $t-1$, construct a new sample set $\{(x_t^j, w_t^j) : j = 1, \dots, N\}$ for step t .

For $j = 1$ to N iterate steps (1)-(3):

(1) Prediction by Sampling

For each particle j , we extend it to eight particles by $x_{1:t}^k = \{x_{1:t-1}^j, x_t^k\}$, where $x_t^k \in N(x_{t-1}^j)$.

2) Importance Weighting

Compute the weights $w_t^k = p(z_t | x_t^k) p(x_t^k | x_{t-1}^j) w_{t-1}^j$ and normalize the weights so that $\sum_k w_t^k = 1$.

3) Subsampling

Draw N particles from the current set of $8N$ particles with probabilities proportional to their weights.

Finally, the particle with the highest weights is selected, which represents a skeleton path. There are two important differences in comparison to the standard sampling importance resampling (SIR) filter. First, our prediction by sampling considers all possible extensions to the eight neighbors, this is why our proposal distribution is uniform. Second, since our prediction by sampling increases the number of particles to $8N$, we replaced resampling with subsampling in order to reduce the number of particles to N . We modified the residual resampling to obtain the residual subsampling.

Fig. 3 shows an example of one skeleton path generated by the above algorithm. The blue and red parts represent the two different parts C_1, C_2 of the contour separately, which are divided by the two vertices. The green line is the skeleton path generated by our algorithm. The skeleton path is in the middle of the two contour parts, which is the main property of an excellent skeleton. The skeleton path does not have any redundant branches and it is insensitive to boundary noise. These properties follow from the fact that the observation density $p(z_t | x_t^k)$ is computed with respect to the contour partitions C_1 and C_2 induced by two salient points. The conditional probability $p(x_t^k | x_{t-1}^j)$ is responsible for computing smooth paths that are one pixel thick. The statistical framework of particle filter assures that the local noise on pixel level does not distort the skeleton paths.

3 Combining Skeleton Paths to Form a Complete Skeleton

The skeleton is the combination of skeleton paths between the end nodes. If we have generated one path of the skeleton, the other paths of the skeleton will be generated in the similar way. The only difference is that when the generating skeleton path meets the generated skeleton path, it should stop. This can preserve the property of the one-pixel wide and keep the connectivity of the skeleton. For example, the skeleton of the heart in Fig. 5 (a) is the skeleton of

Fig. 5. The skeleton in (a) is constructed by combining the paths in (b), (c) together

the heart. The skeleton path of Fig. 5 (b) is first generated. Then, instead of combining the whole skeleton path in Fig. 5 (c), the propose approach will only take part of the skeleton path of it, which is surrounded by the red rectangle.

4 Experiments

In this section, we evaluate the proposed method in two parts: 1) we show that the skeleton is stable to noise and deformation and 2) we compare it to other methods. From all of the results listed below, we can state that the proposed approach can generate excellent skeletons which satisfy the six properties listed in Introduction. Besides, according to the comparison experiments, the proposed method can obtain much better skeletons than many other approaches.

4.1 Test on Noisy Images

The results in Fig. 6 show that the proposed method is insensitive to even substantial noise in contours. For each shape, there is one image without noise and one image with substantial noise. The similarity of the obtained skeletons illustrates the stability of the proposed method. In particular, there are no branches generated by the boundary noise, and the skeletons still preserve the topological and geometric structure of the objects. Other methods cannot obtain stable skeletons on noisy images. Most of them will have extra branches or distorted skeletons. The extraordinary stability of our skeletons in the presence of large inner-class shape variations is demonstrated in Fig. 7.

Fig. 6. For each shape, there is one image without noise and one image with substantial noise. The obtained skeletons are very similar.

Fig. 7. The results on some shapes from the MPEG-7 database [14] illustrate extraordinary stability of our skeletons in the presence of large shape variances. The red lines illustrate the DCE polygons.

Although the objects differ significantly from each other, the obtained skeletons have the same global structure. Moreover, the thin tails of the camels remained in the skeleton, which cannot be achieved by most of the other pruning methods, since they may shorten or disconnect the skeleton. The final DCE simplified polygons are also shown overlaid on the shapes with red segments.

4.2 Comparing to Other Methods

We compare our method to the fixed topology method in [23], which also starts with a small set of salient points. However, the fixed topology skeleton requires also that the skeleton junction points are estimated. We do not need to estimate the junction points. Two example results of [23] are shown in Fig. 8(a),(c). As can be clearly seen, the obtained skeleton is not positioned accurately in that many skeleton points are not centers of maximal disks. In contrast, as shown in Fig. 8(b),(d) all of our skeleton points are the centers of maximal disks, and therefore they are exactly symmetrical to the shape boundary. In addition, observe the presence of phantom horizontal skeleton branches in Fig. 8(c). They do not reflect any real structural information. Due to the stability of DCE, the proposed method does not introduce any phantom branches.

Fig. 9 shows a comparison of our approach (b) with the method in [9] (a), which has inaccurate, half-shortened branches that are not related to any obvious

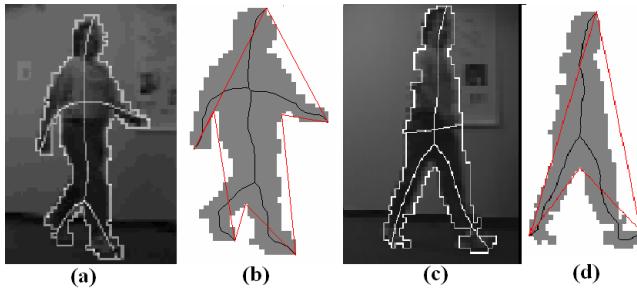


Fig. 8. Comparison between the fixed topology skeleton in [23] in (a), (c) and our skeleton in (b), (d). The red lines illustrate the DCE polygons.

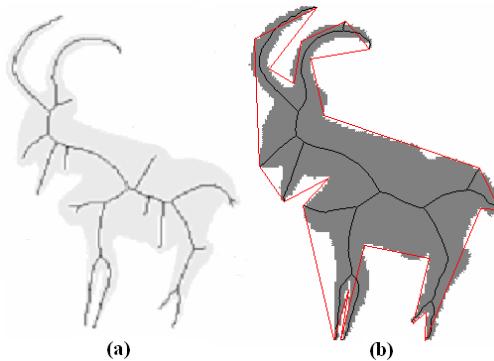


Fig. 9. Comparison between pruning result in [7] in (a) and our results in (b)

boundary features. Other experimental results of the proposed method prove that it is able to completely eliminate all the unimportant branches and still preserve the main structure. Our method does not suffer from the shortening of

Fig. 10. Comparison between pruning result in [7] in (a) and our results in (b)

main skeleton branches and it preserves the topology of the skeleton. Moreover, the obtained skeletons seem to be in accordance with human perception, as it satisfies the six properties of the skeleton.

The method introduced by Bai et al. [7] can obtain excellent skeletons which contain most of the properties of ideal skeletons, but it cannot guarantee that the skeleton is one-pixel wide, which is illustrated in Fig. 10(a). As shown in Fig. 10(b), our method produces one-pixel wide skeletons.

5 Conclusion

In this paper, we establish a novel framework for skeleton computation that combines the geometric method of Discrete Curve Evolution with the statistical method of particle filters. The obtained skeletons do not have redundant skeleton branches and retain all the necessary visual branches. The experimental results demonstrate high stability of the obtained skeletons even for objects with extremely noisy contours, which is the key property required to measure the shape similarity of objects using their skeletons. Moreover, this method can guarantee the skeleton is one-pixel wide. In future, we will extend the proposed approach to generate the skeleton for the shape with holes and 3D shapes, as the particle filter can deal with the condition that the path between two endpoints are not unique.

Acknowledgements

This work was supported by a grant from the Ph.D. Programs Foundation of Ministry of Education of China (No. 20070487028).

References

1. Blum, H.: Biological shape and visual science (part i). *J. Theoretical Biology* 38, 205–287 (1973)
2. Siddiqi, K., Shkoufandeh, A., Dickinson, S., Zucker, S.: Shock graphs and shape matching. In: ICCV, pp. 222–229 (1998)
3. Zhu, S., Yuille, A.: Forms: a flexible object recognition and modeling system. In: ICCV (1995)
4. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. *IEEE Trans. PAMI* 26, 550–571 (2004)
5. Macrini, D., Siddiqi, K., Dickinson, S.: From skeletons to bone graphs: Medial abstraction for object recognition. In: CVPR (2008)
6. Bai, X., Latecki, L.J.: Path similarity skeleton graph matching. *IEEE Trans. PAMI* 30, 1282–1292 (2008)
7. Bai, X., Latecki, L.J., Liu, W.Y.: Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Trans. PAMI* 29, 449–462 (2007)
8. Bai, X., Yang, X.W., Yu, D.G., Latecki, L.J.: Skeleton-based shape classification using path similarity. *Int. Journal of Pattern Recog. and Artif. Intell.* 22, 733–746 (2008)

9. Ogniewicz, R.L., Kübler, O.: Hierachic voronoi skeletons. *Pattern Recognition* 28, 343–359 (1995)
10. Shaked, D., Bruckstein, A.M.: Pruning medial axes. *CVIU* 69, 156–169 (1998)
11. Choi, W.P., Lam, K.M., Siu, W.C.: Extraction of the euclidean skeleton based on a connectivity criterion. *Pattern Recognition* 36, 721–729 (2003)
12. Bai, X., Latecki, L.J.: Discrete skeleton evolution. In: Yuille, A.L., Zhu, S.-C., Cremers, D., Wang, Y. (eds.) *EMMCVPR 2007*. LNCS, vol. 4679, pp. 362–374. Springer, Heidelberg (2007)
13. Adluru, N., Latecki, L.J., Lakämper, R., Young, T., Bai, X., Gross, A.: Contour grouping based on local symmetry. In: *ICCV* (2007)
14. Latecki, L.J., Lakämper, R.: Shape similarity measure based on correspondence of visual parts. *IEEE Trans. PAMI* 22, 1185–1190 (2000)
15. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. In: *ECCV*, pp. 343–356 (1998)
16. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. *IJCV* 29, 5–28 (1998)
17. Black, M.J., Fleet, D.J.: Probabilistic detection and tracking of motion boundaries. *IJCV* 38, 231–245 (2000)
18. Pérez, P., Blake, A., Gangnet, M.: Jetstream: Probabilistic contour extraction with particles. In: *ICCV* (2001)
19. Isard, M.: Pampas: Real-valued graphical models for computer vision. In: *CVPR*, pp. 613–620 (2003)
20. Sudderth, E.B., Ihler, A.T., Freeman, W.T., Willsky, A.S.: Nonparametric belief propagation. In: *CVPR*, pp. 605–612 (2003)
21. Doucet, A., de Freitas, N., Gordon, N.: *Sequential Monte Carlo Methods in Practice*. Springer, New York (2001)
22. Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings-F* 140, 107–113 (1993)
23. Golland, P., Grimson, E.: Fixed topology skeletons. In: *CVPR* (2000)

Using Semantic Web Technologies for the Assessment of Open Questions

Dagoberto Castellanos-Nieves¹, Jesualdo Tomas Fernandez-Breis¹,
Rafael Valencia-Garcia¹, Carlos Cruz², Maria Paz Prendes-Espinosa³,
and Rodrigo Martinez-Bejar⁴

¹ Departamento de Informatica y Sistemas, Universidad de Murcia, Spain

² Departamento de Ciencias de la Computacion e IA, Universidad de Granada, Spain

³ Departamento de Didactica y Organizacion Escolar, Universidad de Murcia, Spain

⁴ Departamento de Ingenieria de la Informacion y las Comunicaciones, Universidad de Murcia, Spain

{dcastellanos, jfernand, valencia, pazprend, rodrigo}@um.es
carloscruz@decsai.ugr.es

Abstract. The use of Semantic Web Technologies in E-learning has turned to be more and more significant in the last years. In this paper, an approach that makes use of semantic web technologies to support the assessment of open questions in e-learning courses is described. The knowledge of the course is represented by means of a domain ontology, which is used to generate semantic annotations for the assessment items and to extract knowledge from the students' answers. This methodology has been applied in different courses and the results are also reported and discussed.

Keywords: E-Learning, Semantic Web, Ontology.

1 Introduction

In [1], a classification of the levels of intellectual behavior during the learning process is presented, identifying six levels in the form of a taxonomy: evaluation, synthesis, analysis, application, understanding and knowledge. Different authors agree on the statement that the higher levels of Bloom taxonomy can only be evaluated through open questions. These are not difficult to design for teachers, but their assessment is problematic. If assessment is based on superficial properties of the answer, such as the presence of important terms, it is easy for students to cheat the evaluator by writing general lines and non-sense contents with the terms the evaluator is looking for.

For an appropriate assessment process, answers must be carefully read, looking for precision, clarity and logic. If teachers have to mark a large number of exams, this becomes an exhausting task. A negative effect of such aspect is that teacher might apply different assessment criteria to different exams or lose concentration due to tiredness and boredom. In this sense, providing mechanisms to support teachers in the assessment of open questions would be beneficial.

The Semantic Web proposes the idea that web contents are defined and linked not only for visualization but for being used by applications[2]. In works such as [3], the

Semantic Web is considered a very promising technology to implement E-learning systems, because it constitutes an environment in which human and software agents can communicate semantically. The use of ontologies in E-learning is commonly accepted, and they have been used for different purposes [4,5]. Hence, Semantic Web technologies may help to provide the aforementioned mechanism for supporting assessment.

In this work, a methodology for evaluating exams in E-Learning settings is presented. This methodology makes use of semantic web technologies, including aspects such as ontological or semantic annotation. This methodology provides the E-learning teaching-learning process with a component able to (semi)automatically evaluate open questions. The ultimate goal of this component is to allow for measuring academic performance and being the basis for providing semantic feedback mechanisms. The structure of this paper is described next. In Section 2, the methodology for evaluating courses in E-Learning settings is explained. An example is shown in Section 3. The validation in real settings is presented in Section 4. Finally, some conclusions are put forward in Section 5.

2 The Methodology

The central concept of the approach is the course, which is associated with some teachers and some students. Every single course can have different exams associated, each being designed by a teacher and comprised of a set of questions. Questions are then designed and created by a teacher, who associates a set of annotations to the questions. These annotations constitute the semantic representation of the expected answer for that question. Moreover, the students answers to those questions also generate a set of semantic annotations. Then, the student mark can be calculated by comparing both sets of annotations.

The knowledge is represented in this methodology by using ontologies. An ontology represents a common, shareable and reusable view of a particular application domain, and they give meaning to information structures that are exchanged by information systems [6]. An ontology can be seen as a semantic model containing concepts, their properties, interconceptual relations, and axioms related to the previous elements.

The Ontology Web Language (OWL) (<http://www.w3.org/TR/owl-ref/>) has been used to implement such ontologies. OWL is the current recommendation of the W3C for exchanging semantic content on the web. OWL-DL has been the OWL flavor used due to its expressivity and complete reasoning. On the one hand, the knowledge of the courses domain has also been modeled by an ontology, the called OeLE ontology (Ontology-based eLearning Evaluation), which is accessible at the project website (<http://klt.inf.um.es/~oele>). This ontology defines the major entities in our domain: course, exam, question, annotation, teacher and student. On the other hand, since the course is the basic entity, the knowledge of the academic courses has to be modeled. This model would account for the knowledge the students should acquire through the course. The course ontology is used in the methodology as the knowledge reference for annotating questions and answers, so providing the context for the marking process. The construction of such ontology is out of the scope of this work.

The proper methodology embraces the design, processing and assessment of open questions-based exams or tests. This methodology allows for performing the following

activities: (1) design of semantically annotated questions; (2) design of exams; (3) student answers' processing; and (4) student answers' marking. In the following subsections these steps are described.

2.1 Designing Questions

The methodology makes use of closed and open questions. A closed question is one with a set of possible answers, among which one is correct. Given that the evaluation of these questions is straightforward, there is no need for creating semantic annotations for them. An open question is one with an answer in natural language, so the correction requires some processing. In this case, the teacher needs to write and annotate the question. For instance, given the open question *In the design of materials, differentiated attention must be paid to pedagogical and technical design. What are the aspects to take into account for each type of design?* the OWL representation for this is shown in Example 1.

Example 1. Partial OWL representation for an open question

```
<open_question rdf:id="question_2">
  <text_description>
    In the design of materials, differentiated attention must be paid to
    pedagogical and technical design. What are the aspects to take
    into account for each type of design?
  </text_description>
  <value>10.0</value>
  <question_id>2</question_id>
  <expected_answer> ....the user interface and the coding system are essential elements of tech-
  nical design </expected_answer>
  <question_created_by_teacher rdf:resource="#teacher_1"/>
</open_question>
```

The open question definition is not complete yet, because evaluation-related information has to be added for further steps. When an open question is created by a teacher, then the expected response must be annotated with respect to the course ontology. So, each open question has a set of annotations associated. The types of annotations are: (1) concepts, which represent the main domain entities (i.e., student); (2) attributes, which represent properties of concepts (i.e., the login of a student); and (3) relations, which establish semantic links between two concepts (i.e., a student takes a course). The annotation properties of a particular ontological category for a question annotation differ from those associated to an answer annotation. The latter ones are considered further in this paper.

A question annotation is defined by the annotation for the knowledge entity (concept, attribute, relation or value) of the course ontology; and the quantitative score (1-10) associated to the question. The quantitative value stands for the importance of the knowledge entity in the context of the individual question. Example 2 shows the annotation corresponding to the relation *coding system is part of the technical design*, whose importance is 4. In this example, the linguistic expression *the coding system is essential element of technical design* is associated to the relation *coding_system_part_of_technical_design*. This is already an existing relation in the course ontology, which is referred by *&course*.

Example 2. Partial OWL representation of the annotation of a relation identified in the expected answer

```
<Question_relation_annotation rdf:ID="Question_relation_annotation_2">
  <relation rdf:resource="#course;coding_system_part_of_technical_design">
    the coding system is essential element of technical design</relation>
  <quantitative_value rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
    4
  </quantitative_value>
</Question_relation_annotation>
```

2.2 Designing Exams

Once the questions have been designed, exams can be defined by selecting the questions to be included in the exam. An exam is comprised of an identifier and a list of questions. Each question is created by a person, has a value in the context of the exam, and has a description. The OWL representation of an exam is shown in Example 3.

Example 3. Partial OWL representation of an Exam

```
<exam rdf:ID="content_design_exam_1">
  <questions>
    <open_question rdf:ID="question_1">
      <question_id>3</question_id>
      <question_created_by_teacher
        rdf:resource="#DAGOBERTO_CASTELLANOS"/>
      <value>30.0</value>
      <text_description>
        What aspects have to be considered in the design of
        contents with New Technologies?
      </text_description>
    </open_question>
  </questions>
  <course_exam rdf:resource="#content_design_course"/>
</exam>
```

2.3 Obtaining Semantic Annotations from Students' Answers

Once the exam has been built, the students can do it. As a result, an exam answer is obtained. The exam answer is defined as a set of pairs of questions and answers to such questions. Each answer is comprised of the natural language text and the set of semantic annotations for such answer. The answer annotation for the relation drawn from *pedagogical theme is a component of the pedagogical design* is shown in Example 4.

Example 4. Partial OWL annotation of a student answer

```
<Answer_relation_annotation rdf:ID="Answer_relation_annotation_6">
  <relation rdf:resource="#course;theme_component_design">
    coding system is part of the technical design
  </relation>
  <linguistic_expression rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    pedagogical theme is a component of the pedagogical design
  </linguistic_expression>
</Answer_relation_annotation>
```

Ideally, the answers annotation should be automatically performed, so we have included an algorithm for automatic detection of ontological elements using NLP Technologies. This algorithm has three sequential phases based on the work presented [7].

2.4 Marking Students' Answers

In this section, the method to combine the previous elements for evaluating open questions is described. Different evaluation policies can be defined by combining such elements in different ways. Our intention was to define a flexible evaluation methodology which can be customized in different aspects. Concerning this issue, OWL provides primitives to define similarity between classes (e.g., owl:differentFrom, owl:disjointWith, and, owl:equivalentClass), which are helpful for our purpose, as it will be shown later.

To mark an answer, the following elements are required: (1) the course ontology in OWL; (2) the expected response to a particular open question; and (3) the student response to that open question. The expected response and the student response are comprised of a set of questions and answers annotations, as described in previous sections.

Now, both annotations structures can be compared in order to calculate their similarity. This approach compares ontological entities of the same category, that is, concepts-concepts, relations-relations, and attributes-attributes. In literature, different similarity measurement approaches can be found (see for instance [8,9]).

In order to design our evaluation functions, different elements from these methodologies have been combined. Each individual similarity function has been designed to return a value in the interval [0,1].

Given our objective of designing a flexible evaluation methodology, the evaluation functions contain some configuration parameters. One of these parameters is a threshold that allows for establishing the minimum acceptable similarity value. A threshold value 1 would represent the strictest evaluation so the students' answer must be identical to the expected, otherwise they would get no mark for their answer. The lower the threshold value is, the lower the strictness of the evaluation process is.

Moreover, the methodology allows for taking two policies when the similarity is greater than the threshold. On the one hand, this part of the answer can completely be considered correct, so that the student gets all the score associated to the particular expected knowledge item (non strict policy), or it can get a value proportional to the similarity (strict policy). Intuitively, the evaluation of a question is the sum of the evaluation of concepts, attributes and relations. Let us define from this viewpoint the score obtained by a student for all the concepts, attributes, and relations included in the student's answer. Given an expected answer P, a student's answer E, a threshold S, and a policy M, the concepts evaluation, written *concEval*, is calculated as shown in Equation 1.

$$\text{concEval}(P, E, S, M) = \sum \alpha_i \frac{\text{value}(P_i)}{\text{totalvalue}} \quad (1)$$

$$\forall i = 1..|P| \beta_i = \max_{j=1..|E|} \{\text{concSim}(P_i, E_j)\}$$

$$\forall i = 1..|P|\alpha_i = \begin{cases} \delta_i, \beta_i \geq S \\ 0, \beta_i < S \end{cases}$$

$$\forall i = 1..|P|\delta_i = \begin{cases} \beta_i, M = strict \\ 1, M = nonstrict \end{cases}$$

where $\text{value}(P_i)$ is the quantitative value of the concept P_i in the question and total_value is $\sum \text{value}(P_i)$.

For each concept in P, its similarity with all the concepts in E is calculated and the highest similarity β_i is compared with S. If $\beta_i \geq S$, then the student gets marks for this item. The amount of marks will depend on the strictness of the evaluation process, defined by M. For relations and attributes, the process is similar. Now, the global evaluation function, written *globEval*, is then defined by Equation 2.

$$\begin{aligned} \text{globEval}(P, E, S, M) = & \quad (2) \\ & \text{concEval}(\text{conc}(P), \text{conc}(E), S, M) + \\ & \text{relEval}(\text{rel}(P), \text{rel}(E), S, M) + \\ & \text{attEval}(\text{att}(P), \text{att}(E), S, M) \end{aligned}$$

Where $\text{conc}(A)$, $\text{att}(A)$, and $\text{rel}(A)$ stand for the sets of concepts, attributes and relations contained in A. Next, the functions used to calculate the similarity between concepts, relations and attributes are shown.

2.4.1 Concepts

Provided that the course ontology is used for annotating the expected and the students' response, the ontological entities to compare belongs to the same ontology. This similarity function, written *concSim*, takes into account the linguistic similarity, the attributes and the relations, is defined by Equation 3.

$$\begin{aligned} \text{concSim}(c_i, c_j) = & cp_1 * \text{concProx}(c_i, c_j) + \quad (3) \\ & cp_2 * \text{propSim}(c_i, c_j) + \\ & cp_3 * \text{eqName}(\text{term}(c_i), \text{term}(c_j)) \end{aligned}$$

where $\sum cp_i = 1$ and $0 \leq cp_i \leq 1$.

Hence, the conceptual similarity is calculated as the weighted average of the conceptual proximity (*concProx*), the linguistic similarity between the involved terms (*eqName*), and how similar the set of properties of each concept are (*propSim*). Let us define now these three functions. The conceptual proximity calculates how near the concepts are placed in the ontology through the ontological structure. Here, nodes stand for the amount of concepts between c_i and c_j through the shortest common path, that is, through the closest common taxonomic parent concept. In case there is no common parent, the proximity returns 0. The function is defined by Equation 4.

$$concProx(c_i, c_j) = \begin{cases} 1 - \frac{distance(c_i, c_j)}{totalNodes} & (anc(c_i) \cup \{c_i\}) \cap (anc(c_j) \cup \{c_j\}) \\ 0 & otherwise \end{cases} \quad (4)$$

where *total_nodes* stands for the total amount of concepts in the ontology, and *anc(c)* returns the taxonomic parents of the concept *c*.

In order to calculate linguistic distances, a linguistic distance function (i.e., Hamming, Levenshtein or similar) is applied to get the linguistic similarity between the terms associated to two concepts. The approach makes use of the Levenshtein distance.

Finally, *propSim* accounts for the similarity between the sets of properties associated to the respective concept (see Equation 5). It is calculated by using the similarity measurement presented in related word (see for instance,[9,10,11,8]).

$$propSim(c_i, c_j) = \frac{|common(c_i, c_j)|}{|common(c_i, c_j)| + \beta_1 * nonCommon(c_i, c_j) + \beta_2 * nonCommon(c_j, c_i)} \quad (5)$$

The factor *common(c_i, c_j)* refers to the amount of properties both concepts share and *nonCommon(c_i, c_j)* is analogous, but considering the set of attributes and relations which do not appear in both concepts.

2.4.2 Attributes

The similarity between two attributes, written *attSim*, is calculated by using three factors: (1) the linguistic similarity; (2) the similarity of the concepts they refer to; and (3) the similarity of their values sets. These elements are combined in Equation 6.

$$attSim(a_i, a_j) = at_1 * eqlName(term(a_i), term(a_j)) + \\ at_2 * valSim(a_i, a_j) + \\ at_3 * concSim(concept(a_i), concept(a_j)) \quad (6)$$

where $\sum at_i = 1$ and $0 \leq at_i \leq 1$.

The first and the third factors have already been described in the concept section. The second factor, written *valSim*, calculates the similarity among values sets as defined in Equation 7.

$$valSim(a_i, a_j) = \frac{|values(a_i) \cap values(a_j)|}{min_{k=i,j} |\{values(a_k)\}|} \quad (7)$$

2.4.3 Relations

The similarity between two relations, written *relSim*, is calculated by using two factors: (1) the linguistic similarity between the relations; and (2) the similarity between their participants. Both factors are combined in Equation 8.

$$\begin{aligned}
 relSim(r_i, r_j) = & rl_1 * eqlName(term(r_i), term(r_j)) + \\
 & rl_2 * concSim(r_i.concept1, r_j.concept1) \\
 & *concSim(r_i.concept2, r_j.concept2)
 \end{aligned} \tag{8}$$

where $\sum rl_i = 1$ and $0 \leq rl_i \leq 1$ and $r_i.concept_j$ stands for the j-th concept associated to the i-th relation.

3 Example

An example of the marking methodology is shown next. The starting point will be the annotation sets for the expected answer and the student's answer:

- Expected answer:
 {Concept₁(pedagogical design), Attribute₁(user interface.simplicity),
 Relation₁(pedagogical design has component content design) }
- Student's answer:
 { Concept₂(pedagogical lines), Attribute₂(user interface.clarity),
 Relation₂(pedagogical design has component pedagogical lines) }

The complete definition of the concepts, attributes and relations can be found in the course ontology, which is accessible at <http://klt.inf.um.es/~oele>. The set of parameters used for the evaluation are: cp₁ =0.5, cp₂ =0.4, cp₃ =0.1; at₁=0.3, at₂=0.2, at₃=0.5, rl₁=0.3, rl₂=0.7, threshold=0.8, and method=non strict.

Hence, the following operations need to be performed. First, the similarity between concepts, attributes and relations have to be calculated. Table 1 displays the results for the similarity between the concepts, attributes and relations. For each cell, the results have already been multiplied by the corresponding parameters, so the value 0.4855 for ConcProx stands for $0.5 * concProx$.

The similarity between the concepts, attributes and relations is, respectively, 0.493, 0.5106 and 0.4273. It should be noted that the linguistic similarities have been calculated between the terms in Spanish. Provided that the threshold is 0.8 and the three

Table 1. Results for the similarity functions for this example

Similarity		C ₁ ,C ₂	A ₁ ,A ₂	R ₁ ,R ₂
Concept	ConcProx	0.4855		
	PropSim	0		
	EqName	0.0071		
	ConcSim	0.493		
Attributes	EqName		0.009	
	ValSim		0.0111	
	ConcSim		0.4905	
	AttSim		0.5106	
Relations	EqName			0.0123
	ConcSim			0.415
	RelSim			0.4273

values are lower, the student would get no marks for these knowledge items. Therefore, the student would get 0 points in this question.

If the threshold was 0.5, then the student would get points for the attribute. If the value assigned to the expected attribute was, for instance, 30% of the total of the question, provided that we are using the non-strict method, the student would get 30% in this question.

In case the strict method is used, the student would get 15.3%. This is the result of weighting the value of the question (30%) with the similarity between the attributes(0.5106).

4 Validation in Real Settings

This methodology has been applied in different learning contexts by using the OeLE platform [12]. This platform aims at supporting the design and evaluation of assessments tests in E-learning courses. It contains a module allowing for the semantic annotation of the contents with respect to the course ontology. These annotations are used for the evaluation of the students' answers. The exams are generated and published as Java Server Pages. The students can use this system to solve and submit the exams.

The validation has been carried out in E-learning and traditional courses. The teachers of the courses built the course ontologies, defined and annotated questions, created exams, the students made them, and these exams were marked in different ways: manually by the teacher(s), and automatically by applying the methodology.

The annotation of the students' answers were semiautomatic, since the NLP algorithm used has incremental learning capabilities. The analysis of the experiments was performed by using ANOVA models with repeated measurements, which allow for studying the effect of one or more factors when there is at least one intra-subject factor.

4.1 Design and Evaluation of Didactic Media

The first experiment was carried out in the E-Learning course *Design and evaluation of didactic media* in the Faculty of Education of the University of Murcia in 2006. Fourteen students took this course. The exam had four open questions, and the students' answers were manually evaluated by two teachers and automatically by the methodology.

Statistically, the mark was the dependent variable. The factor method was defined, having three levels (manual₁, methodology, and manual₂). Our hypothesis was the equality of the average for each level. Univariate analysis methods have been applied to the method factor, obtaining the same results than with a multivariate analysis. The *Sig.* value is greater than 0.05, so the hypothesis cannot be rejected, so it can be said that the manual method and the methodology produce the same students' marks.

4.2 Multimedia Systems and Graphical Interaction

The second experiment was done in the context of the subject *Multimedia Systems and Graphical Interaction* during the years 2005, 2006 and 2007.

Two common open questions were included in the exams of this subject, and 63 exams were processed in this experiment, belonging to different years. These questions

had a different value in the context of the exam, so only the assessment as individual questions was targeted in this experiment. The answers to the questions were both manually and automatically evaluated.

A similar statistical approach was used, using ANOVA for the factor method, with two levels: manual, and methodology. Similar hypotheses were formulated. The application of the Mauchly's Test revealed the following results. The *Sig* value for the method factor is 0.475 (>0.05), so the hypothesis cannot be rejected. This means that both methods obtain similar results.

4.3 Validation Remarks

In summary, the methodology has been applied in two different courses. The first one is an E-learning course and the methodology was applied one year. The second one is a traditional course and the methodology has been applied for three years.

The experiments reveal that the marks obtained by the student when the exam is evaluated by the teachers do not differ from the automatic, so the methodology can be qualified as useful for supporting assessment processes in educative contexts. The course ontologies used in these experiments and more details of these validation experiments can be found at the project website.

5 Discussion and Conclusions

Previous works have proposed approaches to the assessment of students from a semantic perspective. For instance, in [13], a learning and assessment system based on the writing of course hyperbooks and the comparison of domain ontologies is presented. There, each group of students can make its own hyperbook from a course ontology and the different hyperbooks are compared and discussed collaboratively. The student's knowledge is manually built by the students, whereas our approach attempts to make this process (semi)automatic. On the other hand, the Atenea platform [14] is also based on natural language processing and statistical techniques to process students' answers in natural language. However, this approach does not make use of Semantic Web technologies.

One of the most important aspects of the approach presented here is the evaluation function. One of our goals was to design a very flexible function. This has been achieved by defining the customizable set of parameters: (1) three for calculating the similarity between concepts; (2) three for the similarity between attributes; (3) two for the similarity between relations; (4) the threshold; and (5) the evaluation policy. The groups 1, 2, and 3 constitute the internal flexibility of the evaluation function, since they allow the teacher to grade the importance of conceptual, linguistic or properties-based similarity between the different entities. The groups 4 and 5 are the severity instruments for teachers.

Regarding the method of weighting each factor, the following should be said. First, there is no standard nor automatic way to determine the best values for the weights, so an analysis has been carried out in order to suggest the potentially best range values for the weights. Most of the functions have a factor which depends on the linguistic similarity.

We consider that the weight for this factor should be low (e.g. 0,1), because it does not provide information about the particular structure or meaning of the knowledge entity. However, this weight can differ between different exams for the same course.

For the rest of parameters, local decisions should be made due to the local nature of their meaning, combining the context (e.g., cp_1 or at_3) and the internal structure (e.g., cp_2 or at_2).

In the experiments shown in this paper, the same combination of parameters was used. In this approach, the lecturer defines the marking strategy, because (s)he can set the values for the parameters and the threshold. So, the marking process can become more or less severe. Part of our current work is oriented to learn the parameters for particular subjects and teachers from past courses so that automatic configurations can be provided.

Acknowledgments

This work has been possible thanks to the Seneca Foundation through project 05738/PI/07 and the Comunidad Autonoma of the Region de Murcia through project TIC-INF 07/01-0001.

References

1. Bloom, B.S.: Taxonomy of educational objectives: The classification of educational goals. In: Handbook I, cognitive domain. Longmans, Green, New York (1956)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American*, 29–37 (May 2001)
3. Stojanovic, L., Staab, S., Studer, R.: elearning based on the semantic web (2001)
4. Zimmermann, K., Mimkes, J., Kamke, H.-U.: An ontology framework for e-learning in the knowledge society. In: International ISKO Conference (2006)
5. Lytras, M.D., Pouloudi, A., Korfiatis, N.: An ontological oriented approach on e-learning. integrating semantics for adaptive e-learning systems. In: ECIS (2003)
6. Brewster, C., O'Hara, K., Fuller, S., Wilks, Y., Franconi, E., Musen, M.A., Ellman, J., Shum, S.B.: Knowledge representation with ontologies: The present and future. *IEEE Intelligent Systems* 19(1), 72–81 (2004)
7. Valencia-Garcia, R., Castellanos-Nieves, D., Fernandez-Breis, J., Vivancos-Vicente, P.: A methodology for extracting ontological knowledge from spanish documents. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 71–80. Springer, Heidelberg (2006)
8. Rodríguez, M.A., Egenhofer, M.J.: Determining semantic similarity among entity classes from different ontologies. *IEEE Trans. Knowl. Data Eng.* 15(2), 442–456 (2003)
9. Euzenat, J., Valtchev, P.: Similarity-based ontology alignment in owl-lite. In: ECAI, pp. 333–337 (2004)
10. Maedche, A., Staab, S.: Measuring similarity between ontologies. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 251–263. Springer, Heidelberg (2002)
11. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res (JAIR)* 11, 95–130 (1999)
12. Castellanos-Nieves, D., Fernandez-Breis, J., Valencia-Garcia, R., Martinez-Bejar, R.: A semantic web technologies-based system for students assessment in e-learning environments. In: IADIS E-Learning Conference (2007)

13. Falquet, G., Mottaz-Jiang, C.-L., Ziswiler, J.-C.: Ontology based interfaces to access a library of virtual hyperbooks. In: Heery, R., Lyon, L. (eds.) ECDL 2004, vol. 3232, pp. 99–110. Springer, Heidelberg (2004)
14. Alfonseca, E., Pérez, D.: Automatic assessment of open ended questions with a BLEU-inspired algorithm and shallow NLP. In: Vicedo, J.L., Martínez-Barco, P., Muñoz, R., Saiz Noeda, M. (eds.) EsTAL 2004. LNCS (LNAI), vol. 3230, pp. 25–35. Springer, Heidelberg (2004)

Quantifying Commitment

Timothy William Cleaver and Abdul Sattar

Institute for Integrated and Intelligent Systems (IIIS)

Griffith University

PMB 50, GCMC 9726, Australia

Tim.Cleaver@student.griffith.edu.au, a.sattar@griffith.edu.au

Abstract. Bratman argued that the purpose of intentions are to constrain the complexity of future decisions. However, the role of intentions in making future commitments is not well understood. In this paper, we propose a methodology to understand how an agent's commitment to its intentions relates to their dynamics. We quantify commitment through a ranking of intentions determined by their use in making decisions. While previous studies on intention revision hinge on changing beliefs, the current study investigates how changes in intention structure motivates further revision. We propose techniques for updating commitment in light of changed intention structures and in so doing identify new situations in which intention revision is rational.

Keywords: Agents, Knowledge Representation, Reasoning.

1 Introduction

Intention has been a widely studied property of agent systems[1,2,3,4,5,6] since its connection to successful resource bound practical reasoning was recognized [7]. However, a full characterization of the relationship between intentions and the commitment made towards these remains an open question. In this paper we provide an initial step in this direction by outlining a theory based on the intuition that the commitment an agent has towards an intention is proportional to the potential cost of dropping it. This cost is derived from the dependencies between intentions that arise as a consequence of decision making.

Based on this theory of intention we then examine its role in intention revision. Intention revision poses two problems: When is it appropriate to reconsider an existing intention? and, How is the replacement selected? We will concern ourselves only with the former. An optimal intention reconsideration strategy grounds the agent's behavior in its environment while minimizing the reasoning required to do so. If an agent reconsiders its intentions too infrequently then its behavior can diverge from that required by the environment. If it reconsiders its intentions too often then its behavior may become undirected, inefficient and opportunities for action may be missed. The structure of an agent's intentions and its commitments to achieve these greatly influence the nature of a rational policy an agent adopts. By quantifying commitment we allow an agent to reason about its preferences to retain its intentions in circumstances in which it may revise them.

2 Intention Dynamics

2.1 Adopting Intentions

Example 1. Imagine an office assistant robot whose jobs include fetching coffee, delivering hard-copies of documents, vacuuming carpets and guiding office guests. One of these robots is unoccupied when it receives a request for coffee. Since the robot is otherwise unoccupied it chooses to deliver the coffee. The robot has two means of satisfying this opportunity: deliver the coffee itself, or cooperate with another robot to do so. The robot chooses to deliver the coffee itself resulting in a commitment towards this behavior. Subsequently, a request to relocate a document is made of the robot. The robot has two means of satisfying this request: deliver the document itself, or, request the services of another agent. Because the robot is already committed to delivering the coffee and the document requires delivery to a distant area of the building, it cannot compatibly deliver both the coffee and the document. Therefore, it negotiates with another agent to jointly handle the task instead. Because the robot's commitment towards delivering coffee influenced its decision to request the services of another robot to handle the document delivery request, there is a dependency between the two commitments.

Every time an intention plays an active role in a decision an agent should increase its commitment towards this intention. Should the agent retract this intention at some future point, doing so would contribute to the reasons why the agent should reconsider decisions in which it affected the outcome. For an existing intentional structure to play an active role in a particular decision it must invalidate one or more of the options over which the agent is deliberating. However, for each option that is invalidated by the agent's current intentions, multiple intentions may invalidate it. Thus, there is a potentially many-to-many relationship between options and the elements of the intention structure that invalidate them. The agent should, therefore, distribute the cost of such a decision to each intention in a way that is proportional to its contribution.

Imagine an agent is adopting an intention to satisfy some need. The agent has a set of options (O) for satisfying this need and the selected option must be compatible with the behavioral background (I). The agent must check each element $o \in O$ for consistency against all intention structures $i \in I$. Assume that a subset of O is found to be incompatible with at least one element of the intention structure using its conflict recognition function¹.

$$O \setminus I = \{o \in O \mid \exists i. i \in I \wedge \text{conflict}(o, i) = \top\} \quad (1)$$

Because each option may be inconsistent with multiple intentions and each intention may be inconsistent with multiple options, we require both the sets of options in conflict with a given intention, and the intentions in conflict with a given option. For each element ($c \in O \setminus I$) in the conflict set, the set of intention structures ($i \in I$) that are incompatible with it is defined as:

$$I \setminus c = \{i \in I \mid \text{conflict}(c, i) = \top\} \quad (2)$$

¹ The conflict function maps an option and an intention to true (\top) when they are in conflict, and to false (\perp) otherwise.

and conversely, the set:

$$(O \setminus I) \setminus i = \{c \in O \setminus I \mid \text{conflict}(c, i) = \top\} \quad (3)$$

is the set of conflicted options ($c \in O \setminus I$) that are invalidated by a given element i of the behavioral background. Assume that an agent selects s from the compatible options available to it ($O - O \setminus I$). The agent has $\frac{|O \setminus I|}{|O|}$ of the cost of intending s to distribute among the intentions involved in the decision, provided there were any. Thus the initial commitment towards intending s is:

Definition 1

$$\text{base}(s) = \begin{cases} \text{if } O \setminus I = \emptyset \text{ cost}(s) \\ \text{else } \left(1 - \frac{|O \setminus I|}{|O|}\right) \times \text{cost}(s) \end{cases}$$

where:

$\text{base}(s)$: is the initial commitment associated with the newly adopted intention s .

$O \setminus I$: is the set of options where each element is in conflict with at least one element of the agent's intentions.

O : is the initial set of options available to satisfy need n .

$\text{cost}(s)$: is the initial cost of adopting s . An example and further discussion are provided in section Intention Cost (2.3, 59).

The weight of the dependency between each intention ($\{i \in I \mid \text{conflict}(s, i) = \top\}$) involved in the decision to intend s is:

Definition 2

$$\text{weight}(i, s) = \frac{|(O \setminus I) \setminus i| \times |O \setminus I|}{|O| \times \sum_{c \in O \setminus I} |I \setminus c|}$$

where:

s : is the option selected to satisfy need n .

$\text{weight}(i, s)$: is the weight assigned to the dependency between intentional structure i and the option selected s .

$|(O \setminus I) \setminus i|$: is the number of options ($c \in O \setminus I$) in the conflict set that intentional background element i is in conflict with.

$|O \setminus I|$: is the number of options that have been invalidated by an element of the background.

$|O|$: is the number of options for achieving s among which the agent is deliberating.

$|I \setminus c|$: is the number of intentional structures that are in conflict with the option $c \in O \setminus I$.

The weight of the dependency between s and each $i \in O \setminus I$ is the proportion of the cost inherent in the adoption of s attributable to i . This is equivalent to the contribution of i to the selection of s over the agent's other options. $\frac{|O \setminus I|}{|O|}$ is the proportion of the commitment towards s that the agent should distribute among all the options that were

invalidated by the intentional structure. For each element of the intentional background $i \in I$ that is in conflict with a particular option ($c \in O \setminus I$) there are $|I \setminus c|$ elements to distribute the commitment between. Thus the total number of intentional structures used to invalidate all the options in $O \setminus I$ is the sum of the number of intentional structures involved in invalidating each option in $O \setminus I$ ². This gives us the amount of commitment to s to distribute to each intentional structure that contributed to the selection of s . However, each $i \in I$ may have contributed to invalidating multiple options in the conflict set. Thus we need to multiply the above by the number of elements in the conflict set that a particular intentional structure invalidated $|O \setminus I \setminus i|$.

We justify the initial setting of $\text{base}(s)$ as the commitment to the result of the means selection process by observing that this is the commitment remaining after assigning the appropriate portions to the elements that played an active role in the decision process. After invalidating some of the alternatives, for this option to be selected it must be based on its merits relative to the other remaining valid options. Therefore, the fewer options that are invalidated by the context in which the decision is made, the more commitment we make towards the outcome of the decision making process. Conversely, the more options invalidated by the context the less we attribute to the actual result of the decision process. This means that in situations in which there are few options remaining, the initial commitment towards the selected option will be slight. At first glance this seems unintuitive: there were few options so we should commit strongly to the selection made since future revisions have a high chance of selecting the same outcome given the stability of the background intentional structure. However, this analysis fails to consider the cost of deliberation. Making a selection amongst few options is less costly than making a selection amongst many. Thus, we attribute commitment proportional to the cost in making the selection. Furthermore, since there is a high probability that the same selection will be made provided the stability of the agent's beliefs and intentions it will accumulate commitment as a result of this reconsideration (see Definition 3).

When applied to the office robot example we find:

$$\begin{aligned} O &= \{\text{'deliver coffee'}, \text{'outsource coffee'}\}, \\ I &= \emptyset, \\ O \setminus I &= \emptyset, \\ s &= \text{'deliver coffee'}, \\ \text{base}(\text{'deliver coffee'}) &= \text{cost}(\text{'deliver coffee'}) \end{aligned}$$

due to the initially empty set of commitments. However, when selecting a behavior to handle the request for document delivery we have:

$$\begin{aligned} O &= \{\text{'deliver document'}, \text{'outsource document'}\}, \\ I &= \{\text{'deliver coffee'}\}, \\ O \setminus I &= \{\text{'deliver document'}\}, \end{aligned}$$

² We must use the summation as opposed to $|O \setminus I| \times |I \setminus c|$ because each $I \setminus c$ may be of a different size.

$$\begin{aligned}
 I \setminus 'deliver document' &= \{ 'deliver coffee' \} \\
 (O \setminus I) \setminus 'deliver coffee' &= \{ 'deliver document' \} \\
 s &= 'outsource document' \\
 \text{base}('outsource document') &= \frac{\text{cost}('outsource document')}{2} \\
 \text{weight}('deliver coffee', 'outsource document') &= \frac{1}{2}
 \end{aligned}$$

As can be seen, the agent's commitment towards delivering coffee is increased as a consequence of its use in forming the commitment towards outsourcing the delivery of the document.

2.2 Revised Intention

Example 2. Imagine the robot as described above has been acting in such a way as to deliver the coffee requested of it. Because it needs to collect the document and deliver it to the robot with whom it is cooperating, it has decided and committed to getting the coffee from the coffee machine between its current location and the meeting place instead of the closer gourmet coffee dispenser in the cafeteria. After making progress towards the coffee machine the other agent informs the robot that it will not be able to meet at the agreed time to accept the document due to unexpected circumstances. Consequently the agent reconsiders its intention to get the coffee from the coffee machine and instead sources the coffee from the cafeteria, committing to meet the robot at a later time. Because this change in behavior has wasted the agent's time and energy it now commits itself more strongly to its new intention to source the coffee from the cafeteria in order to minimize the chance of further reconsideration and wasted efforts.

Should an agent revise an intention, the commitment the agent adopts towards the replacement should be greater than the commitment it had towards the original. This is because the agent has made a choice to drop the existing intention and accepts all the costs related in doing so (as the cost should be part of the input to the decision to reconsider). This is important in maintaining the stability of the new intention. If it were not to inherit the commitment from the intention it replaced then it is likely that the new intention would quickly become a candidate for further revision.

Definition 3

$$\text{accum}(i') = \text{commit}(i) + \text{base}(i')$$

where:

i' : is the new intention.

i : is the old intention.

$\text{commit}(i)$: is the total commitment the agent had towards i (see Definition 5).

$\text{base}(i')$: the base commitment the agent associated with the original intention (see Definition 1).

Applying this to the example above we have: before the revision:

$$\text{base}(\text{'coffee from machine'}) = \frac{\text{cost}(\text{'coffee from machine'})}{2}$$

$$\text{base}(\text{'outsource document'}) = \frac{\text{cost}(\text{'outsource document'})}{2}$$

$$\text{base}(\text{'deliver coffee'}) = \text{cost}(\text{'deliver coffee'})$$

$$\text{weight}(\text{'deliver coffee'}, \text{'outsource document'}) = \frac{1}{2}$$

$$\text{weight}(\text{'outsource document'}, \text{'coffee from machine'}) = \frac{1}{2}$$

$$\text{weight}(\text{'deliver coffee'}) = \text{accum}(\text{'outsource document'}) =$$

$$\text{accum}(\text{'coffee from machine'}) = 0$$

and after the revision:

$$\text{base}(\text{'coffee from cafe'}) = \frac{\text{cost}(\text{'coffee from cafe'})}{2}$$

$$\text{base}(\text{'outsource document'}) = \frac{\text{cost}(\text{'outsource document'})}{2}$$

$$\text{base}(\text{'deliver coffee'}) = \text{cost}(\text{'deliver coffee'})$$

$$\text{weight}(\text{'deliver coffee'}, \text{'outsource document'}) = \frac{1}{2}$$

$$\text{weight}(\text{'outsource document'}, \text{'coffee from cafe'}) = \frac{1}{2}$$

$$\text{accum}(\text{'deliver coffee'}) = 0$$

$$\text{accum}(\text{'outsource document'}) = 0$$

$$\text{accum}(\text{'coffee from cafe'}) = \frac{\text{cost}(\text{'coffee from machine'})}{2}$$

2.3 Intention Cost

The theory, presented thus far, is agnostic to the methodology for calculating the cost of adopting a given intention. To demonstrate the feasibility of such a cost function we present an example approach below.

When adopting an intention, an agent must make a selection between competing options. In so doing, an agent discourages the pursuit of any desires that conflict with the newly intended option (as doing so would require a reconsideration of the agent's intentions). It also strengthens any conflicts with desires that it shares with other intentions. Intuitively, the cost of adopting a given intention i is a measure of the desires that the agent is unable to pursue due to intending i . Multiple intentions may conflict with a given desire and single intention may conflict with multiple desires. Therefore

we must distribute the cost of each lost opportunity among the intentions contributing to its invalidity, including the newly adopted intention.

In order to formalize this we assume that there is a set D of the agent's desires. These need not be consistent. For a given intention i , there is a subset of D with which it is inconsistent³:

$$D \setminus i = \{d \in D \mid \text{consistent}(i, d) = \perp\} \quad (4)$$

Given this set of conflicting desires and the agent's other intentions, the cost of an intention i is:

Definition 4

$$\text{cost}(i) = \begin{cases} \text{if } D \setminus i = \emptyset & 0 \\ \text{else} & \sum_{d \in D \setminus i} \frac{1}{|I(d)|} \end{cases}$$

where:

$\text{cost}(i)$: is the cost associated with intention i .

$I \setminus d$: is the set of intentions in conflict with a given desire d .

Let us now apply the above to the example office robot both prior to and after its adoption of a commitment to outsource the delivery of the document. To do so, assume the agent's set of desires are:

$$D = \left\{ \begin{array}{l} \text{'guide guest', 'vacuum', 'deliver coffee',} \\ \text{'deliver document'} \end{array} \right\}$$

and its conflict function is:

	'guide guest'	'vacuum'	'deliver coffee'	'deliver document'	'outsource document'
'guide guest'	\perp	\perp	\top	\top	\perp
'vacuum'	\perp	\perp	\perp	\perp	\perp
'deliver coffee'	\top	\perp	\perp	\top	\perp
'deliver document'	\top	\perp	\top	\perp	\top
'outsource document'	\perp	\perp	\perp	\top	\perp

This results in the following prior to adopting a commitment to outsource the document delivery:

$$D \setminus \text{'deliver coffee'} = \{ \text{'guide guest', 'transport document'} \}$$

$$I \setminus \text{'guide guest'} = \{ \text{'deliver coffee'} \}$$

$$I \setminus \text{'transport document'} = \{ \text{'deliver coffee'} \}$$

and thus:

$$\text{cost}(\text{'deliver coffee'}) = 2$$

and further to the following after the commitment to outsource the document delivery:

$$D \setminus \text{'deliver coffee'} = \{ \text{'guide guest', 'transport document'} \}$$

³ Consistent(i, d) is a function that maps an intention and a desire to true (\top) if they are simultaneously adoptable, and false (\perp) otherwise.

$$I \setminus \text{'guide guest'} = \{\text{'deliver coffee'}\}$$

$$I \setminus \text{'transport document'} = \{\text{'deliver coffee'}, \text{'outsource document'}\}$$

and thus:

$$\text{cost}(\text{'deliver coffee'}) = 1\frac{1}{2}$$

and:

$$D \setminus \text{'outsource document'} = \{\text{'transport document'}\}$$

$$I \setminus \text{'transport document'} = \{\text{'deliver coffee'}, \text{'outsource document'}\}$$

and thus:

$$\text{cost}(\text{'outsource'}) = 1\frac{1}{2}$$

2.4 Relative Commitment

Intentions are related by their use in decision making. When an agent makes a decision towards the adoption of a new intention, the outcome of such a decision is dependent on the agent's existing intentions. This dependency arises due to the consistency required of intentions. An agent with inconsistent intentions is likely to act in a way that prevents the achievement of its ends. The dependencies between intentions are not all equivalent. Some intentions may play a stronger role in invalidating the options available in a given decision than others. This strength is reflected in the weights associated with each dependency. The commitment an agent places in a given intention is heavily reliant on the weight of its dependencies and the commitment the agent places in the dependent intentions. We formalize this as follows:

Definition 5

$$\text{commit}(i) = \left(\sum_{i' \in I} \text{commit}(i') \times \text{weight}(i, i') \right) + \text{accum}(i) + \text{base}(i)$$

where:

i : is the intention in question.

i' : is an intentional structure dependent on i .

$\text{weight}(i, i')$: is a function that returns the weight of the dependency between intentions i and i' (see Definition 2).

$\text{accum}(i)$: is any commitment the agent has accumulated and attributed directly to intention i (see Definition 3).

$\text{base}(i)$: is the initial cost associated in adopting the intention (see Definition 1).

Notice that we utilize the definition of cost in the calculation of total commitment. That is because as an agent behaves this cost changes also. Should a dependency be broken for any reason the commitment the agent has towards the involved intention is automatically adjusted accordingly.

Applying the above to the office robot example we have:

$$\text{commit}(\text{'deliver coffee'}) = \frac{\text{commit}(\text{'outsource document'}) + \text{cost}(\text{'deliver coffee'})}{2}$$

$$\text{commit}(\text{'outsource document'}) = \frac{\text{commit}(\text{'coffee from cafe'}) + \text{cost}(\text{'outsource document'})}{2}$$

$$\text{commit}(\text{'coffee from cafe'}) = \frac{\text{cost}(\text{'coffee from machine'}) + \text{cost}(\text{'coffee from cafe'})}{2}$$

3 Triggering Intention Reconsideration

In the preceding section a methodology for capturing the commitment an agent has towards its evolving intentions was proposed. One of the primary uses for such a methodology is in the specification of when it is rational for an agent to reconsider these intentions in light of changing commitments. This takes the form of a revision function that given an intention and the agent's current intentions maps to true (\top) if the agent should reconsider the input intention and false (\perp) otherwise.

3.1 Changed Dependencies Reconsideration

The simplest policy is to reconsider an intention when any of its dependencies change.

Definition 6

$$\text{revise}(i, I) = \begin{cases} \top & \text{if } \exists i'. \text{weight}(i', i) > 0 \wedge I \cap i' = \emptyset \\ \perp & \text{otherwise} \end{cases}$$

where:

I : is the agent's current set of intentions.

i : is the intention we are deciding whether to reconsider or not.

i' : is an intention that was used in the decision to adopt i but is no longer held. (i.e. there is a weight between i' and i but i' is no longer in I).

If we apply this to the office robot scenario at the point at which the agent has committed to outsource the document delivery request the effect of the agent dropping the commitment to deliver coffee on the commitment towards outsourcing the delivery of the document is:

$$\text{revise}(\text{'outsource document'}, \{\text{'outsource document'}\}) = \top$$

that the agent will reconsider its commitment towards outsourcing the document. Conversely, the effect on dropping the intention towards outsourcing the document delivery:

$$\text{revise}(\text{'deliver coffee'}, \{\text{'deliver coffee'}\}) = \perp$$

is to not reconsider its intention towards the delivery of coffee.

3.2 Dependent Non-reconsideration

Although changes in the dependencies of an intention(i) motivate reconsideration, the intentions dependent on i provide justification for the non-reconsideration of i . This is because the dependents of i require its stability. If the agent reconsiders i , this further motivates the agent to reconsider all dependent intentions. Thus an agent should take the potential cost of such a reconsideration into account when deciding to reconsider. One simple policy an agent could adopt could be to not reconsider any intention which has dependents:

Definition 7

$$\text{revise}(i, I) = \begin{cases} \top & \text{if } \forall i'. i' \in I \wedge \text{weight}(i, i') \leq 0 \\ \perp & \text{otherwise} \end{cases}$$

where:

I : is the agent's current set of intentions.

i : is the intention we are deciding whether to reconsider or not.

$\text{weight}(i, i') \leq 0$: indicates that there is no dependency between i and i' .

Applying this to the office robot scenario at the same point as above, the effect of the agent dropping the commitment to deliver coffee on the commitment towards outsourcing the delivery of the document is:

$$\text{revise}(\text{'outsource document'}, \{\text{'outsource document'}\}) = \top$$

that the agent will reconsider its commitment towards outsourcing the document. Similarly, the effect on dropping the intention towards outsourcing the document delivery:

$$\text{revise}(\text{'deliver coffee'}, \{\text{'deliver coffee'}\}) = \top$$

is for the agent to reconsider its intention towards delivering coffee. This is because neither commitment has any dependencies at this point. However, if we apply the above prior to dropping either mentioned commitment:

$$\text{revise}(\text{'outsource document'}, \{\text{'deliver coffee'}, \text{'outsource document'}\}) = \top$$

$$\text{revise}(\text{'deliver coffee'}, \{\text{'deliver coffee'}, \text{'outsource document'}\}) = \perp$$

the agent will reconsider its intentions towards outsourcing the document but not towards the delivery of coffee.

3.3 Proportional Dependent Non-reconsideration

Alternatively, an agent should only reconsider an intention provided that doing so will not result in more dependent intentions than a predefined limit being reconsidered:

Definition 8

$$\text{revise}(i, I) = \begin{cases} \top & \text{if } \frac{|\{i' \in I \mid \text{weight}(i, i') > 0 \wedge \text{revise}(i', \{I - i\}) = \top\}|}{|\{i'' \in I \mid \text{weight}(i, i'') > 0\}|} < p \\ \perp & \text{otherwise} \end{cases}$$

where:

I : is the agent's current set of intentions.

i : is the intention we are deciding whether to reconsider or not.

p : the proportion of intentions that the agent is willing to potentially reconsider should it reconsider and drop the intention over which it is deliberating.

We illustrate this through the use of the office robot example:

$$\text{revise}(\text{'outsource document'}, \{\text{'outsource document'}\}) = \top$$

$$\text{revise}(\text{'deliver coffee'}, \{\text{'deliver coffee'}, \text{'outsource document'}\}) = \perp$$

Clearly an agent should balance the required stability of its intentions against the opportunities arising due to its changing intentions.

4 Discussion

An analysis of whether an agent should reconsider its current intentions in light of the need to adopt a conflicting intention was presented in [8,9]. The rationality behind such reconsideration was evaluated in terms of whether it results in a change to the agent's intentions. In addition, [10] argues that adapting or adopting existing intentions for the achievement of multiple goals is required for the efficient use of an agent's intentional background. It was noted, however, that doing so is not a panacea and should only be used to promote the known side-effects of plan execution to intention.

[11] investigated the question of when to reconsider an intention in the context of simple planning agents in the tile-world setting. Two reconsideration policies were examined: reconsideration after executing a set number of plan steps and as triggered by changing beliefs about the environment. [12] continued this empirical investigation on intention reconsideration policies. Two further approaches were considered: discrete deliberation scheduling and partially observable Markov decision processes (POMDP). Discrete deliberation scheduling treats the process of intention reconsideration as an action that the agent may schedule. POMDPs on the other hand are a learning mechanism that provided a probability distribution over the agent's beliefs can generate an optimal reconsideration policy.

Our approach differs significantly from those outlined above. The agents investigated above were planning agents that generate, adapt and adopt a single complete plan for the achievement of their intentions. We consider agents that adopt multiple independent, though consistent, intentions. Consequently the intention structure we consider is richer and facilitates additional relationships between intentions. We use the dynamics of these relationships to identify new intention reconsideration policies.

Another aspect unique to our approach is the explicit modeling of the commitments an agent makes towards its intentions. Typically commitment is an emergent property of the intention revision policy of the agent. This makes reasoning about the commitment an agent places in an intention impossible. Although we have explicit representation, we do not have a concrete interpretation of the commitment values other than in specifying

a preference ordering over the intentions. Without such semantics we cannot reason about compound commitments using the component commitments.

In order to focus on the role of the decision dependencies between intentions it was necessary to disregard some aspects that are crucial for a practical theory of intention and commitment. These aspects include priorities between intentions, resource usage, means-end reasoning, intention overloading and the effects of intentions. These topics remain the focus of future work.

5 Conclusion

This paper provides an initial investigation into intention, the commitment agents make towards their intentions and the uses to which intentions may be put in decision making. A crucial element of this investigation was how the dynamics of intentions affects the commitment an agent places in its intentions and how this commitment in turn affects the dynamics of an intention. Through a series of examples the intuitions and formalities of our approach were illustrated. Given this as a background a number of intention reconsideration strategies were proposed. These were then applied to the running examples to demonstrate their feasibility. Finally a brief comparison of this approach and the state of the art in intention reconsideration was presented, noting the unique properties and advantages our proposal provides.

References

1. Cohen, P., Levesque, H.: Intention = Choice + Commitment. In: Proceedings of the Sixth National Conference on Artificial Intelligence, pp. 410–415 (1987)
2. Georgeff, M., Rao, A.: The semantics of intention maintenance for rational agents. In: Mellich, C. (ed.) Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pp. 704–710 (1995)
3. Rao, A., Georgeff, M.: Decision procedures for bdi logic. *Journal of Logic and Computation* 8 (1998)
4. Meyer, J., van der Hoek, W., van Linder, B.: A logical approach to the dynamics of commitment. *Artificial Intelligence* 113, 1–40 (1999)
5. Singh, M., Asher, N.: Towards a formal theory of intentions. *Logic in Artificial Intelligence* 478, 472–486 (1990)
6. Singh, M.: Intentions, commitments and rationality. In: Proceedings of the Annual Conference of the Cognitive Science Society (1991)
7. Bratman, M.: Intention, Plans and Practical Reason. Harvard University Press (1987)
8. Bratman, M., Israel, D., Pollack, M.: Plans and resource-bounded practical reasoning. *Computational Intelligence* 4, 349–355 (1988)
9. Pollack, M.: The uses of plans. *Artificial Intelligence* 57, 43–68 (1992)
10. Pollack, M.: Overloading intentions for efficient practical reasoning. *Noûs* 25, 513–536 (1991)
11. Kinny, D., Georgeff, M.: Commitment and effectiveness of situated agents. In: Proceedings of the twelfth International Joint Conference on Artificial Intelligence, pp. 82–88 (1991)
12. Schut, M., Wooldridge, M., Parsons, S.: The theory and practice of intention reconsideration. *Journal of Experimental and Theoretical Artificial Intelligence* 16, 261–293 (2004)

Temporal Data Mining for Educational Applications

Carole R. Beal and Paul R. Cohen

University of Arizona
crbeal@email.arizona.edu, cohen@cs.arizona.edu

Abstract. Intelligent tutoring systems (ITSs) acquire rich data about students behavior during learning; data mining techniques can help to describe, interpret and predict student behavior, and to evaluate progress in relation to learning outcomes. This paper surveys a variety of data mining techniques for analyzing how students interact with ITSs, including methods for handling hidden state variables, and for testing hypotheses. To illustrate these methods we draw on data from two ITSs for math instruction. Educational datasets provide new challenges to the data mining community, including inducing action patterns, designing distance metrics, and inferring unobservable states associated with learning.

1 Introduction

Teachers, school administrators and parents have always wanted to know how their students are doing in the classroom. Recently, interest in tracking student learning has grown dramatically due to increased emphasis on accountability in educational settings. For example, in the United States, educators are in urgent need of accessible and meaningful information about how students are learning, in order to meet annual progress report requirements resulting from the 2002 No Child Left Behind act [1]. Schools face significant pressure to improve learning outcomes, yet improvement depends on the ability to identify in the short-term those student behaviors that are likely to be unproductive in the long term.

Intelligent tutoring systems (ITSs) potentially address the problem of assessing and tracking students and the problem of improving learning outcomes. It is possible to record keystroke-by-keystroke information as students use ITSs, and to process it quickly to provide teachers up-to-the-minute assessments of their students' performance, rather than waiting for weekly tests or annual end-of-year assessments [6]. ITSs have been shown to provide effective instruction, with some results showing improvement of 20-25% on pre- and post-test measures [2, 3]. Yet at the same time, there has been growing concern about the tendency of students to use such systems ineffectively. Indeed, students may actively avoid effort by adopting behavioral strategies that allow them to avoid learning. For example, a student may deliberately enter incorrect answers to elicit hints and, eventually, the correct answer from the ITS [4]. Thus, although ITS instruction is beneficial, its effectiveness might be increased if maladaptive student behaviors could be identified.

Thus, data mining techniques are essential to both assessing students' performance and enhancing the effectiveness of their efforts with ITSs. Only recently have researchers begun to examine ITS data in detail [5]. This paper surveys several techniques we developed to model the behaviors of students using two mathematics ITSs. One can look at ITS data on several scales, from characteristics of individual problems (Sec. 2) to sequences of problems for individual students (Sec. 3) to samples of entire tutoring sessions for groups of students (Secs. 4,5). At each scale one can both extract descriptive information (e.g., the average time required to solve a problem or the distribution of the number of problems in a tutoring session), and estimate the structure and parameters of models (e.g., Markov models of actions or long-run patterns of attention during tutoring sessions). One also can cluster students based on parameters of models (Sec. 4.1) and test hypotheses about groups of students (Sec. 6). This paper is primarily concerned with models of student behavior that can be used to improve the experiences students have with tutoring systems. Thus, we model the amount of time students are willing to spend on problems and how it changes over the course of a session, and we model unobservable factors such as engagement, using hidden variable models (Sec. 4.1).

The techniques in this paper were developed to analyze data from two mathematics ITSs: Wayang Outpost is an ITS for high school math; the dataset includes two samples of students (MA and CA). The MA dataset was weighted with a larger number of low achieving students, whereas the CA sample included a full range of student achievement levels. The second ITS, AnimalWatch, is for middle school math. Demonstration versions of these tutoring systems may be viewed at <http://k12.usc.edu>.

The basic pattern of interactions with both the Wayang and AnimalWatch systems is this: The ITS selects a problem that it thinks will advance the students learning. The student solves the problem and enters the answer, or selects an answer from a multiple-choice menu. The ITS uses the answer to update its model of the students competence. If the answer is wrong, the ITS presents hints sequentially until the student gets the answer correct. The data include characteristics of problems (e.g., expected and empirical difficulty, and the topics and skills they exercise), characteristics of hints (e.g., whether they are multimedia or text hints), and characteristics of the students performance (e.g., the number of hints they require and the time they devote to each).

2 Actions and Classifiers

The smallest scale at which data mining has a role is that of the individual problem-solving interactions, of which the most interesting involve wrong answers and hints. Because ITSs provide customized instruction to students, every interaction is literally unique. Said differently, the number of combinations of student attributes, versions of the ITS, problem topics, orders of presentation, selected hints, and latencies or student responses, is so large that some abstraction is necessary if we are to find useful regularities in data. Consequently, we

build classifiers of individual actions taken by students on single problems. These actions are different for Wayang Outpost and AnimalWatch, and thus required different pattern classifiers.

We stress that the classifiers we are about to discuss were designed by us, not learned from data. The automatic induction of classifiers for problems and hints is an open challenge for data mining, as discussed in Section 6, below. Because the classification of actions is so fundamental to all the analyses in the rest of this paper, we deal with it at some length.

Wayang Outpost, our high school math ITS, presented problems that included a figure, table or other graphic, the question or equation to be solved, and five answer options. The student received feedback (correct, incorrect) by clicking on an answer option. The student could also click on a “hint” icon to view an ordered series of audio and animation hints. The student could choose an answer at any point, or continue to view hints until the solution was displayed. Wayang Outpost thus required the student to request help.

The problems presented by the middle school math tutor, AnimalWatch, each included a text introduction with information necessary to solve the problem, the problem question or equation, an illustration, table or graphic, and an answer box. AnimalWatch followed a “help-forcing” model in that incorrect answers triggered help. Hints were ordered: Hint 1 provided text feedback (correct, incorrect); Hint 2 offered an “operation” text hint (e.g., “Are you sure you are subtracting?”); Hint 3 offered a multimedia interactive hint that could include multiple steps (e.g., dragging one-bars into tens-units on the screen to compose the additive quantity).

We defined action patterns that might be associated with effective and ineffective learning behaviors. They are patterns (as opposed to atomic actions) because they include latency information and are rough interpretations of the students intentions. For example, for Wayang Outpost, a student who chooses multiple incorrect answers before the correct answer might be trying to correct his or her own errors, but might also be simply guessing. We use the time between clicks on the answers to distinguish these interpretations, e.g., inter-click intervals of less than 5 seconds signal guessing. Similarly, if the latency between presenting a problem and the students first answer is less than ten seconds, the student may be guessing. These latencies were based on average times for the highest-performing students.

We defined five patterns associated with the high school math ITS: Guessing, Learning by using multimedia help, Solving problem independently and accurately, Solving problem independently but with errors, and Skipping problem. We defined a larger number of action patterns — nine in all — for the middle school ITS to differentiate the number of hints received by the student.

3 Student Behavior within Problems

We asked the following question of AnimalWatch data: For every problem that required at least one hint, what fraction of the total time required to solve the

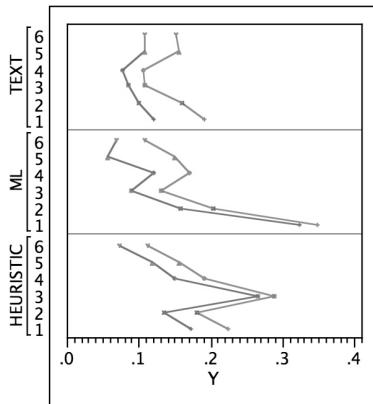


Fig. 1. Median and mean proportion of problem solving time in hint sequences for three versions of AnimalWatch ITS

problem was spent attending to each hint? The results are shown in Figure 1. The horizontal axis shows the proportion of the total time required to solve a problem. The vertical axis is organized by two variables: Hint number has six levels, for 1, 2, 3, 4, 5, > 5 hints, respectively. Group is either Heuristic, denoting a version of AnimalWatch that provided up to two textual hints and then multimedia help on all subsequent hints; ML, denoting a version that followed the same hint schedule but “pushed” problems of increasing difficulty more aggressively than Heuristic; or Text, which provided no help at all besides the textual feedback that an answer was right or wrong.

The left and right lines in Figure 1 represent mean and median proportions of problem-solving time, respectively. Some aspects of Figure 1 were anticipated but some were surprising. We expected the proportion of time on the third hint to be high because this is where multimedia help is provided for the first time. We had no such expectation for the Text group because they do not receive multimedia help. We were surprised that the ML group spent so little time on the multimedia hints, in contrast to the Heuristic group. In fact, they spent most time on the first hint (which was “wrong, try again”) and roughly the same amount of time as the Heuristic group on the second (“did you use the appropriate operator”), but then, unlike the Heuristic group, they spent even less time attending to the third hint. Thus, the detailed analysis of patterns in students’ hint usage revealed that trying to accelerate the pace of students’ learning had the unintended effect of reducing their attention to hints.

3.1 Rule Mining for Action Pattern Sequences

Each student’s data record included an ordered sequence of action patterns (as discussed in Sec. 2) representing her or his behavior on the math problems over the course of the tutoring session. Each action pattern was coded as a letter, and the sequence of problems solved by a student could be coded as a sequence

of letters such as [AAECBD]. Can we find predictive rules in these sequences? For example, if a student has guessed on two problems in a row, can we infer anything about what the student will do on the current problem? Predictive rules will have the form $A_{n-j} \dots A_n \rightarrow A_{n+1}$, where the left hand side is a subsequence of actions and the right hand side is the one we wish to predict. Can we find rules of this form, and if so, what is the best value of j ? If we view the generation of action patterns as a Markov process then j is the order of the Markov chain. Or, j might be allowed to vary, as described below.

First, a word about estimating the accuracy of predictions: In the case of Wayang Outpost, students can exhibit one of five action patterns on a math problem. However, these patterns are not equally likely so the default accuracy of predictions is not 1/5. We define the default accuracy to be the probability of the majority class action pattern because, lacking any other information, one's best guess is that the student's next action will be the majority class action. This turns out to be action 2, guess/abuse help. With 2028 instances in the combined corpus of MA and CA students, this is the most common of 7316 actions. Thus, the default accuracy is $(2028 / 7316) = .277$. This is the accuracy level we have to beat with any predictor.

3.2 Accuracy of Prediction Rules

We developed a rule-mining algorithm to find rules of the form $A_{n-j} \dots A_n \rightarrow A_{n+1}$. This algorithm, called Very Predictive Ngrams (VPN) finds rules with different values of j (unlike a Markov chain predictor of fixed order) so as to maximize predictiveness for a given number of rules [7]. The rules found by VPN predict the action that maximizes the conditional probability $Pr(A_{n+1}|A_{n-j} \dots A_n)$. The algorithm searches iteratively for K such rules. At each iteration the algorithm greedily adds the rule that will most reduce the errors in predictions. Let $A_{n-j-1}, A_{n-j} \dots A_n \rightarrow A_{n+1}$ be a child of the parent rule $A_{n-j} \dots A_n \rightarrow A_{n+1}$. The VPN algorithm is greedy in the sense that it will not necessarily find the child, even if it has better prediction accuracy than the parent, if it first finds another rule with better prediction accuracy than the parent. Nevertheless, VPN finds relatively small sets of rules that have prediction accuracies comparable to exhaustive sets of all possible rules of a given length (i.e., to Markov chain predictions of fixed order j).

The performance of VPN on Wayang Outpost action pattern sequences is shown in Figure 2. The horizontal axis is K, the number of ngrams found by VPN. The curve that drops from left to right is the error rate associated with each corpus of K rules. The curve that rises from left to right is the average length of a rule in the corpus (i.e., the average value of j).

VPN immediately finds five rules of length 1. With each, the error rate drops, going from roughly 85% to roughly 55%. After that, the error rate decreases very slowly, almost imperceptibly, while the average rule length increases. For these data, it is hard to get much better accuracy than is provided by a Markov chain predictor of order one. (For comparison purposes, an exhaustive corpus of 554 rules had an error rate of 0.506, while VPNs 50 rules have an error rate of

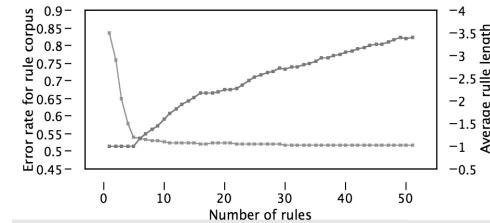


Fig. 2. Ngrams for Wayang Outpost action pattern sequences

0.516.) The lesson for designers of ITSs seems to be that the action pattern on the next problem depends to a very great extent on the previous pattern and not much on earlier patterns.

4 Session-Scale Patterns

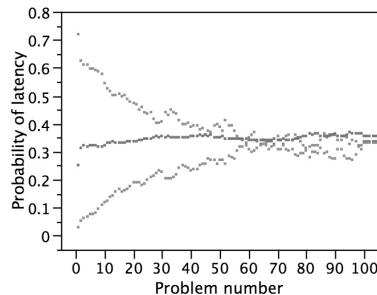
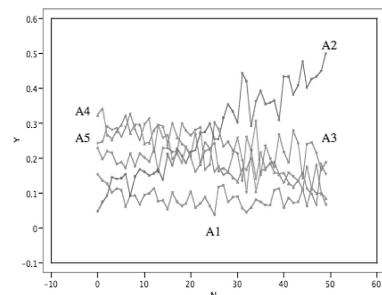
At the scale of short subsequences of action patterns, the behavior of students appears to be nearly Markov, but at larger scales, useful patterns emerge. We will focus on changes in the amount of time students spend on problems and hints over the course of a session with the AnimalWatch ITS.

The algorithm for finding these larger-scale patterns is simply to track the proportions of mutually exclusive and exhaustive attributes of problems over time for each student, and then average these proportions over all students within a group.

To illustrate, consider the time students spend on problems. Although this is a continuous variable, it can be binned into mutually exclusive and exhaustive ranges. Each problem falls into one bin, and we can track the proportions of all problems that fall into each bin over all tutoring sessions. The results, smoothed with a simple exponential smoother, are shown in Figure 3. The three trajectories correspond to spending less than 15 seconds on a problem, spending 15–35 seconds, and spending more than 35 seconds. The probability of each of these is plotted on the vertical axis. The horizontal axis represents the number of problems the student has seen.

Note that the mean probability (over all students sessions) of spending more than 35 seconds on a problem falls sharply as the number of problems increases, just as the average probability of spending fewer than 15 seconds increases. Interestingly, the average probability of spending 15–35 seconds on a problem remains roughly constant throughout sessions.

The amount of time that a student spends on a problem is to a large extent under the student's control. Thus, Figure 3 suggests that students are increasingly willing to rush through problems. Conversely, they are less willing to spend a long time working on a problem as a session wears on. However, they are willing to spend 15–35 seconds working on a problem, irrespective of the number of problems they have already seen. This kind of information is valuable for the

**Fig. 3.** Time in problem across problems**Fig. 4.** Action patterns across problems

designers of ITSs, and for teachers who integrate technology-based instruction into the classroom.

A similar analysis can be done for action patterns. Figure 4 shows five series, one for each of the five action patterns observed in interactions with the Wayang Outpost ITS. Each point on a line represents the probability of observing the action pattern associated with the line, on the n th problem.

Note that $\text{Pr}(A1)$, which is action pattern “skip problem”, is quite low and remains fairly constant across the session. $\text{Pr}(A2)$, which is “guess/abuse help”, starts low and increases steadily through the sequences, while $\text{Pr}(A4)$, “solve with help”, and $\text{Pr}(A5)$, “solve without help”, start high and decline. An enigmatic pattern is $A3$, which is attempting but failing to solve the problem, and not using multimedia help.

These results for Wayang Outpost (Fig. 4) echo those for AnimalWatch (Fig. 3). The interpretation of both results is that students are less willing to work on problems as the tutoring session goes on. Under this interpretation it would seem students are less and less engaged or motivated as the session goes on. Is there a way to model students’ engagement, even though it is not directly observable? Little is known about how students’ engagement evolves over time, or whether there are consistent trends that might be exploited in the design of more effective pedagogical models.

4.1 Unobservable States

A challenge for data mining is to infer unobservable factors that might explain patterns in students data. Students do not always perform in the most optimal manner, but we do not yet have ways to estimate the intentional states that influence their actions. Engagement has been suggested as a possible mechanism, referring to transient processes such as fluctuations in the learners attention, willingness to engage in effortful cognition, and emotions associated with successful and unsuccessful problem solving. These processes are not observable, but they might be inferred by models with hidden states, such as Hidden Markov Models (HMMs).

We fit HMMs to sequences of action patterns for students who worked with the Wayang Outpost high school math ITS, with three hidden states representing

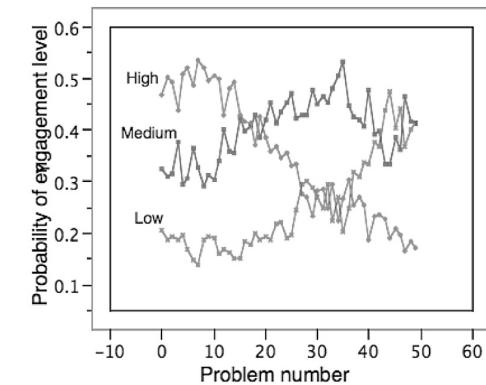


Fig. 5. Estimated engagement with the high school math ITS over problems

levels of engagement: low, average and high [8]. We tested the HMMs in several ways summarized here. First, we used the Viterbi parse of each student's HMM to predict his or her next action pattern and compared the accuracy of these predictions with a) an order one markov chain fit to the student's action sequence and b) an order one Markov chain fit to the entire group of students (CA or MA). Prediction accuracies for students' individual HMMs were 10%-15% higher than using the student's own Markov chain and compared very well with the Markov chain fit to the entire group.

We can also track probabilities of engagement levels in exactly the same way as probabilities of action patterns, above. First, for each student, we produce a Viterbi parse of engagement levels. This is the sequence of transitions through engagement states that makes the observed sequence of action patterns most likely. Then, having inferred what a student's engagement levels were likely to be over time, we get time series of proportions of engagement levels. Figure 5 shows engagement levels across problems for students who worked with the high school math ITS. It shows that students tend to start out the sessions in high engagement states, meaning that their action choices and latencies suggest that they are attending and concentrating. The probability of high engagement starts to decline around the 10th math problem, and continues to decline throughout the rest of the session. This decline is paralleled by a corresponding increase in the low engagement state.

5 Learning Outcomes

We have explored opportunities for mining structure at several scales, from individual problem solving actions to long-term changes in unobservable factors such as engagement. Now we ask a different though related question: How can we tell whether an ITS works well? One answer is to look at outcome variables such as scores on a test after a tutoring session. However, outcome variables tell

us little about what students do during tutoring sessions that might explain the outcomes.

By “explain” we mean that an experimental group exhibits a particular behavior associated with a high learning outcome whereas a control group does not exhibit this behavior and has a lower learning outcome. Thus, explanation requires us to compare behaviors across groups in a statistically valid way [10]. Here is one such method for comparing two groups:

1. Derive one or more functions $\theta(\sigma_i, \sigma_j)$ to compare students sequences of actions. Typically this function returns a real number.
2. Let $C_i = n_1(n_i - 1)/2$ be the number of pairwise comparisons between students within group G_i which contains n_i students. Define C_j in the same way.
3. Let $C_{i\cup j} = ((n_i + n_j)^2 - (n_i + n_j))/2$ be the number of pairwise comparisons between students across groups.
4. Let $\delta(i) = \sum_{a,b \in G_i} \theta(a, b)$ be the sum of all pairwise comparisons within G_i . Define $\delta(j)$ in the same way.
5. Let $\Delta(i, j) = \frac{(\delta(i) + \delta(j)) / (C_i + C_j)}{C_{i\cup j}}$.
6. If groups G_i and G_j are not different then one would expect $\Delta(i, j) = 1.0$. If $\Delta(i, j) \neq 1.0$, then we will wish to test whether it is significantly so. For this, we use a randomization procedure:
7. Randomization: Throw all the sequences in G_i and G_j into a single bucket G_{i+j} . Draw n_i sequences at random from G_{i+j} and call them G_i^* . Call the remaining n_j sequences G_j^* . Repeat steps 1-5 to get a single value of $\Delta(i, j)^*$. Repeat this process to get a few hundred values of $\Delta(i, j)^*$. The distribution of $\Delta(i, j)^*$ serves as a sampling distribution under the null hypothesis that groups G_i and G_j are not different. We compare $\Delta(i, j)$ to this distribution to get a p value, a probability of incorrectly rejecting the null hypothesis when it is true. (See [11] for details on randomization and bootstrap hypothesis testing.)

This procedure generalizes to multiple groups in the obvious way: If there are no differences between the groups then the average comparison among elements in each group will equal the average comparison among elements of the union of all the groups.

5.1 Comparing Sequences for Text and Heuristic Students

We used the preceding method to compare progress for students who worked with either the Text or Heuristic version of the AnimalWatch ITS (Text provided only feedback about answer accuracy; Heuristic provided multimedia help). We looked at each student after 0, 10, 20, ..., 90 problems and recorded how many problems on each of nine mathematics topics the student solved. Students sequences were compared with the following function:

$$\theta(\sigma_i, \sigma_j) = \sum_{t=0,10,20\dots} \sqrt{\sum_{i=1,2,\dots,9} (m_{i,a} - m_{i,b})^2}$$

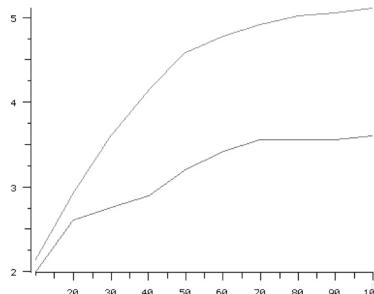


Fig. 6. Problem classes mastered to criterion for two versions of the AnimalWatch ITS

That is, for two students, a and b , we look at the number of problems, m_i , of each class $i = 1, 2, \dots, 9$ solved by each student, and square the differences. In other words, we treat a student's sequence as a trajectory through a nine-dimensional space, where each dimension is a problem class and each location in the space is given by a nine-vector of the number of problems of each class solved at that point. Then we compare sequences by the sum of Euclidean distances between corresponding points (i.e., points with equal values of t) in nine-space.

The test statistic was rejected only twice in 1000 randomization trials, so we can reject the null hypothesis that progress through the nine-topic problem space is the same for students in the Text and Heuristic conditions, with $p = .002$, a highly significant result.

It is one thing to test whether student in different experimental groups are different, another to visualize how they are different. This is not easily done when progress is in a nine-dimensional space. However, we can track the number of problem classes a student has mastered to some level at each point in a session. We chose a 50% criterion level, meaning that at a given point in a sequence, a student must have mastered 50% of the problems within a class to be given credit for mastery of that class at that point. We divided the students' sequences into subsequences of size 10 and calculated mastery, as defined, for each subsequence. Then we averaged the results over students, shown in Figure 6.

The vertical axis is the mean number of problem classes mastered at the 50% level, the horizontal axis is actually ten points, one for each subsequence of ten problems, and so represents 100 problems. The higher of the two lines corresponds to the Heuristic condition, the lower to Text. One sees that on average, a student in the Heuristic condition masters roughly five topics to the criterion level of 50% in the first 100 problems, whereas students in the Text condition master only 3.5 topics to this level in the same number of attempts. These curves also can be compared with our randomization procedure, and are significantly different.

6 Challenges for Educational Data Mining

We have shown that ITS data has structure at several scales, from micro-sequences of hints, to short sequences of actions, to long-term patterns over sessions and even between sessions. We have demonstrated the utility (in terms of prediction accuracy) of models with hidden state. We introduced a method to derive p values for statistical comparisons of groups of sequences. Some of our analyses were done algorithmically, others the old-fashioned way, with statistics packages. Thus, the first challenge we would like to issue to the data mining community is to develop algorithms to find some of the regularities we found by hand.

Of these, the most important are what we call action patterns. They are important because they are the elements on which all subsequent analyses are based. Recall, we defined five action patterns for Wayang Outpost (Sec. 2). To define these, we used information about problem difficulty, the distribution of latencies, the best-performing students, and the latencies and correctness of responses on particular problems. A fine challenge for data mining is to exploit these and other kinds of information about problems to induce action patterns automatically.

Another challenge is to invent new distance measures for comparing sequences. The generalized Euclidean distance in the previous section is adequate but it does not acknowledge that every problem instance has structure, which includes latencies, the sequences of hints, levels of engagement, the problem topic, the number of problems a student has seen, and so on. It is one thing to compare the raw counts of each of nine kinds of problems with Euclidean distance, another to compare highly-structured records as just described.

A third challenge is to develop more sophisticated ways to induce intentional states such as engagement. Our HMMs induced something we called engagement, and we were happy to see engagement profiles for students cluster nicely and comparably across student populations. Yet we have no independent verification that the hidden states in our HMMs actually correspond to intentional states. This experimental work needs to be done and the HMM models (or other hidden-state models) need to be refined.

The value of data mining for educational applications is enormous. National standards pertain to end results — performance — and provide no information about the process of learning. Teachers cannot easily individualize instruction because they do not have fine-grained analyses of how their students learn. Report cards are crude assessments and arrive too late to help. Whether one applies data mining techniques to the rich information available from ITSs or to other information gathered in classrooms, there is great potential for data mining to improve the quality of educational experiences.

Acknowledgments

The research described in this paper was supported by grants from the U. S. National Science Foundation and the U.S. Institute of Education Sciences. The

views expressed are not necessarily those of the funding agencies. We would like to recognize and thank our project colleagues Sinjini Mitra, Erin Shaw, Jean-Phillippe Steinmetz, and Lei Qu at the Information Sciences Institute-USC, and Ivon Arroyo and Beverly Woolf at the University of Massachusetts-Amherst.

References

- [1] (Retrieved July 8, 2006), <http://www.ed.gov/nclb/landing.jhtml>
- [2] Beal, C.R., Qu, L., Lee, H.: Classifying learner engagement through integration of multiple data sources. In: Proceedings of the 21st National Conference on Artificial Intelligence. AAAI Press, Menlo Park (2006)
- [3] Koedinger, K.R., Corbett, A.T., Ritter, S., Shapiro, L.J.: Carnegie Learnings Cognitive Tutor: Summary of research results. Carnegie Learning, Pittsburgh (2000)
- [4] Baker, R.S., Corbett, A.T., Koedinger, K.R., Roll, I.: Detecting when students game the system, across tutor subjects and classroom cohorts. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 220–224. Springer, Heidelberg (2005)
- [5] Beck, J.: Engagement tracing: Using response times to model student disengagement. In: Looi, C., McCalla, G., Bredeweg, B., Breuker, J. (eds.) Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology, pp. 88–95. IOS Press, Amsterdam (2005)
- [6] Stevens, R., Johnson, D., Soller, A.: Probabilities and prediction: Modeling the development of scientific problem solving skills. Cell Biology Education 4, 42–57 (2005)
- [7] Sutton, D., Cohen, P.R.: Very predictive Ngrams for space-limited probabilistic models. In: Pfennig, F., et al. (eds.) Advances in intelligent data analysis V, pp. 134–142. Springer, Berlin (2003)
- [8] Beal, C.R., Mitra, S., Cohen, P.R.: Modeling learning patterns of students with a tutoring system using Hidden Markov Models. In: Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED), Rey, CA (2006) (July 2007)
- [9] Ramoni, M., Sebastiani, P., Cohen, P.R.: Bayesian clustering by dynamics. Machine Learning 47, 91–121 (2001)
- [10] Beal, C.R., Cohen, P.R.: Computational methods for evaluating student and group learning histories in intelligent tutoring systems. In: Looi, C., McCalla, G., Bredeweg, B., Breuker, J. (eds.) Artificial Intelligence in Education: Supporting learning through intelligent and socially-informed technology, pp. 80–87. IOS Press, Amsterdam (2005)
- [11] Cohen, P.R.: Empirical methods for artificial intelligence. MIT Press, Cambridge (1995)

Dual Properties of the Relative Belief of Singletons

Fabio Cuzzolin

Oxford Brookes University
fabiocuzzolin@gmail.com

Abstract. In this paper we prove that a recent Bayesian approximation of belief functions, the relative belief of singletons, meets a number of properties with respect to Dempster’s rule of combination which mirrors those satisfied by the relative plausibility of singletons. In particular, its operator commutes with Dempster’s sum of plausibility functions, while perfectly representing a plausibility function when combined through Dempster’s rule. This suggests a classification of all Bayesian approximations into two families according to the operator they relate to.

1 Introduction: A New Bayesian Approximation

The theory of evidence (ToE) [1] extends classical probability theory through the notion of *belief function* (b.f.), a mathematical entity which independently assigns probability values to *sets* of possibilities rather than single events. A belief function $b : 2^\Theta \rightarrow [0, 1]$ on a finite set (“frame”) Θ has the form $b(A) = \sum_{B \subseteq A} m_b(B)$ where $m_b : 2^\Theta \rightarrow [0, 1]$, is called “basic probability assignment” (b.p.a.), and meets normalization $\sum_{A \subseteq \Theta} m_b(A) = 1$ and positivity $m_b(A) \geq 0 \forall A \subseteq \Theta$ axioms. Events associated with non-zero basic probabilities are called “focal elements”. A b.p.a. can be uniquely recovered from a belief function through Moebius inversion:

$$m_b(A) = \sum_{B \subseteq A} (-1)^{|A-B|} b(B). \quad (1)$$

As probability measures or *Bayesian* belief functions are just a special class of b.f.s (for which $m(A) = 0$ when $|A| > 1$), the relation between beliefs and probabilities plays a major role in the theory of evidence [2,3,4,5,6]. Tessem [7], for instance, incorporated only the highest-valued focal elements in his $m_{k\ell x}$ approximation. In Smets’ “Transferable Belief Model” [8] beliefs are represented at credal level (as convex sets of probabilities), while decisions are made by resorting to a Bayesian belief function called pignistic transformation [9]. More recently, two new Bayesian approximations of b.f.s have been derived from purely geometric considerations [10] in the context of the geometric approach to the ToE.

Another classical approximation is based on the plausibility function (pl.f.) $pl_b : 2^\Theta \rightarrow [0, 1]$, where

$$pl_b(A) \doteq 1 - b(A^c) = \sum_{B \cap A \neq \emptyset} m_b(B)$$

represent of the evidence not against a proposition A . Voorbraak [11] proposed the so-called *relative plausibility of singletons* (rel.plaus.) \tilde{pl}_b as the unique probability that, given a belief function b with plausibility pl_b , assigns to each singleton $x \in \Theta$ its normalized plausibility:

$$\tilde{pl}_b(x) = \frac{pl_b(x)}{\sum_{y \in \Theta} pl_b(y)}. \quad (2)$$

He proved that \tilde{pl}_b is a perfect representative of b when combined with other probabilities p through Dempster's rule \oplus [12]: $\tilde{pl}_b \oplus p = b \oplus p$. Its properties have been later discussed by Cobb and Shenoy [13].

Another Bayesian approximation based on normalizing the *belief* (instead of plausibility) values of singletons has been recently introduced [14]:

$$\tilde{b}(x) \doteq \frac{b(x)}{\sum_{y \in \Theta} b(y)} = \frac{m_b(x)}{\sum_{y \in \Theta} m_b(y)}. \quad (3)$$

(3) is called *relative belief of singletons* \tilde{b} (rel.bel.). Clearly \tilde{b} exists iff b assigns some mass to singletons:

$$\sum_{x \in \Theta} m_b(x) \neq 0. \quad (4)$$

The different semantics and limitations of the relative belief of singletons have been studied by F. Cuzzolin [14]. In particular, rel.bel. provides a conservative estimate of the evidence supporting each singleton $x \in \Theta$, and can indeed be seen as the relative plausibility of a plausibility function.

1.1 Aim of the Paper

On our side, in this paper we focus on the behavior of the relative belief of singletons with respect to evidence combination in the form of Dempster's combination rule. We prove that rel.bel. meets a number of properties with respect to Dempster's rule of combination which mirrors those satisfied by the relative plausibility of singletons (2). In particular: 1. its operator commutes with Dempster's sum of plausibility functions, and 2. rel.bel. perfectly represents a plausibility function when combined through Dempster's rule.

These results together with those holding for the relative plausibility suggest a clear subdivision of all Bayesian approximations in two families, related to Dempster's sum and affine combination respectively.

After briefly recalling the different semantics of the relative belief of singletons we summarize the properties of rel.plaus. with respect to Dempster's rule, whose

dual we are going to prove here for \tilde{b} . To this purpose we introduce the notion of "pseudo belief functions", i.e. b.f.s which admit negative b.p.a.s, as the basic tool we need in the course of this work.

We prove that the relative belief operator commutes with respect to Dempster's combination of plausibility functions, and enjoys idempotence properties similar to those met by the relative plausibility. Analogously, convergence results for rel.bel. can also be proven. In the last Section we prove that the relative belief of singletons perfectly represents the corresponding plausibility function pl_b when combined with any probability through (extended) Dempster's rule.

2 Semantics of the Relative Belief of Singletons

2.1 A Conservative Estimate

A first insight on the meaning of \tilde{b} comes from the original semantics of belief functions as constraints on the actual allocation of mass of an underlying unknown probability distribution. A focal element A with mass $m_b(A)$ indicates that this mass can "float" around in A and be distributed arbitrarily between the elements of A . In this framework \tilde{pl}_b (2) can be interpreted as follows:

- for each singleton $x \in \Theta$ the most optimistic hypothesis in which the mass of *all* $A \supseteq \{x\}$ focuses on x is considered, yielding $\{pl_b(x), x \in \Theta\}$;
- this assumption, however, is contradictory as it is supposed to hold for all singletons (many of which belong to the same higher-size events);
- nevertheless, the obtained values are normalized to yield a Bayesian belief function.

\tilde{pl}_b is associated with the less conservative (but incoherent) scenario in which all the mass that can be assigned to a singleton is actually assigned to it.

The relative belief of singletons (3) has in turn the following interpretation in terms of mass assignments:

- for each singleton $x \in \Theta$ the most *pessimistic* hypothesis in which only the mass of $\{x\}$ itself actually focuses on x is considered, yielding $\{b(x) = m_b(x), x \in \Theta\}$;
- this assumption is also contradictory, as the mass of all higher-size events is not assigned to any singletons;
- the obtained values are again normalized.

Dually, \tilde{b} reflects the most conservative (but still not coherent) choice of assigning to x only the mass that the b.f. b (seen as a constraint) assures it belongs to x , in perfect analogy to the case of rel.plaus.

One can argue that the existence of rel.bel. is subject to quite a strong condition (4). However it can be proven that the case in which \tilde{b} does not exist is indeed pathological, as it excludes a great deal of belief and probability measures [14].

2.2 Convergence under Quasi-bayesianity

A different angle on the utility of \tilde{b} comes from a discussion of what classes of b.f.s are “suitable” to be approximated by means of (3). As it only makes use of the masses of singletons, working with \tilde{b} requires storing n values to represent a belief function. As a consequence, the computational cost of combining new evidence through Dempster’s rule or disjunctive combination [15] is reduced to $O(n)$ as only the mass of singletons has to be calculated.

When the actual values of $\tilde{b}(x)$ are close to those provided by, for instance, pignistic function or rel.plaus. is then more convenient to resort to the relative belief transformation.

Let us call *quasi-Bayesian* b.f.s the belief functions b for which the mass assigned to singletons is very close to one:

$$k_{m_b} \doteq \sum_{x \in \Theta} m_b(x) \rightarrow 1.$$

Proposition 1. *For quasi-Bayesian b.f.s all Bayesian approximations converge:*

$$\lim_{k_{m_b} \rightarrow 1} \text{BetP}[b] = \lim_{k_{m_b} \rightarrow 1} \tilde{pl}_b = \lim_{k_{m_b} \rightarrow 1} \tilde{b}.$$

For quasi-Bayesian b.f.s the relative belief works as a low-cost proxy for the other Bayesian approximations.

3 Relative Plausibility and Dempster’s Rule

Rel.bel. and rel.plaus. are then strictly related. In this paper we prove indeed that \tilde{b} and \tilde{pl}_b share also an intimate relationship with Dempster’s evidence combination rule \oplus , as they meet a set of dual properties with respect to \oplus .

Definition 1. *The orthogonal sum or Dempster’s sum of two belief functions b_1, b_2 on the same frame Θ is a new belief function $b_1 \oplus b_2$ on Θ with b.p.a.*

$$m_{b_1 \oplus b_2}(A) = \frac{\sum_{B \cap C = A} m_{b_1}(B) m_{b_2}(C)}{\sum_{B \cap C \neq \emptyset} m_{b_1}(B) m_{b_2}(C)} \quad (5)$$

where m_{b_i} denotes the b.p.a. associated with b_i .

We denote with $k(b_1, b_2)$ the denominator of Equation (5).

Cobb and Shenoy [13] proved that the relative plausibility function \tilde{pl}_b commutes with respect to Dempster’s rule. More precisely, they proved that the relative plausibility of singletons meets the following properties¹ which relates to Dempster’s combination rule.

¹ Original statements from [13] have been reformulated according to the notation of this paper.

Proposition 2. 1. if $b = b_1 \oplus \dots \oplus b_m$ then $\tilde{pl}_b = \tilde{pl}_{b_1} \oplus \dots \oplus \tilde{pl}_{b_m}$. In other words, Dempster's sum and the relative plausibility operator

$$\begin{aligned} \tilde{pl} : \mathcal{B} &\rightarrow \mathcal{P} \\ b &\mapsto \tilde{pl}[b] = \tilde{pl}_b \end{aligned} \quad (6)$$

commute.

2. if m_b is idempotent with respect to Dempster's rule, i.e. $m_b \oplus m_b = m_b$, then \tilde{pl}_b is idempotent with respect to \oplus .
3. let us define the limit of a belief function b as

$$b^\infty \doteq \lim_{n \rightarrow \infty} b^n \doteq \lim_{n \rightarrow \infty} b \oplus \dots \oplus b \quad (n \text{ times}); \quad (7)$$

- if $\exists x \in \Theta$ such that $pl_b(x) > pl_b(y) \forall y \neq x, y \in \Theta$, then $\tilde{pl}_{b^\infty}(x) = 1$, $\tilde{pl}_{b^\infty}(y) = 0 \forall y \neq x$.
- 4. if $\exists A \subseteq \Theta$ ($|A| = k$) s.t. $pl_b(x) = pl_b(y) \forall x, y \in A$, $pl_b(x) > pl_b(z) \forall x \in A, z \in A^c$, then $\tilde{pl}_{b^\infty}(x) = \tilde{pl}_{b^\infty}(y) = 1/k \forall x, y \in A$, $\tilde{pl}_{b^\infty}(z) = 0 \forall z \in A^c$.

On his side, Voorbraak has shown [11] that the relative plausibility function perfectly represents a belief function when combined with a probability.

Proposition 3. The relative plausibility of singletons \tilde{pl}_b is a perfect representative of b in the probability space when combined through Dempster's rule, i.e.

$$b \oplus p = \tilde{pl}_b \oplus p, \quad \forall p \in \mathcal{P}.$$

4 Pseudo Belief Functions

The study of the properties of \tilde{b} requires first to extend the set of objects we work on from that of b.f.s to the more general class of "pseudo belief functions". Namely, the b.p.a. m_b associated with a b.f. b meets the positivity axiom: $m_b(A) \geq 0 \forall A \subseteq \Theta$. If we relax this condition we get functions ς of the form

$$\varsigma(A) = \sum_{B \subseteq A} m_\varsigma(B).$$

or *sum function* [16] whose Moebius inverse (1) $m_\varsigma : 2^\Theta \setminus \emptyset \rightarrow \mathbb{R}$ may assume negative values: $m_\varsigma(B) \not\geq 0 \forall B \subseteq \Theta$. Sum functions meeting the normalization axiom $\sum_{\emptyset \subsetneq A \subseteq \Theta} m_\varsigma(A) = 1$, or *pseudo belief functions* (p.b.f.s) [17], are then natural extensions of belief functions.

4.1 Plausibilities as Pseudo Belief Functions

Plausibility functions are p.b.f.s, as they meet the normalization constraint $pl_b(\Theta) = 1$ for all b . Their Moebius inverse [18]

$$\mu_b(A) \doteq \sum_{B \subseteq A} (-1)^{|A \setminus B|} pl_b(B) = (-1)^{|A|+1} \sum_{B \supseteq A} m_b(B) \quad (8)$$

when $A \neq \emptyset$, $\mu_b(\emptyset) = 0$ is called *basic plausibility assignment* (b.p.l.a.).

Each pl.f. is an affine combination of *basis* belief functions

$$b_A \doteq b \in \mathcal{B} \text{ s.t. } m_b(A) = 1, m_b(B) = 0 \forall B \neq A \quad (9)$$

with coefficients given by its b.p.l.a. [18]:

$$pl_b = \sum_{A \subseteq \Theta} \mu_b(A) b_A. \quad (10)$$

4.2 Dempster's Sum of Pseudo Belief Functions

The orthogonal sum can be naturally extended to pseudo b.f.s by applying (5) to the Moebius inverses $m_{\varsigma_1}, m_{\varsigma_2}$ of a pair of p.b.f.s. As Cuzzolin has proven [19].

Proposition 4. *Dempster's rule defined as in Equation (5) when applied to a pair of pseudo belief functions ς_1, ς_2 yields again a pseudo belief function.*

We denote the orthogonal sum of two p.b.f.s ς_1, ς_2 by $\varsigma_1 \oplus \varsigma_2$.

5 Dual Results for Relative Belief Operator

5.1 The Relative Belief Operator

As pl.f.s are pseudo b.f.s, Dempster's rule can then be formally applied to pl.f.s too. We can then prove a dual commutativity result for relative beliefs, once introduced (in full analogy to what done for the other Bayesian approximations) the *relative belief operator*

$$\begin{aligned} \tilde{b} : \mathcal{PL} &\rightarrow \mathcal{P} \\ pl_b &\mapsto \tilde{b}[pl_b] \end{aligned}$$

where

$$\tilde{b}[pl_b](x) \doteq \frac{m_b(x)}{\sum_{y \in \Theta} m_b(y)} \quad \forall x \in \Theta \quad (11)$$

is defined as usual for b.f.s b such that $\sum_x m_b(x) \neq 0$.

As a matter of fact, since b and pl_b are in 1-1 correspondence, we could indifferently define two operators mapping respectively a belief function b onto its relative belief, or the unique plausibility function pl_b associated with b onto \tilde{b} . We chose to consider the operator in this second form as this is instrumental to prove the following theorem, the dual of point 1. in Proposition 2.

5.2 Commutativity

A useful property of μ_b is that [14].

Lemma 1. $m_b(x) = \sum_{A \supseteq \{x\}} \mu_b(A)$.

Theorem 1. *The relative belief operator commutes with respect to Dempster's combination of plausibility functions, namely*

$$\tilde{b}[pl_1 \oplus pl_2] = \tilde{b}[pl_1] \oplus \tilde{b}[pl_2].$$

Theorem 1 implies that

$$\tilde{b}[(pl_b)^n] = (\tilde{b}[pl_b])^n. \quad (12)$$

5.3 Idempotence

Another consequence of Theorem 1 is an idempotence property which is the dual of point 2. of Proposition 2.

Theorem 2. *If pl_b is idempotent with respect to Dempster's rule, i.e. $pl_b \oplus pl_b = pl_b$, then $\tilde{b}[pl_b]$ is itself idempotent: $\tilde{b}[pl_b] \oplus \tilde{b}[pl_b] = \tilde{b}[pl_b]$.*

Proof. By Theorem 1 $\tilde{b}[pl_b] \oplus \tilde{b}[pl_b] = \tilde{b}[pl_b \oplus pl_b]$, and if $pl_b \oplus pl_b = pl_b$ the thesis immediately follows. \square

5.4 Convergence

The dual statements of the convergence results of Proposition 2 can be proven in a similar fashion.

Theorem 3. *If $\exists x \in \Theta$ such that $b(x) > b(y) \forall y \neq x, y \in \Theta$ then*

$$\tilde{b}[pl_b^\infty](x) = 1, \quad \tilde{b}[pl_b^\infty](y) = 0 \quad \forall y \neq x.$$

A similar proof can be provided for the following generalization of Theorem 3.

Theorem 4. *if $\exists A \subseteq \Theta$ ($|A| = k$) s.t. $b(x) = b(y) \forall x, y \in A$, $b(x) > b(z) \forall x \in A, z \in A^c$ then*

$$\tilde{b}[pl_b^\infty](x) = \tilde{b}[pl_b^\infty](y) = 1/k \quad \forall x, y \in A, \quad \tilde{b}[pl_b^\infty](z) = 0 \quad \forall z \in A^c.$$

5.5 Example

Let us consider the belief function b on the frame of size four $\Theta = \{x, y, z, w\}$ defined by the following basic probability assignment:

$$m_b(\{x, y\}) = 0.4, \quad m_b(\{y, z\}) = 0.4, \quad m_b(w) = 0.2. \quad (13)$$

The corresponding b.p.l.a. is by (8)

$$\begin{aligned} \mu_b(x) &= 0.4, & \mu_b(y) &= 0.8, & \mu_b(z) &= 0.4, \\ \mu_b(w) &= 0.2, & \mu_b(\{x, y\}) &= -0.4, & \mu_b(\{y, z\}) &= -0.4. \end{aligned} \quad (14)$$

To check the validity of Theorems 1 and 3 let us then compute the series $(\tilde{b}[pl_b])^n$ and $\tilde{b}[(pl_b)^n]$. By applying Dempster's rule to the b.p.l.a. (14) ($pl_b^2 = pl_b \oplus pl_b$) we get a new b.p.l.a. μ_b^2 with (see Figure 1)

$$\begin{aligned} \mu_b^2(x) &= 4/7, & \mu_b^2(y) &= 8/7, & \mu_b^2(z) &= 4/7, \\ \mu_b^2(w) &= -1/7, & \mu_b^2(\{x, y\}) &= -4/7, & \mu_b^2(\{y, z\}) &= -4/7. \end{aligned}$$

{y,z}		{y}	{z}		{y}	{y,z}
{x,y}	{x}	{y}			{x,y}	{y}
{w}				{w}		
{z}			{z}			{z}
{y}		{y}			{y}	{y}
{x}	{x}				{x}	

{x} {y} {z} {w} {x,y}{y,z}

Fig. 1. Intersection of focal elements in Dempster's combination of the b.p.l.a. (14) with itself. Non-zero mass events for each addendum $\mu_1 = \mu_2 = \mu_b$ correspond to rows/columns of the table, each entry of the table hosting the related intersection.

To compute the corresponding relative belief $\tilde{b}[pl_b^2]$ we first need to get the plausibility values

$$\begin{aligned} pl_b^2(\{x, y, z\}) &= \mu_b^2(x) + \mu_b^2(y) + \mu_b^2(z) + \mu_b^2(\{x, y\}) + \mu_b^2(\{y, z\}) = 8/7, \\ pl_b^2(\{x, y, w\}) &= pl_b^2(\{x, z, w\}) = pl_b^2(\{y, z, w\}) = 1 \end{aligned}$$

which imply by Definition $pl_b(A) \doteq 1 - b(A^c)$

$$b^2(w) = -1/7, \quad b^2(z) = 0, \quad b^2(y) = 0, \quad b^2(x) = 0$$

i.e. $\tilde{b}[pl_b^2] = [0, 0, 0, 1]'$.

Theorem 1 is confirmed as by (13) (being $\{w\}$ the only singleton with non-zero mass) $\tilde{b} = [0, 0, 0, 1]'$ so that $\tilde{b} \oplus \tilde{b} = [0, 0, 0, 1]'$ and $\tilde{b}[\cdot]$ commutes with $pl_b \oplus$.

By combining pl_b^2 with pl_b one more time we get the b.p.l.a.

$$\begin{aligned} \mu_b^3(x) &= \mu_b^3(z) = 16/31, & \mu_b^3(y) &= 32/31, \\ \mu_b^3(w) &= -1/31, & \mu_b^3(\{x, y\}) &= \mu_b^3(\{y, z\}) = -16/31 \end{aligned}$$

which corresponds to

$$\begin{aligned} pl_b^3(\{x, y, z\}) &= 32/31, & pl_b^3(\{x, y, w\}) &= 1, \\ pl_b^3(\{x, z, w\}) &= 1, & pl_b^3(\{y, z, w\}) &= 1 \end{aligned}$$

i.e.

$$b^3(w) = -1/31, \quad b^3(z) = 0, \quad b^3(y) = 0, \quad b^3(x) = 0$$

and $\tilde{b}[pl_b^3] = [0, 0, 0, 1]'$ which again is equal to $\tilde{b} \oplus \tilde{b} \oplus \tilde{b}$ as Theorem 1 guarantees. Clearly the series of the basic plausibilities $(\mu_b)^n$ converges to

$$\begin{aligned} \mu_b^n(x) &\rightarrow 1/2^+, & \mu_b^3(y) &\rightarrow 1^+, & \mu_b^3(z) &\rightarrow 1/2^+, \\ \mu_b^3(w) &\rightarrow 0^-, & \mu_b^3(\{x, y\}) &\rightarrow -1/2^-, & \mu_b^3(\{y, z\}) &\rightarrow -1/2^- \end{aligned}$$

associated with the following plausibility values

$$\begin{aligned} \lim_{n \rightarrow \infty} pl_b^n(\{x, y, z\}) &= 1^+, & pl_b^n(\{x, y, w\}) &= 1, \\ pl_b^n(\{x, z, w\}) &= 1, & pl_b^n(\{y, z, w\}) &= 1 \end{aligned} \quad \forall n \geq 1$$

which correspond to $\lim_{n \rightarrow \infty} b^n(w) = 0^-$, $b^n(z) = b^n(y) = b^n(x) = 0 \forall n \geq 1$, so that

$$\begin{aligned} \lim_{n \rightarrow \infty} \tilde{b}[pl_b^\infty](w) &= \lim_{n \rightarrow \infty} \frac{b^n(w)}{b^n(w)} = 1 \\ \lim_{n \rightarrow \infty} \tilde{b}[pl_b^\infty](x) &= \lim_{n \rightarrow \infty} \tilde{b}[pl_b^\infty](y) = \\ \lim_{n \rightarrow \infty} \tilde{b}[pl_b^\infty](z) &= \lim_{n \rightarrow \infty} \frac{0}{b^n(w)} = \lim_{n \rightarrow \infty} 0 = 0 \end{aligned}$$

in agreement with Theorem 3.

5.6 Combination of Plausibilities Versus Combination of Beliefs

It is crucial to notice that Theorem 1 (and by consequence Theorem 3) are about combination of *plausibility functions* (as pseudo b.f.s) and *not* combinations of belief functions. Hence, it is *not* true in general that $\widetilde{b^\infty} = (\tilde{b})^\infty$ or for that matters that commutativity holds. If we go back to the above example, it is straightforward to see that the combination $b \oplus b$ of b with itself has b.p.a.

$$\begin{aligned} m_{b \oplus b}(\{x, y\}) &= \frac{m_b(\{x, y\})m_b(\{x, y\})}{k(b, b)} = \frac{0.16}{0.68} = 0.235, \\ m_{b \oplus b}(\{y, z\}) &= \frac{m_b(\{y, z\})m_b(\{y, z\})}{k(b, b)} = \frac{0.16}{0.68} = 0.235, \\ m_{b \oplus b}(w) &= \frac{m_b(w)m_b(w)}{k(b, b)} = \frac{0.04}{0.68} = 0.058, \quad m_{b \oplus b}(y) \\ &= \frac{m_b(\{x, y\})m_b(\{y, z\}) + m_b(\{y, z\})m_b(\{x, y\})}{k(b, b)} = \frac{0.32}{0.68} = 0.47 \end{aligned}$$

which obviously yields

$$\widetilde{b \oplus b} = \left[0, \frac{0.47}{0.528}, 0, \frac{0.058}{0.528} \right]' \neq \tilde{b} \oplus \tilde{b} = [0, 0, 0, 1]'$$

The basic reason for this is that the plausibility function of a sum of two belief functions is *not* the sum of the associated plausibilities:

$$[pl_{b_1} \oplus pl_{b_2}] \neq pl_{b_1 \oplus b_2}.$$

6 Representation Theorem for Relative Beliefs

A dual of the representation theorem (Proposition 3) for relative beliefs can also be proven, once we recall a result on Dempster's sum of affine combinations [19].

Proposition 5. *The orthogonal sum $b \oplus \sum_i \alpha_i b_i$, $\sum_i \alpha_i = 1$ of a b.f. b with any² affine combination of b.f.s can be written as $b \oplus \sum_i \alpha_i b_i = \sum_i \gamma_i (b \oplus b_i)$, where*

² In fact the collection $\{b_i\}$ is required to include *at least* a b.f. which is combinable with b , [19].

$$\gamma_i = \frac{\alpha_i k(b, b_i)}{\sum_j \alpha_j k(b, b_j)} \quad (15)$$

and $k(b, b_i)$ is the normalization factor of the combination between b and b_i .

Theorem 5. *The relative belief of singletons \tilde{b} represents perfectly the corresponding plausibility function pl_b when combined with any probability through (extended) Dempster's rule:*

$$\tilde{b} \oplus p = pl_b \oplus p$$

for each Bayesian belief function $p \in \mathcal{P}$.

Theorem 5 can be obtained by replacing b with pl_b , and pl_b by \tilde{b} in Proposition 3. It is natural to suppose other properties of upper probabilities could in the future be found by analogous transformations of known propositions on lower probabilities, as a useful mathematical characterization of the relation between them.

7 Conclusions: Two Families of Bayesian Approximations

In this paper we studied the properties of the relative belief of singletons as a novel Bayesian approximation of a belief function, and discussed its interpretations and applicability. We proved that relative belief and plausibility of singletons form a distinct family of Bayesian approximations related to Dempster's rule, as they both commute with \oplus , and meet dual representation and idempotence properties. On one side, this suggests a new mathematical form of the duality which exists between upper and lower probabilities that can be used to prove new results. On the other side, once we recall that [10]

Proposition 6. *Both pignistic function $BetP[b]$ and orthogonal projection $\pi[b]$ commute with respect to affine combination:*

$$\pi \left[\sum_i \alpha_i b_i \right] = \sum_i \alpha_i \pi[b_i], \quad BetP \left[\sum_i \alpha_i b_i \right] = \sum_i \alpha_i BetP[b_i], \quad \sum_i \alpha_i = 1.$$

the present results bring about a subdivision of all Bayesian approximations in two families, related to Dempster's sum and affine combination respectively.

References

1. Shafer, G.: A mathematical theory of evidence. Princeton University Press, Princeton (1976)
2. Daniel, M.: On transformations of belief functions to probabilities. International Journal of Intelligent Systems, special issue on Uncertainty Processing 21(3), 261–282 (2006)
3. Kramosil, I.: Approximations of believability functions under incomplete identification of sets of compatible states. Kybernetika 31, 425–450 (1995)

4. Yaghlane, A.B., Denceux, T., Mellouli, K.: Coarsening approximations of belief functions. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 362–373. Springer, Heidelberg (2001)
5. Haenni, R., Lehmann, N.: Resource bounded and anytime approximation of belief function computations. International Journal of Approximate Reasoning 31(1-2), 103–154 (2002)
6. Bauer, M.: Approximation algorithms and decision making in the Dempster-Shafer theory of evidence—an empirical study. International Journal of Approximate Reasoning 17, 217–237 (1997)
7. Tessem, B.: Approximations for efficient computation in the theory of evidence. Artificial Intelligence 61(2), 315–329 (1993)
8. Smets, P.: Belief functions versus probability functions. In: Bouchon, B., Saitta, L., Yager, R. (eds.) Uncertainty and Intelligent Systems, pp. 17–24. Springer, Berlin (1988)
9. Smets, P.: Decision making in the TBM: the necessity of the pignistic transformation. International Journal of Approximate Reasoning 38(2), 133–147 (2005)
10. Cuzzolin, F.: Two new Bayesian approximations of belief functions based on convex geometry. IEEE Transactions on Systems, Man, and Cybernetics - Part B 37(4) (2007)
11. Voorbraak, F.: A computationally efficient approximation of Dempster-Shafer theory. International Journal on Man-Machine Studies 30, 525–536 (1989)
12. Dempster, A.: Upper and lower probabilities generated by a random closed interval. Annals of Mathematical Statistics 39, 957–966 (1968)
13. Cobb, B., Shenoy, P.: On the plausibility transformation method for translating belief function models to probability models. Int. J. Approx. Reasoning 41(3), 314–330 (2006)
14. Cuzzolin, F.: Semantics of the relative belief of singletons. In: Workshop on Uncertainty and Logic, Kanazawa, Japan, March 25–28 (2008)
15. Smets, P.: Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem. International Journal of Approximate reasoning 9, 1–35 (1993)
16. Aigner, M.: Combinatorial Theory. In: Classics in Mathematics. Springer, New York (1979)
17. Smets, P.: The canonical decomposition of a weighted belief. In: Proceedings of IJCAI 1995, Montréal, Canada, pp. 1896–1901 (1995)
18. Cuzzolin, F.: Geometry of upper probabilities. In: Proceedings of ISIPTA 2003 (2003)
19. Cuzzolin, F.: Geometry of Dempster’s rule of combination. IEEE Transactions on Systems, Man and Cybernetics part B 34(2), 961–977 (2004)

Appendix

Proof of Theorem 1

The basic plausibility assignment of $pl_1 \oplus pl_2$ is, according to (5),

$$\mu_{pl_1 \oplus pl_2}(A) = \frac{1}{k(pl_1, pl_2)} \sum_{X \cap Y = A} \mu_1(X)\mu_2(Y)$$

so that the corresponding relative belief of singletons $\tilde{b}[pl_1 \oplus pl_2](x)$ (11) is proportional to

$$\begin{aligned} m_{pl_1 \oplus pl_2}(x) &= \sum_{A \supseteq \{x\}} \mu_{pl_1 \oplus pl_2}(A) = \frac{\sum_{A \supseteq \{x\}} \sum_{X \cap Y = A} \mu_1(X)\mu_2(Y)}{k(pl_1, pl_2)} \\ &= \frac{\sum_{X \supseteq \{x\}} \mu_1(X)\mu_2(Y)}{k(pl_1, pl_2)} \end{aligned} \quad (16)$$

where $m_{pl_1 \oplus pl_2}(x)$ denotes the b.p.a. of the (pseudo) b.f. corresponding to the pl.f. $pl_1 \oplus pl_2$. As $\sum_{X \supseteq \{x\}} \mu_b(X) = m_b(x)$ by Lemma 1,

$$\tilde{b}[pl_1](x) \propto m_1(x) = \sum_{X \supseteq \{x\}} \mu_1(X), \quad \tilde{b}[pl_2](x) \propto m_2(x) = \sum_{X \supseteq \{x\}} \mu_2(X)$$

so that their Dempster's combination is

$$(\tilde{b}[pl_1] \oplus \tilde{b}[pl_2])(x) \propto \left(\sum_{X \supseteq \{x\}} \mu_1(X) \right) \left(\sum_{Y \supseteq \{x\}} \mu_2(Y) \right) = \sum_{X \cap Y \supseteq \{x\}} \mu_1(X)\mu_2(Y)$$

and by normalizing we get (16).

Proof of Theorem 3

Taking the limit on both sides of Equation (12) we get

$$\tilde{b}[pl_b^\infty] = (\tilde{b}[pl_b])^\infty. \quad (17)$$

Let us now focus on the quantity on the right hand side: $(\tilde{b}[pl_b])^\infty = \lim_{n \rightarrow \infty} (\tilde{b}[pl_b])^n$. Since $(\tilde{b}[pl_b])^n(x) = K(b(x))^n$ (where K is a constant independent on x) and x is the unique most believed state, it follows that

$$(\tilde{b}[pl_b])^\infty(x) = 1, \quad (\tilde{b}[pl_b])^\infty(y) = 0 \quad \forall y \neq x. \quad (18)$$

Hence by (17) $\tilde{b}[pl_b^\infty](x) = 1$, and $\tilde{b}[pl_b^\infty](y) = 0 \quad \forall y \neq x$.

Proof of Theorem 5

Once expressed a plausibility function in terms of its basic plausibility assignment (10) we can apply the commutativity property (Proposition 5), obtaining

$$pl_b \oplus p = \sum_{A \subseteq \Theta} \nu(A)p \oplus b_A \quad (19)$$

where

$$\nu(A) = \frac{\mu_b(A)k(p, b_A)}{\sum_{B \subseteq \Theta} \mu_b(B)k(p, b_B)}, \quad p \oplus b_A = \frac{\sum_{x \in A} p(x)b_x}{k(p, b_A)}$$

with $k(p, b_A) = \sum_{x \in A} p(x)$. Once replaced these expressions in (19) we get

$$pl_b \oplus p = \frac{\sum_{A \subseteq \Theta} \mu_b(A) \left(\sum_{x \in A} p(x)b_x \right)}{\sum_{B \subseteq \Theta} \mu_b(B) \left(\sum_{y \in B} p(y) \right)} = \frac{\sum_{x \in \Theta} p(x) \left(\sum_{A \supseteq \{x\}} \mu_b(A) \right) b_x}{\sum_{y \in \Theta} p(y) \left(\sum_{B \supseteq \{y\}} \mu_b(B) \right)} = \frac{\sum_{x \in \Theta} p(x)m_b(x)b_x}{\sum_{y \in \Theta} p(y)m_b(y)}$$

again by Lemma 1. But this is exactly $\tilde{b} \oplus p$, as a direct application of Dempster's rule (5) shows.

Alternative Formulations of the Theory of Evidence Based on Basic Plausibility and Commonality Assignments

Fabio Cuzzolin

Oxford Brookes University
fabiocuzzolin@gmail.com

Abstract. In this paper we introduce indeed two alternative formulations of the theory of evidence by proving that both plausibility and commonality functions share the same combinatorial structure of sum function of belief functions, and computing their Moebius inverses called basic plausibility and commonality assignments. The equivalence of the associated formulations of the ToE is mirrored by the geometric congruence of the related simplices. Applications to the probabilistic approximation problem are briefly presented.

1 Introduction

The *theory of evidence* (ToE) is one of the most popular uncertainty theory [1,2], even though it has been recently subjected to criticism [3]. Subjective probability is there represented by *belief function* (b.f.) rather than a Bayesian mass distribution, assigning probability values to *sets* of possibilities rather than single events. Variants or continuous extensions of the ToE in terms of hints [4] or allocations of probability [5] have since been proposed.

From a combinatorial point of view, in their finite incarnation, b.f.s are *sum functions*, i.e. functions on the power set $2^\Theta = \{A \subseteq \Theta\}$ of a finite domain Θ $b(A) = \sum_{B \subseteq A} m_b(B)$ induced by a *basic probability assignment* (b.p.a.) $m_b : 2^\Theta \rightarrow [0, 1]$ which is combinatorially the Moebius inverse [6] of b . The same evidence associated with a b.f. is carried by the related plausibility (pl.f.) $pl_b(A) = 1 - b(A^c)$ and commonality $Q_b(A) = \sum_{B \supseteq A} m_b(B)$ (comm.f.) functions, which lack though a similar coherent mathematical characterization.

In this paper we introduce indeed two alternative formulations of the theory of evidence by proving that both pl.f.s and comm.f.s share the same combinatorial structure of sum function, and computing their Moebius inverses which is natural to call *basic plausibility* and *commonality assignments*. We achieve this by resorting to a recent geometric approach to the theory of evidence [7] in which belief functions are represented by points of a Cartesian space. Besides giving the overall mathematical structure of the theory of evidence a more elegant symmetry, the notions of b.pl.a.s and b.comm.a.s turn out to be useful when solving problems like finding probabilistic approximations [8,9,10] of belief functions, or computing the canonical decomposition of support functions. Moreover, as they

are discovered through geometric methods, basic plausibility and commonality assignments inherit the same simplicial geometry as that of b.f.s.

The novel contributions of this paper are then the proofs that:

- commonality functions have a Moebius inverse that we call basic commonality assignment (Theorem 1), the study of its properties and geometries (Theorems 2 and 3);
- the equivalence of the alternative formulations of the ToE is geometrically mirrored by the congruence of the corresponding simplices (Theorem 4);

To support the usefulness of these alternative formulations, some applications of basic plausibility assignments to the approximation problem are discussed. We first recall the basic notions of the ToE and its geometric approach.

2 Belief, Plausibility, and Commonality Functions

Even though belief functions can be given several alternative but equivalent definitions in terms of multi-valued mappings, random sets [11,12], inner measures [13], in Shafer's formulation [1] a central role is played by the notion of "basic probability assignment". A *basic probability assignment* (b.p.a.) over a finite set (*frame of discernment* [1]) Θ is a function $m : 2^\Theta \rightarrow [0, 1]$ on its power set $2^\Theta = \{A \subset \Theta\}$ such that $m(\emptyset) = 0$, $\sum_{A \subseteq \Theta} m(A) = 1$, $m(A) \geq 0 \forall A \subset \Theta$. Subsets of Θ associated with non-zero values of m are called *focal elements*.

The *belief function* (b.f.) $b : 2^\Theta \rightarrow [0, 1]$ associated with a b.p.a. m_b is

$$b(A) = \sum_{B \subseteq A} m_b(B). \quad (1)$$

A finite probability or *Bayesian* belief function is a special b.f. assigning non-zero masses only to singletons : $m_b(A) = 0$, $|A| > 1$.

Functions of the form (1) on a partially ordered set are called *sum functions* [6]. A belief function b is then the sum function associated with a basic probability assignment m_b on the partially ordered set $(2^\Theta, \subseteq)$.

Conversely, the unique basic probability assignment m_b associated with a given belief function b can be recovered by means of the *Moebius inversion formula*

$$m_b(A) = \sum_{B \subseteq A} (-1)^{|A-B|} b(B). \quad (2)$$

A sum function can be seen as the discrete counterpart of the indefinite integral in calculus, and Moebius inversion as the discrete counterpart of the derivative.

A dual mathematical representation of the evidence encoded by a belief function b is the *plausibility function* (pl.f.) $pl_b : 2^\Theta \rightarrow [0, 1]$, $A \mapsto pl_b(A)$, where

$$pl_b(A) \doteq 1 - b(A^c) = 1 - \sum_{B \subseteq A^c} m_b(B) = \sum_{B \cap A \neq \emptyset} m_b(B)$$

expresses the amount of evidence *not against* A .

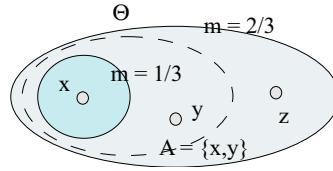


Fig. 1. The belief function of the example has two focal elements, $\{x\}$ and Θ

A third mathematical model of the evidence carried by a b.f. is represented by the *commonality function* (comm.f.) $Q_b : 2^\Theta \rightarrow [0, 1]$, $A \mapsto Q_b(A)$, where the *commonality number* $Q_b(A)$ can be interpreted as the amount of mass which can move freely through the entire event A ,

$$Q_b(A) \doteq \sum_{B \supseteq A} m_b(B). \quad (3)$$

Example. Let us consider a b.f. b on a frame of size 3, $\Theta = \{x, y, z\}$ with b.p.a. (see Figure 1) $m_b(x) = 1/3$, $m_b(\Theta) = 2/3$. The belief values of b on all possible events of Θ are (Eq. 1): $b(x) = m_b(x) = 1/3$, $b(y) = b(z) = 0$, $b(\Theta) = m_b(x) + m_b(\Theta) = 1$, $b(\{x, y\}) = m_b(x) = 1/3$, $b(\{x, z\}) = m_b(x) = 1/3$, $b(\{y, z\}) = 0$. To appreciate the difference between belief, plausibility, and commonality let us consider in particular the event $A = \{x, y\}$. Its belief value $b(\{x, y\}) = \sum_{A \subseteq \{x, y\}} m_b(A) = m_b(x) = 1/3$ represents the amount of evidence which *surely support* $\{x, y\}$ as it counts all the events which imply $\{x, y\}$. On the other side, $pl_b(\{x, y\}) = 1 - b(\{x, y\}^c) = 1 - b(z) = 1$ measures the evidence *not surely against* it, as it counts all the events which do not imply its complement $\{x, y\}^c$. Finally, the commonality number $Q_b(\{x, y\}) = \sum_{A \supseteq \{x, y\}} m_b(A) = m_b(\Theta) = 2/3$ tells us which is the amount of evidence which can (possibly) *equally support* each of the outcomes in $\{x, y\}$ (i.e. x and y), as the evidence represented by events $A \supseteq \{x, y\}$ can focus on both elements.

3 Two Alternative Formulations of the ToE

As plausibility and commonality functions are both equivalent representations of the evidence carried by a belief function, it is natural to guess that they should share the form of sum function on the power set 2^Θ .

We can indeed use results and tools provided by the geometric interpretation of the ToE to develop alternative models of uncertainty which are parallel to the standard formulation of the ToE. Evidence is there represented by cumulating basic probabilities on intervals of events $\{B \subseteq A\}$ (yielding a belief value $b(A) = \sum_{B \subseteq A} m(B)$). Equivalently we can represent pieces of evidence as basic plausibility (commonality) assignments on the power set, and compute the related plausibility (commonality) set function by adding basic assignments over intervals. Let us first recall the geometry of belief measures.

Belief space. A b.f. $b : 2^\Theta \rightarrow [0, 1]$ on a frame of discernment Θ is completely specified by its $N - 2$ belief values $\{b(A), \emptyset \subsetneq A \subsetneq \Theta\}$, $N \doteq 2^{|\Theta|}$ (since $b(\emptyset) = 0$, $b(\Theta) = 1$ always). It can then be represented as a point of \mathbb{R}^{N-2} like

$$b = \sum_{\emptyset \subsetneq A \subsetneq \Theta} b(A) \mathbf{v}_A$$

where $\{\mathbf{v}_A : \emptyset \subsetneq A \subseteq \Theta\}$ is a reference frame in \mathbb{R}^{N-2} . The set of points \mathcal{B} of \mathbb{R}^{N-1} which correspond to a b.f. is called "belief space" [7], i.e. the *simplex*

$$\mathcal{B} = Cl(b_A, \emptyset \subsetneq A \subseteq \Theta),$$

where b_A is the unique belief function assigning all the mass to a single subset A of Θ (A -th *dogmatic belief function*), and Cl denotes the convex closure operator: $Cl(b_1, \dots, b_k) = \{b \in \mathcal{B} : b = \alpha_1 b_1 + \dots + \alpha_k b_k, \sum_i \alpha_i = 1, \alpha_i \geq 0 \forall i\}$. The *faces* of a simplex $Cl(b_1, \dots, b_k)$ are all possible simplices generated by a subset of its vertices. Each b.f. $b \in \mathcal{B}$ can be written as a convex sum as follows:

$$b = \sum_{\emptyset \subsetneq A \subseteq \Theta} m_b(A) b_A. \quad (4)$$

A b.p.a. (the Moebius inverse of a belief function) is then the set of simplicial coordinates of b in \mathcal{B} : The simplicial form of \mathcal{B} is the geometric counterpart of the nature of b.f.s as sum functions. The set \mathcal{P} of all Bayesian b.f.s is the simplex formed by all dogmatic b.f.s associated with singletons: $\mathcal{P} = Cl(b_{\{x\}}, x \in \Theta)$.

Binary case. Consider as an example a frame of discernment with just two elements $\Theta_2 = \{x, y\}$. Each b.f. $b : 2^{\Theta_2} \rightarrow [0, 1]$ is completely determined by its belief values $b(x), b(y)$ (since $b(\emptyset) = 0$, $b(\Theta) = 1$ for all b). This means that we can represent b as the vector

$$b(x) \mathbf{v}_x + b(y) \mathbf{v}_y = [b(x), b(y)]' = [m_b(x), m_b(y)]' \in \mathbb{R}^2. \quad (5)$$

where $\mathbf{v}_x = [1, 0]'$ is the versor of the x axis, and $\mathbf{v}_y = [0, 1]'$ that of the y axis. Since $m_b(x) \geq 0$, $m_b(y) \geq 0$, and $m_b(x) + m_b(y) \leq 1$ the set \mathcal{B}_2 of all the possible belief functions on Θ_2 is the triangle in the Cartesian plane of Figure 2, whose vertices are the vacuous belief function $b_\Theta = [0, 0]'$ with $m_{b_\Theta}(\Theta) = 1$, the Bayesian b.f. $b_x = [1, 0]'$ with $m_{b_x}(x) = 1$, and the Bayesian b.f. $b_y = [0, 1]'$ with $m_{b_y}(y) = 1$. Bayesian b.f.s on Θ_2 obey the constraint $m_b(x) + m_b(y) = 1$ and form then the points of the segment \mathcal{P}_2 joining $b_x = [1, 0]'$ and $b_y = [0, 1]'$.

In the binary case each b.f. b decomposes according to Equation (4) as

$$b = m_b(x) b_x + m_b(y) b_y.$$

Change of reference frame. In the case of a general domain Θ , the dogmatic belief functions $\{b_A : \emptyset \subsetneq A \subsetneq \Theta\}$ form a set of independent vectors in \mathbb{R}^{N-2} , so that the collections $\{\mathbf{v}_A\}$ and $\{b_A\}$ represent two distinct coordinate frames in \mathcal{B} . We can then compute the transformation linking them [14].

Lemma 1. *The two coordinate frames $\{\mathbf{v}_A : \emptyset \subsetneq A \subsetneq \Theta\}$ and $\{b_A : \emptyset \subsetneq A \subsetneq \Theta\}$ are linked by the relation $\mathbf{v}_A = \sum_{B \supseteq A} (-1)^{|B \setminus A|} b_B$.*

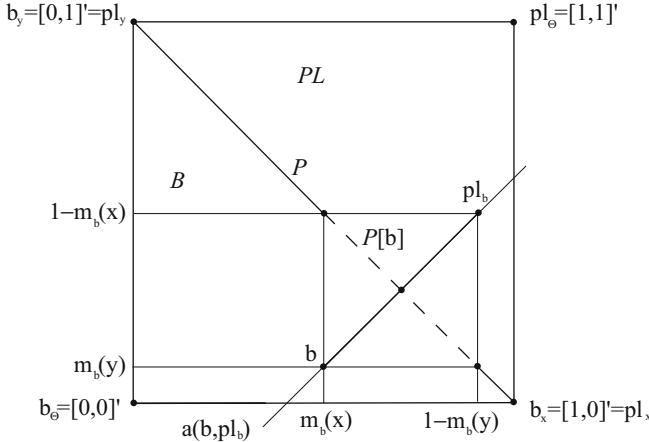


Fig. 2. The belief space \mathcal{B} for a binary frame is a triangle in \mathbb{R}^2 whose vertices are the dogmatic b.f.s focused on $\{x\}, \{y\}$ and $\Theta, b_x, b_y, b_\Theta$ respectively. The probability region is the segment $\mathcal{P} = Cl(b_x, b_y)$. Belief and plausibility functions lie on opposite locations with respect to \mathcal{P} . The line $a(b, pl_b)$ joining them intersect \mathcal{P} in the intersection probability $p[b]$ (Section 5).

3.1 Basic Plausibility Assignment

The geometry of belief measures can be exploited to prove the structure of sum function of both plausibility and commonality functions, establishing this way two equivalent formulations of the ToE in terms of basic plausibilities and commonalities. To get there we need to compute the Moebius inverse of pl.f.s and comm.f.s respectively.

Plausibility space. Plausibility functions are indeed also completely specified by their $N - 2$ plausibility values $\{pl_b(A), \emptyset \subsetneq A \subsetneq \Theta\}$ and can then be represented in the same reference frames as before as

$$pl_b = \sum_{\emptyset \subsetneq A \subsetneq \Theta} pl_b(A) \mathbf{v}_A \in \mathbb{R}^{N-2}. \quad (6)$$

It can be proved that [14]

Proposition 1. *The region \mathcal{PL} of \mathbb{R}^{N-2} whose points correspond to admissible pl.f.s is a simplex $\mathcal{PL} = Cl(pl_A, \emptyset \subsetneq A \subseteq \Theta)$ whose vertices are given by $pl_A = -\sum_{\emptyset \subsetneq B \subseteq A} (-1)^{|B|} b_B$, and represent the plausibility functions associated with all dogmatic belief functions $b_A: pl_A = pl_{b_A}$.*

Figure 2 shows the geometry of belief and plausibility spaces for a binary frame $\Theta_2 = \{x, y\}$, where pl.f.s are also vectors of \mathbb{R}^2 : $pl_b = [pl_b(x) = 1 - m_b(y), pl_b(y) = 1 - m_b(x)]'$. The two simplices $\mathcal{B} = Cl(b_\Theta = \mathbf{0}, b_x, b_y)$, $\mathcal{PL} = Cl(pl_\Theta = \mathbf{1}, pl_x = b_x, pl_y = b_y)$ are symmetric with respect to the segment of all probability

measures \mathcal{P} and congruent, so that they can be moved onto each other by means of a rigid transformation.

Plausibility assignment. We can use Lemma 1 to compute the Moebius inverse of a pl.f., by putting (6) in the same form as Equation (4). We get that $pl_b = \sum_{\emptyset \subsetneq A \subseteq \Theta} \mu_b(A)b_A$, where

$$\mu_b(A) \doteq \sum_{B \subseteq A} (-1)^{|A-B|} pl_b(B). \quad (7)$$

It is natural to call the function $\mu_b : 2^\Theta \rightarrow \mathbb{R}$ defined by expression (7) *basic plausibility assignment* (b.pl.a.). By comparing (7) with the Moebius formula for b.f.s (2) it is easy to recognize the Moebius equation for plausibilities: hence

$$pl_b(A) = \sum_{B \subseteq A} \mu_b(B). \quad (8)$$

PL.F.s are then sum functions on 2^Θ of the form (8), whose Moebius inverse is the b.pl.a. (7). Basic probabilities and plausibilities are obviously related.

Proposition 2. $\mu_b(A) = (-1)^{|A|+1} \sum_{C \supseteq A} m_b(C)$ for $A \neq \emptyset$, $\mu_b(\emptyset) = 0$.

As b.p.a.s do, basic plausibility assignments meet the normalization constraint. In other words, pl.f.s are *normalized sum functions* [6].

However, $\mu_b(A)$ is not always positive on all events $A \subseteq \Theta$.

Example. Let us consider as an example a b.f. b on the binary frame $\Theta_2 = \{x, y\}$ with b.p.a. $m_b(x) = \frac{1}{3}$, $m_b(\Theta) = \frac{2}{3}$. The corresponding pl. vector is

$$pl_b = [pl_b(x), pl_b(y)]' = [1 - b(\{x\}^c), 1 - b(\{y\}^c)]' = [1, 2/3]'.$$

Using Equation (7) we can compute its b.pl.a. as

$$\begin{aligned} \mu_b(x) &= (-1)^{|x|+1} \sum_{C \supseteq x} m_b(C) = (-1)^2 (m_b(x) + m_b(\Theta)) = 1, \\ \mu_b(y) &= (-1)^{|y|+1} \sum_{C \supseteq y} m_b(C) = (-1)^2 m_b(\Theta) = 2/3, \\ \mu_b(\Theta) &= (-1)^{|\Theta|+1} \sum_{C \supseteq \Theta} m_b(C) = (-1)m_b(\Theta) = -2/3 < 0 \end{aligned}$$

confirming that b.pl.a. meet the normalization but not the positivity constraint.

3.2 Basic Commonality Assignment

It is straightforward to prove that commonality functions are also sum functions and possess some interesting similarities with pl.f.s. They present though some peculiarities we need to take care of. We know that b.f.s and pl.f.s are such that

$$b(\emptyset) = pl_b(\emptyset) = 0, \quad b(\Theta) = pl_b(\Theta) = 1;$$

in other words, both b and pl_b can be represented by vectors with $N - 2$ coordinates as we have previously seen. On the other side

$$Q_b(\emptyset) = \sum_{A \supseteq \emptyset} m_b(A) = \sum_{A \subseteq \Theta} m_b(A) = 1, \quad Q_b(\Theta) = \sum_{A \supseteq \Theta} m_b(A) = m_b(\Theta)$$

so that Q_b needs N coordinates to be represented (even though the dimension of \mathcal{Q} is still $N - 2$). A comm.f. corresponds then to a vector of $\mathbb{R}^N = \mathbb{R}^{2^{|\Theta|}}$

$$Q_b = \sum_{\emptyset \subseteq A \subseteq \Theta} Q_b(A) \mathbf{v}_A$$

where $\{\mathbf{v}_A : \emptyset \subseteq A \subseteq \Theta\}$ is an extended reference frame in \mathbb{R}^N ($A = \Theta, \emptyset$ this time included).

Commonality assignment. We can as before express Q_b as a sum function by computing its Moebius inverse. We can use Lemma 1 to change the coordinate base and get the coordinates of Q_b with respect to the base $\{b_A, \emptyset \subseteq A \subseteq \Theta\}$:

$$\begin{aligned} Q_b &= \sum_{\emptyset \subseteq A \subseteq \Theta} Q_b(A) \left(\sum_{B \supseteq A} b_B (-1)^{|B \setminus A|} \right) \\ &= \sum_{\emptyset \subseteq B \subseteq \Theta} b_B \left(\sum_{A \subseteq B} (-1)^{|B \setminus A|} Q_b(A) \right) = \sum_{\emptyset \subseteq B \subseteq \Theta} q_b(B) b_B \end{aligned}$$

i.e. Q_b is a sum function with Moebius inverse $q_b : 2^\Theta \rightarrow [0, 1]$, $B \mapsto q_b(B)$ with

$$q_b(B) = \sum_{\emptyset \subseteq A \subseteq B} (-1)^{|B \setminus A|} Q_b(A)$$

which we can call *basic commonality assignment* (b.comm.a.).

q_b has an interesting interpretation in terms of belief values.

Theorem 1. $q_b(B) = (-1)^{|B|} b(B^c)$.

Proof

$$\begin{aligned} q_b(B) &= \sum_{\emptyset \subseteq A \subseteq B} (-1)^{|B \setminus A|} \left(\sum_{C \supseteq A} m_b(C) \right) = \sum_{\emptyset \subsetneq A \subseteq B} (-1)^{|B \setminus A|} \left(\sum_{C \supseteq A} m_b(C) \right) + \\ &+ (-1)^{|B| - |\emptyset|} \sum_{C \supsetneq \emptyset} m_b(C) = \sum_{B \cap C \neq \emptyset} m_b(C) \left(\sum_{\emptyset \subsetneq A \subseteq B \cap C} (-1)^{|B \setminus A|} \right) + (-1)^{|B|}. \end{aligned}$$

But now, since $B \setminus A = B \setminus C + B \cap C \setminus A$, we have that

$$\begin{aligned} \sum_{\emptyset \subsetneq A \subseteq B \cap C} (-1)^{|B \setminus A|} &= (-1)^{|B \setminus C|} \sum_{\emptyset \subsetneq A \subseteq B \cap C} (-1)^{|B \cap C| - |A|} \\ &= (-1)^{|B \setminus C|} [(1 - 1)^{|B \cap C|} - (-1)^{|B \cap C| - |\emptyset|}] = (-1)^{|B| + 1} \end{aligned}$$

so that the b.comm.a. $q_b(B)$ can be expressed as

$$q_b(B) = (-1)^{|B| + 1} \sum_{B \cap C \neq \emptyset} m_b(C) + (-1)^{|B|} = (-1)^{|B|} (1 - \sum_{B \cap C \neq \emptyset} m_b(C)) = (9)$$

$= (-1)^{|B|} (1 - pl_b(B))$ i.e. we have as desired. Note that $q_b(\emptyset) = (-1)^{|\emptyset|} b(\emptyset) = 1$.

Properties of basic commonality assignments. Basic commonality assignments are not normalized, as

$$\sum_{\emptyset \subseteq B \subseteq \Theta} q_b(B) = Q_b(\Theta) = m_b(\Theta).$$

In other words, whereas belief functions are normalized sum functions (n.s.f.) with non-negative Moebius inverse, and plausibility functions are normalized sum functions, commonality functions are *unnormalized* sum functions.

Going back to the above example, the b.comm.a. associated with $m_b(x) = 1/3$, $m_b(\Theta) = 2/3$ is (by Equation (9))

$$\begin{aligned} q_b(\emptyset) &= (-1)^{|\emptyset|} b(\Theta) = 1, & q_b(x) &= (-1)^{|x|} b(y) = -m_b(y) = 0, \\ q_b(y) &= (-1)^{|y|} b(x) = -m_b(x) = -1/3, & q_b(\Theta) &= (-1)^{|\Theta|} b(\emptyset) = 0 \end{aligned}$$

so that $\sum_{\emptyset \subseteq B \subseteq \Theta} q_b(B) = 2/3 = m_b(\Theta) = Q_b(\Theta)$.

Commonality space. Analogously to the case of belief and plausibility functions, we can use here the notion of basic commonality assignment (Theorem 1) to recover the shape of the space $\mathcal{Q} \subset \mathbb{R}^N$ of all commonality functions, or "commonality space".

Theorem 2. *The commonality space \mathcal{Q} is a simplex*

$$\mathcal{Q} = Cl(Q_A, \emptyset \subsetneq A \subseteq \Theta)$$

whose vertices are

$$Q_A \doteq \sum_{\emptyset \subseteq B \subseteq A^c} (-1)^{|B|} b_B. \quad (10)$$

Proof

$$\begin{aligned} Q_b &= \sum_{\emptyset \subseteq B \subseteq \Theta} (-1)^{|B|} b_B \left(\sum_{\emptyset \subseteq A \subseteq B^c} m_b(A) \right) = \sum_{\emptyset \subseteq A \subseteq \Theta} m_b(A) \left(\sum_{\emptyset \subseteq B \subseteq A^c} (-1)^{|B|} b_B \right) \\ &= \sum_{\emptyset \subseteq A \subseteq \Theta} m_b(A) Q_A \end{aligned}$$

with Q_A given by Equation (10). □

Theorem 3. *Q_A is the commonality function associated with the dogmatic belief function b_A , i.e.*

$$Q_{b_A} = \sum_{\emptyset \subseteq B \subseteq \Theta} q_{b_A}(B) b_B.$$

Proof. Indeed $q_{b_A}(B) = (-1)^{|B|}$ if $B^c \supseteq A$ i.e. $B \subseteq A^c$, while $q_{b_A}(B) = 0$ otherwise, so that $Q_{b_A} = \sum_{\emptyset \subseteq B \subseteq A^c} (-1)^{|B|} b_B = Q_A$ and the two quantities coincide. □

Binary case. In the binary case \mathcal{Q}_2 needs $N - 1 = 3$ coordinates to be represented. We have indeed $Q_b(\emptyset) = 1$, $Q_b(x) = \sum_{A \supseteq \{x\}} m_b(A) = pl_b(x)$, $Q_b(y) = \sum_{A \supseteq \{y\}} m_b(A) = pl_b(y)$, and $Q_b(\Theta) = m_b(\Theta)$.

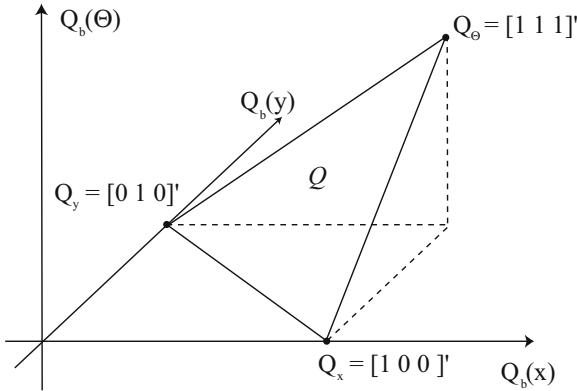


Fig. 3. Commonality space in the binary case

If we neglect the coordinate $Q_b(\emptyset)$ which is constant $\forall b$, the commonality space Q_2 can then be drawn as in Figure 3. The vertices of Q_2 are, according to Equation (10) and using all N coordinates, $Q_\Theta = b_\emptyset = [1111]',$

$$Q_x = \sum_{\emptyset \subseteq B \subseteq \{y\}} (-1)^{|B|} b_B = b_\emptyset - b_y = [1111]' - [0011]' = [1100]' = Q_{b_x}$$

$$Q_y = \sum_{\emptyset \subseteq B \subseteq \{x\}} (-1)^{|B|} b_B = b_\emptyset - b_x = [1111]' - [0101]' = [1010]' = Q_{b_y}.$$

4 Congruence of Equivalent Models

The *equivalence* of the three models based on basic probability, plausibility, and commonality assignments as descriptions of uncertainty geometrically translates as *congruence* of the associated simplices.

We saw that for binary frames, \mathcal{B} and \mathcal{PL} are congruent, i.e. they can be superposed by means of a rigid transformation. This is indeed a general property.

Lemma 2. *The corresponding 1-dimensional sides $Cl(b_A, b_B)$ and $Cl(pl_A, pl_B)$ of belief and plausibility spaces are congruent, namely*

$$\|pl_B - pl_A\|_p = \|b_A - b_B\|_p$$

where $\|\cdot\|_p$ denotes the classical norm $\|\mathbf{v}\|_p \doteq \sqrt{\sum_{i=1}^N |v_i|^p}$, for all $p = 1, 2, \dots, +\infty$.

Proof. This a direct consequence of the definition of plausibility function. Let us denote with C, D two generic subsets of Θ . As $pl_A(C) = 1 - b_A(C^c)$ we have $b_A(C^c) = 1 - pl_A(C)$, which implies

$$b_A(C^c) - b_B(C^c) = 1 - pl_A(C) - 1 + pl_B(C) = pl_B(C) - pl_A(C).$$

This in turn means that

$$\sum_{C \subset \Theta} |pl_B(C) - pl_A(C)|^p = \sum_{C \subset \Theta} |b_A(C^c) - b_B(C^c)|^p = \sum_{D \subset \Theta} |b_A(D) - b_B(D)|^p \quad \forall p.$$

A straightforward implication is then that

Theorem 4. \mathcal{B} and \mathcal{PL} are congruent.

as their corresponding 1-dimensional faces have the same length. This is due to the generalization of a well-known Euclid's theorem stating that triangles with sides of the same length are congruent.²

The situation is a bit more complicated for plausibility and commonality spaces, but we can still prove that \mathcal{Q} and \mathcal{PL} are congruent in the case of unnormalized belief functions [15].

5 Applications of Basic Plausibility Assignments

Besides being a natural complement to the mathematical apparatus of the theory of evidence, these alternative models of the ToE and the related basic assignments can actually be useful in the solution of practical problems. This is true when dealing with plausibility functions as we can recur to their equivalent basic assignments and operate on them. In particular, it becomes necessary when we need to apply combination rules for the aggregation of evidence to those plausibility functions.

Relative belief of singletons. The problem of approximating a given belief function with a probability, for instance, has been studied by many researchers [8,9,16]. The “relative plausibility of singletons”

$$\tilde{pl}_b(x) = \frac{pl_b(x)}{\sum_{y \in \Theta} pl_b(y)},$$

in particular, is an interesting candidate as it can be proven that it commutes with Dempster's combination \oplus [2,16] and it perfectly represents a belief function when combined with a probability: $pl_b \oplus p = b \oplus p$ for all $p \in \mathcal{P}$.

Definition 1. The Dempster's sum of two belief functions b_1, b_2 on the same frame Θ is a new belief function $b_1 \oplus b_2$ on Θ with b.p.a.

$$m_{b_1 \oplus b_2}(A) = \frac{\sum_{B \cap C = A} m_{b_1}(B) m_{b_2}(C)}{\sum_{B \cap C \neq \emptyset} m_{b_1}(B) m_{b_2}(C)} \quad (11)$$

where m_{b_i} denotes the b.p.a. associated with b_i .

² Note that this holds for *simplices* but not for *polytopes* in general, think of a square and a rhombus with sides of length 1.

However, as belief and plausibilities are dual representations of the same evidence, a dual probability can be defined as the *relative belief of singletons*

$$\tilde{b}(x) \doteq \frac{b(x)}{\sum_{y \in \Theta} b(y)}. \quad (12)$$

We can prove that \tilde{b} meets a set of dual properties with respect to \oplus , which are the dual of those of \tilde{pl}_b [9,16]. These dual properties involve the Dempster's sum of *plausibility functions* (instead of belief functions).

This should not surprise at this point. We have proven in Section 3.1 that plausibility functions are themselves sum functions, which admit a Moebius inverse: the basic plausibility assignment. But then nothing prevents from applying Equation (11) to the b.p.l.a.s of a pair of plausibility functions, instead of belief functions. We can then easily prove that

Proposition 3. *The relative belief of singletons \tilde{b} represents perfectly the corresponding plausibility function pl_b when combined with any probability through (extended) Dempster's rule: $\tilde{b} \oplus p = pl_b \oplus p \ \forall p \in \mathcal{P}$.*

Intersection probability. From a different point of view, each belief function determines an “interval probability”, i.e. a set of probability measures $p : \Theta \rightarrow [0, 1]$ on the same domain Θ which meet a lower bound associated with the belief values on all outcomes $x \in \Theta$, and an upper bound determined by the corresponding plausibility values:

$$(b, pl_b) \doteq \{b(x) \leq p(x) \leq pl_b(x), \forall x \in \Theta\}. \quad (13)$$

Now, there are clearly many ways of selecting one of those measures as representative of the above interval probability. However, as each interval $[b(x), pl_b(x)]$ has the same weight in the interval probability, there is no reason for the different singletons x to be treated differently.

Mathematically this translates into seeking the unique probability $p[b]$ such that

$$p[b](x) = b(x) + \alpha(pl_b(x) - b(x)), \quad \alpha \in [0, 1].$$

This function has been called *intersection probability* [17], as it is geometrically located on the segment joining a pair of belief-plausibility functions. The situation is clearly visible in the binary case of Figure 2, where the line $a(b, pl_b)$ joining such a pair is drawn: $p[b]$ lies at the intersection of this line with the region \mathcal{P} of all probability functions.

Again, as linear combination of a b.f. and a pl.f., its analysis requires the Moebius inversion of pl_b [17].

6 Conclusions

In this paper we introduced two alternative formulations of the theory of evidence by proving that both pl.f.s and comm.f.s share with belief functions the combinatorial structure of sum function, and computing their Moebius inverses which we

called basic plausibility and commonality assignments. From a combinatorial point of view, b.f.s, pl.f.s and comm.f.s form a hierarchy of sum functions whose Moebius inverse meets both normalization and positivity axiom (b.p.a.), only the normalization constraint (b.pl.a.), and none of them (b.comm.a.) respectively. The related spaces possess a similar convex geometry. Their congruence is the geometric reflection of the equivalence of those alternative formulations, which can be successfully applied to problems like the probabilistic approximation of a belief function.

References

1. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
2. Dempster, A.: Upper and lower probabilities generated by a random closed interval. *Annals of Mathematical Statistics* 39, 957–966 (1968)
3. Xiong, W., Luo, X., Ju, S.: An analysis of a defect in Dempster-Shafer theory. In: Proceedings of the 10th IEEE International Conference on Fuzzy Systems, pp. 793–796 (2001)
4. Kohlas, J.: Mathematical foundations of evidence theory. In: Coletti, G., Dubois, D., Scozzafava, R. (eds.) Mathematical Models for Handling Partial Knowledge in Artificial Intelligence, pp. 31–64. Plenum Press (1995)
5. Shafer, G.: Allocations of probability. *Annals of Probability* 7(5), 827–839 (1979)
6. Aigner, M.: Combinatorial Theory. In: Classics in Mathematics. Springer, Heidelberg (1979)
7. Cuzzolin, F.: A geometric approach to the theory of evidence. *IEEE Transactions on Systems, Man and Cybernetics - Part C* (2008)
8. Smets, P.: Decision making in the TBM: the necessity of the pignistic transformation. *International Journal of Approximate Reasoning* 38(2), 133–147 (2005)
9. Voorbraak, F.: A computationally efficient approximation of Dempster-Shafer theory. *International Journal on Man-Machine Studies* 30, 525–536 (1989)
10. Bauer, M.: Approximation algorithms and decision making in the Dempster-Shafer theory of evidence—an empirical study. *International Journal of Approximate Reasoning* 17, 217–237 (1997)
11. Nguyen, H.T.: On random sets and belief functions. *J. Mathematical Analysis and Applications* 65, 531–542 (1978)
12. Hestir, H., Nguyen, H., Rogers, G.: A random set formalism for evidential reasoning. In: Conditional Logic in Expert Systems, pp. 309–344. North-Holland, Amsterdam (1991)
13. Fagin, R., Halpern, J.Y.: Uncertainty, belief and probability. In: Proc. of IJCAI 1988, pp. 1161–1167 (1988)
14. Cuzzolin, F.: Geometry of upper probabilities. In: Proceedings of ISIPTA 2003 (2003)
15. Smets, Ph.: The nature of the unnormalized beliefs encountered in the transferable belief model. In: Proceedings of UAI 1992, p. 292. Morgan Kaufmann, San Mateo (1992)
16. Cobb, B., Shenoy, P.: On the plausibility transformation method for translating belief function models to probability models. *International Journal of Approximate Reasoning* 41(3), 314–330 (2006)
17. Cuzzolin, F.: Two new Bayesian approximations of belief functions based on convex geometry. *IEEE Transactions on Systems, Man and Cybernetics part B* 37(4), 993–1008 (2007)

Non-negative Sparse Principal Component Analysis for Multidimensional Constrained Optimization

Thanh D.X. Duong^{1,2} and Vu N. Duong^{2,3}

¹ Ton Duc Thang University, Vietnam

² University of Science, VNU-HCM, Vietnam

³ EUROCONTROL Experimental Center, France
`{thanhdxx@tut.edu.vn, dnvu@fit.hcmuns.edu.vn}`

Abstract. One classic problem in air traffic management (ATM) has been the problem of detection and resolution of conflicts between aircraft. Traditionally, a conflict between two aircraft is detected whenever the two protective cylinders surrounding the aircraft intersect. In Trajectory-based Air Traffic Management, a baseline for the next generation of air traffic management system, we suggest that these protective cylinders be deformable volumes induced by variations in weather information such as wind speed and directions subjected to uncertainties of future states of trajectory controls. Using contact constraints on deforming parametric surfaces of these protective volumes, a constrained minimization algorithm is proposed to compute collision between two deformable bodies, and a differential optimization scheme is proposed to resolve detected conflicts. Given the covariance matrix representing the state of aircraft trajectory and its control and objective functions, we consider the problem of maximizing the variance explained by a particular linear combination of the input variables where the coefficients in this combination are required to be non-negative, and the number of non-zero coefficients is constrained (e.g. state of trajectory and estimated time of arrival over one change point). Using convex relaxation and re-weighted l_1 technique, we reduce the problem to solving some semi-definite programming ones, and reinforce the non-negative principal components that satisfy the sparsity constraints. Numerical results show that the method presented in this paper is efficient and reliable in practice. Since the proposed method can be applied to a wide range of dynamic modeling problems such as collision avoidance in dynamic autonomous robots environments, dynamic interactions with 4D computer animation scenes, financial asset trading, or autonomous intelligent vehicles, we also attempt to keep all descriptions as general as possible.

Keywords: principal component analysis, semi-definite relaxation, semi-definite programming, l_1 -minimization, iterative reweighting.

1 Introduction

Principal component analysis (PCA) is a popular technique used to reduce multi-dimensional data sets to lower dimensions for analysis with applications throughout science and engineering, see [14]. This reduction is achieved by transforming to a new set of variables, the principal components, which are uncorrelated and ordered so that the first few retain most of the variation present in all of the original variables. PCA was first introduced by Pearson in [19], and developed independently by Hotelling in [9]. It can be performed via a singular value decomposition of the data matrix.

However, PCA has drawbacks since the principal components are usually linear combinations of all variables and the loadings are typically non-zero. This makes it often difficult to be applied in many applications where the principal components would be convenient if these components contained very few non-zero loadings. Besides the application to trajectory-based air traffic management [7], some other applications are financial asset trading strategies in which fewer non-zero loadings imply fewer transaction costs and gene expression data analysis where the sparsity is necessary for finding focalized local patterns hidden in the data, see [2].

Hence, it is desirable to study sparse principal components explaining most of the variance present in the data. To achieve this, it is necessary to sacrifice some of the explained variance and the orthogonality of the principal components. There are some approach to sparse PCA. Rotation techniques in [12] can be consider the first approach. In [22], the author studied simple principal components by restricting the loadings to take values from a small set of allowable integers such as 0, 1 and -1. Simple thresholding techniques [4] was an ad hoc way to deal with the problem, where the loadings with small absolute value are thresholded to zero. SCoTLASS [13] and SLRA [24,25] were introduced to get modified principal components with possible zero loadings. ESPCA [17] used discrete spectral formulation based on variational eigenvalue bounds and an effective greedy strategy to give provably optimal solutions via branch-and-bound search. For very large problems, SPCA [26] was proposed via a regression type optimization problem and DSPCA [6] via relaxing a hard cardinality constraint with a convex approximation.

But sparsity is still not enough for some applications where the nonnegativity property of the loadings are required. In particular, non-negative loadings increase efficiency of risk reduction in large portfolios, see [10], and is required due to the robustness of biological systems in [2]. Using matrix factorization approach, NSPCA in [23] studied PCA with both nonnegativity and sparsity properties. This method depends on two parameters - the first one is a balancing parameter between reconstruction and orthogonality and the second controls the amount of additional sparsity required. However, there is no algorithm designed for finding the suitable parameters.

By directly incorporating a sparsity criterion in the PCA problem formulation as in [6], we propose a direct approach improving the sparsity of the non-negative principal components - also called NSPCA. Then the problem is relaxed to be a

semi-definite program (SDP), which *can be solved efficiently in polynomial time* via interior-point methods [20,21]. We want to add a post-processing technique, since the outputs of all available approaches do not satisfy sparsity constraint - i.e. if we hope to find a principal component with less than k non-zero entries, the output often contains more than k non-zero entries. Re-weighted l_1 minimization [3,5] is recent technique to enhancing sparsity output of the combinatorial optimization:

$$\min \|x\|_{l_0} \quad \text{subject to } y = \Phi x,$$

where $\|x\|_{l_0} = |\{i : x_i \neq 0\}|$. Replacing the linear equation constraints in the above combinatorial optimization by linear matrix inequality constraints, we get an approach to *make the output of our NSPCA satisfying the sparsity constraint*. To our best knowledge, this is the first paper using re-weighted l_1 minimization technique with linear matrix inequality constraints.

This paper is organized as follows. The next section is the main results, where we present our NSPCA method by applying relaxing technique and re-weighted l_1 minimization technique. Section 3 is devoted to compare NSPCA with existing methods on both artificial data and real-life data.

Notation. In this paper, we denote the set of symmetric matrices of size n by \mathbf{S}^n , the vector of ones by $\mathbf{1}$, the cardinality (number of non-zero elements) of a vector x by $\mathbf{Card}(x)$, and the number of non-zero coefficients in a matrix X by $\mathbf{Card}(X)$. For $X \in \mathbf{S}^n$, the notation $X \succeq 0$ means that X is positive semi-definite, and $\|x\|_2$ is the 2-Euclidean norm for $x \in \mathbb{C}^n$.

2 Main Results

In this section, we derive an SDP relaxation for the problem of maximizing the variance explained by a non-negative vector while constraining its cardinality via re-weighted l_1 technique. Then, we apply the problem to decompose a data matrix into non-negative sparse factors.

2.1 Semi-definite Relaxation

Let $A \in \mathbf{S}^n$ be a covariance matrix, i.e. $A \succeq 0$, and k be an integer with $1 \leq k \leq n$. We consider the problem of maximizing the variance of a non-negative vector $x \in \mathbb{R}^n$ while constraining its cardinality:

$$\begin{aligned} & \text{maximize} && x^T A x, \\ & \text{subject to} && \|x\|_2 = 1, \\ & && \mathbf{Card}(x) \leq k, \\ & && x \geq 0. \end{aligned} \tag{1}$$

Let $X = xx^T \succeq 0$, the convex maximization objective $x^T A x$ and the non-convex constraint $\|x\|_2 = 1$ are transformed into a linear objective and a linear constraint since $\mathbf{Tr}(AX) = x^T A x$ and $\mathbf{Tr}(X) = \|x\|_2$; moreover, $\mathbf{Card}(X) =$

$\text{Card}(x)^2$. Hence, the lifting procedure for semi-definite relaxation, see [1,6,15,16], gives a equivalent problem of (1) as follows:

$$\begin{aligned} & \text{maximize} && \mathbf{Tr}(AX), \\ & \text{subject to} && \mathbf{Tr}(X) = 1, \\ & && \mathbf{Card}(X) \leq k^2, \\ & && X \geq 0, \\ & && X \succeq 0, \mathbf{rank}(X) = 1. \end{aligned} \tag{2}$$

Problem (1) is still non-convex. Hence, using classic technique to replace the non-convex cardinality constraint $\mathbf{Card}(X) \leq k^2$ with the weaker convex l_1 norm constraint $\mathbf{1}^T X \mathbf{1} \leq k$, see for example [3,8], and drop constraint $\mathbf{rank}(X) = 1$, we get a relaxation of (2) as follows:

$$\begin{aligned} & \text{maximize} && \mathbf{Tr}(AX), \\ & \text{subject to} && \mathbf{Tr}(X) = 1, \\ & && \mathbf{1}^T X \mathbf{1} \leq k, \\ & && X \geq 0, \\ & && X \succeq 0. \end{aligned} \tag{3}$$

It is important to remark that dropping constraint $\mathbf{rank}(X) = 1$ is truncation technique as in [1,15]. This means, we will solve the semi-definite problem (3) to get solution X , and an approximation solution of (1) is the non-negative parts of the dominant eigenvector of X .

2.2 Cardinality Constraint Refinement

Let x_* be the approximation solution of (1). It is clear that x_* does not satisfying cardinality constraint $\mathbf{Card}(x) \leq k$ in general. Hence, we consider the following cardinality constraint refinement problem:

$$\begin{aligned} & \text{minimize} && \mathbf{Card}(x), \\ & \text{subject to} && \|x\|_2 = 1, \\ & && x^T Ax \geq c_*, \\ & && x \geq 0, \end{aligned} \tag{4}$$

where $c_* := x_*^T Ax_*$.

By the same arguments as in the last subsection and setting, $X = xx^T \succeq 0$, we can relax the problem (4) as follows:

$$\begin{aligned} & \text{minimize} && \mathbf{Card}(X), \\ & \text{subject to} && \mathbf{Tr}(X) = 1, \\ & && \mathbf{Tr}(AX) \geq c_*, \\ & && X \geq 0, \\ & && X \succeq 0. \end{aligned} \tag{5}$$

Using the re-weighted l_1 minimization which is recent technique to enhancing sparsity, see [5], we consider the following relaxation of the problem (5):

$$\begin{aligned} & \text{minimize} && \mathbf{Tr}W^TX, \\ & \text{subject to} && \mathbf{Tr}(X) = 1, \\ & && \mathbf{Tr}(AX) \geq c_*, \\ & && X \geq 0, \\ & && X \succeq 0, \end{aligned} \tag{6}$$

where $W > 0$ is positive weight matrix. We use a simple iterative algorithm that alternates between estimating X and redefining the weights, see [5]:

1. Set the iteration count m to zero and $w_{ij}^{(0)} = 1$, $i, j = 1, \dots, n$.
2. Solve the weighted l_1 minimization problem (5) to get the solution $X^{(m)}$.
3. Update the weights: for each $i, j = 1, \dots, n$,

$$w_{ij}^{(m+1)} = \frac{1}{X_{ij}^{(m)} + \varepsilon} \tag{7}$$

4. Terminate when $\mathbf{Card}(x^{(m)}) \leq k$ or m attains a specified maximum number of iterations m_{max} , where $x^{(m)}$ is the dominant eigenvector of $X^{(m)}$. Otherwise, increment m and go to step 2.

The SDP (3) and (6) can be solved efficiently using interior-point solvers such as SEDUMI [20] or SDPT3 [21]. And we should set a *threshold* for expected non-zero-valued component X_{ij} of the solution of the problems (3) and (6). Moreover, the parameter $\varepsilon > 0$ in (7) should be chosen as $\varepsilon = \text{threshold}$ to provide stability and to ensure that a zero-valued component in X_{ij} does not strictly prohibit a non-zero estimate at the next step.

2.3 Sparse Decomposition

Let $A \in \mathbb{S}^n$ be a covariance matrix, we obtain a non-negative sparse PCA decomposition with target sparsity k as the following algorithm:

repeat

1. Solve the SDP (3) and (6) to get solution X .
 2. Let x be is the dominant eigenvector with non-negative largest entry of X . Add $\max\{x, 0\}$ to the solution set of non-negative sparse PCA decomposition.
 3. Update $A := A - (x^T Ax)xx^T$.
- until** $\max\{|A_{ij}| : i, j = 1, 2, \dots, n\} < \text{threshold}$ or the number of principle components attains a specified maximum number.

It is remarkable that the specified maximum number used to terminate the above algorithms should be $\text{rank}(A)$. Since, in PCA, the number of principle components is at most $\text{rank}(A)$, see for example [14].

3 Numerical Experiments

In this section, we compare the effectiveness of the proposed approach (NSPCA) with the other approaches mentioned in the introduction. First, we perform the test on an artificial data for all approaches - PCA, simple thresholding, SCoTLASS, SPCA, DSPCA, except for NSPCA in [23] since this approach depends on two parameters and there is no algorithm for finding the suitable parameters while the result for this data is also not available. Next, we consider a well-known real data set - Pit Props data, and restrict on the recent qualified approaches: SPCA, DSPCA and NSPCA in [23].

3.1 Artificial Data

To compare the result with that of existing algorithms, we consider the simulation example proposed by [26]. In this example, three hidden factors are first created

$$V_1 \sim N(0, 290), \quad V_2 \sim N(0, 300), \quad V_3 = 0.3V_1 + 0.925V_2 + \varepsilon, \quad \varepsilon \sim N(0, 1),$$

where V_1 , V_2 and ε are independent. Then 10 observed variables are generated as the follows

$$\begin{aligned} X_i &= V_1 + \varepsilon_i^1, & \varepsilon_i^1 &\sim N(0, 1), & i &= 1, 2, 3, 4, \\ X_i &= V_2 + \varepsilon_i^2, & \varepsilon_i^2 &\sim N(0, 1), & i &= 5, 6, 7, 8, \\ X_i &= V_3 + \varepsilon_i^3, & \varepsilon_i^3 &\sim N(0, 1), & i &= 9, 10, \\ \varepsilon_i^j &\text{ are independent, } & j &= 1, 2, 3, & i &= 1, \dots, 10. \end{aligned}$$

To avoid the simulation randomness, the exact covariance matrix which is an infinity amount of data generated from the above model is used to compute principal components using the different approaches. The variance of the three underlying factors is nearly the same (290, 300 and 283.8, respectively). Since the first two are associated with four variables while the last one is associated with only two variables, V_1 and V_2 are almost equally important, and they are both significantly more important than V_3 . In [26], the first two principal components explain 99.6% of the total variance. Hence, we shall choose the sparsity constraint is $k = 4$ when using only the variables X_1 , X_2 , X_3 , and X_4 to recover the factor V_1 , and only X_5 , X_6 , X_7 , and X_8 for the second sparse principal component to recover V_2 . In Table 1, we denote the simple thresholding method by 'ST', all the other methods - SPCA, DSPCA, SCoTLASS, and NSPCA - by 'Other', and the first and second principal components by PC_1 and PC_2 respectively. The results show that SPCA algorithms explains less variance than PCA as in Table 1, since the number of non-zero entries is constrained in SPCA. Similarly, NSPCA explains less variance than SPCA. However, in Table 1, NSPCA explains the same variance as SPCA, DSPCA, SCoTLASS and performs better than the simple thresholding method. The results here do not only illustrate the efficiency

Table 1. The first two principal components with k=4

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	<i>Explained variance</i>
PCA, PC_1	.12	.12	.12	.12	-.39	-.39	-.39	-.39	-.40	-.40	60.0%
PCA, PC_2	-.48	-.48	-.48	-.48	-.14	-.14	-.14	-.14	.01	.01	39.6%
ST, PC_1	0	0	0	0	0	0	-.5	-.5	-.5	-.5	38.8%
ST, PC_2	-.5	-.5	-.5	-.5	0	0	0	0	0	0	38.6%
Other, PC_1	0	0	0	0	.5	.5	.5	.5	0	0	40.9%
Other, PC_2	.5	.5	.5	.5	0	0	0	0	0	0	39.5%

Table 2. The first two principal components with k=5

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	<i>Explained variance</i>
DSPCA, PC_1	0	0	0	0	.49	.49	.49	.49	.14	.14	50.2%
DSPCA, PC_2	-.49	-.49	-.49	-.49	0	0	0	0	.14	.14	41.9%
NSPCA, PC_1	0	0	0	0	.47	.47	.47	.47	0	.38	49.7%
NSPCA, PC_2	.5	.5	.5	.5	0	0	0	0	0	0	39.5%

of SPCA methods but also show that the nonnegativity of the output does not rely on having non-negative input matrices to the process thereby *permitting zero-mean covariance matrices to be fed into the process of NSPCA just as being done with PCA or SPCA.*

Table 2 shows the output of DSPCA and NSPCA when sparsity constraint is $k = 5$, the sparsity constraint refinement process in NSPCA is performed with noise level being 10^{-2} . And there are two reasons for why NSPCA explains less variance than SPCA. The first is limitation to non-negative entries of the principal vectors in NSPCA, and the second is that sparsity constraint is forced to be hold - not as in SPCA. Even though, we hope to get principal components with atmost 5 non-zero entries, DSPCA gives the result with 6 non-zero entries with 92.1% explained variance. *NSPCA not only explains nearly the same variance - 89.2% - but also satisfies the sparsity constraint* when the first principal component has 5 non-zero entries after 27 sparsity constraint refinement iteration and the second principal component has 4 non-zero entries without any sparsity constraint refinement iteration.

3.2 Pit Props Data

The pit props data (consisting of 180 observations and 13 measured variables) was introduced in [11] and is another benchmark example used to test SPCA. All simple thresholding [4], SCOTLASS [13], SPCA [26], ESPCA [17], and DSPCA [6] have been tested on this data set. As reported in [26], SPCA performs better than SCOTLASS in the sense that it identifies principal components with 7, 4, 4, 1, 1, and 1 non-zero loadings respectively - while explaining nearly the same variance as SCOTLASS, the result SPCA of is much sparser; and better than simple thresholding in the sense that it explains more variance. As reported in

Table 3. The first six principal components of DSPCA with sparsity constraint 5, 2, 2, 1, 1, and 1

Variable	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6
topdiam	-0.56	0	0	0	0	0
length	-0.58	0	0	0	0	0
moist	0	0.71	0	0	0	0
testsg	0	0.71	0	0	0	0
ovensg	0	0	0	-1	0	0
ringtop	0	0	-0.79	0	0	0
ringbut	-0.26	0	-0.61	0	0	0
bowmax	-0.1	0	0	0	0	0
bowdist	-0.37	0	0	0	0	0
whorls	-0.36	0	0	0	0	0
clear	0	0	0	0	1	0
knots	0	0	0	0	0	1
diaknot	0	0	0.01	0	0	0
Number of non-zero loadings	6	2	3	1	1	1
Explained variance	26.6	14.48	13.15	7.69	7.69	7.69

Table 4. The first six principal components of NSPCA with sparsity constraint 5, 2, 2, 1, 1, and 1

Variable	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6
topdiam	0.48	0	0	0	0	0
length	0.50	0	0	0	0	0
moist	0	0.71	0	0	0	0
testsg	0	0.71	0	0	0	0
ovensg	0	0	0	0	0	0
ringtop	0	0	0.81	0	0	0
ringbut	0.40	0	0.58	0	0	0
bowmax	0	0	0	1.00	0	0
bowdist	0.42	0	0	0	0	0
whorls	0.43	0	0	0	0	0
clear	0	0	0	0	1.00	0
knots	0	0	0	0	0	1.00
diaknot	0	0	0	0	0	0
Number of refinement iteration	64	0	1115	0	0	0
Explained variance	26.2	14.48	12.20	7.69	7.69	7.69

[6], DSPCA performs better than SPCA in the sense that it identifies principal components with 6, 2, 3, 1, 1, and 1 non-zero loadings (with respect to sparsity constraint 5, 2, 2, 1, 1, and 1) as in Table 3 while also explaining nearly the same variance.

Here, we want to compare the results of NSPCA - using the same sparsity constraint (5, 2, 2, 1, 1, and 1) - with those of DSPCA and ESPCA. The results are given in Table 4 with noise level being 10^{-2} . While explaining 75.95%

Table 5. The first six principal components of NSPCA with sparsity constraint 4, 2, 2, 1, 1, and 1

Variable	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6
topdiam	0.54	0	0	0	0	0
length	0.55	0	0	0	0	0
moist	0	0.71	0	0	0	0
testsg	0	0.71	0	0	0	0
ovensg	0	0	0	0	0	0
ringtop	0	0	0.71	0	0	0
ringbut	0	0	0.71	0	0	0
bowmax	0	0	0	1.00	0	0
bowdist	0.46	0	0	0	0	0
whorls	0.43	0	0	0	0	0
clear	0	0	0	0	1.00	0
knots	0	0	0	0	0	1.00
diaknot	0	0	0	0	0	0
Number of refinement iteration	595	0	0	0	0	0
Explained variance	22.59	14.48	13.95	7.69	7.69	7.69

Table 6. The first six principal components of NSPCA with sparsity constraint 5, 2, 3, 1, 1, and 1

Variable	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6
topdiam	0.48	0	0	0	0	0
length	0.50	0	0	0	0	0
moist	0	0.71	0	0	0	0
testsg	0	0.71	0	0	0	0
ovensg	0	0	0.47	0	0	0
ringtop	0	0	0.71	0	0	0
ringbut	0.40	0	0.52	0	0	0
bowmax	0	0	0	1.00	0	0
bowdist	0.42	0	0	0	0	0
whorls	0.43	0	0	0	0	0
clear	0	0	0	0	1.00	0
knots	0	0	0	0	0	1.00
diaknot	0	0	0	0	0	0
Number of refinement iteration	64	0	912	0	0	0
Explained variance	26.20	14.48	14.19	7.69	7.69	7.69

variance - nearly the same as DSPCA (77.3%) and ESPCA (75.9%)- the results of NSPCA satisfies both the sparsity constraint and nonnegativity constraint. However, we can see that there is an overlap between the first principal component and the third principal component on entry "ringbut". Hence, it is reasonable to think about a better sparsity constraint as 4, 2, 2, 1, 1, and 1. The outputs for this case are shown in Table 5, where they also explain a large amount of the variance as 74.09%. Finally, with the less sparsity results (5, 2, 3, 1, 1, and 1) than

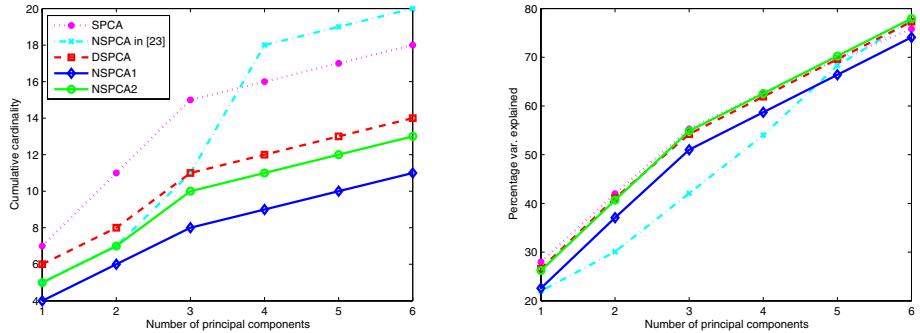


Fig. 1. Cumulative cardinality and percentage of total variance explained versus number of principal components, for SPCA, DSPCA, NSPCA in [23] and NSPCA with sparsity constraint (4, 2, 2, 1, 1, and 1) and (5, 2, 3, 1, 1, and 1) respectively on the pit props data

DSPCA , the results of NSPCA in Table 6 explain more variance than DSPCA (77.95% compared with 77.3%). Figure 1 shows the cumulative number of non-zero loadings and the cumulative explained variance. In this figure, we can observe that NSPCA with sparsity constraint (5, 2, 3, 1, 1, and 1) explains more variance than SPCA, DSPCA, and NSPCA in [23] while having the smallest cumulative cardinality.

4 Conclusions and Perspectives

The application specific solution will be discussed elsewhere since we want to keep our method general for other multi-dimensional constrained optimizations. In this paper, we attempted to present the Non-negative Sparse PCA method (NSPCA) to find the non-negative principal components not only explaining most of the variance present in the data but also satisfying sparsity constraints through the solving of semi-definite problems. The numerical results show that zero-mean covariance matrices can be fed into the process of NSPCA just as being done with PCA or SPCA and re-weighted l_1 minimization technique with linear matrix equality constraints is a useful tool to satisfying sparsity constraint. Hence, we do believe that this re-weighted l_1 minimization technique can be applied to others sparse PCA methods as well as other semi-definite problems containing sparsity constraint.

The drawback of NSPCA is that the SDP problems involved in (3) and (6) contain more than $O(n^2)$ constraints, which make the memory requirements of Newton's method prohibitive for very large-scale problems. This should be the subject of a future investigation by using smoothing technique, which has recently shown to be reducing memory requirements in solving large-scale SDP problems, see [6,18]. Finally, finding an efficient re-weighted function in (7) is also one of our priorities.

Acknowledgment

The authors acknowledge discussions with the colleagues in Information Technology and Applied Mathematic Department of Ton Duc Thang University related to the work presented in this paper. They are particularly grateful to the referees for their careful reading and helpful remarks.

References

1. Alizadeh, F.: Interior point methods in semi-definite programming with applications to combinatorial optimization. *SIAM J. Optim.* 5, 13–51 (1995)
2. Badea, L., Tilivea, D.: Sparse factorizations of gene expression guided by binding data. In: Pacific Symposium on Biocomputing (2005)
3. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press, Cambridge (2004)
4. Cadima, J., Jolliffe, I.T.: Loadings and correlations in the interpretation of principal components. *J. Appl. Statist.* 22, 203–214 (1995)
5. Candes, E.J., Wakin, M.B., Boyd, S.: Enhancing sparsity by re-weighted l_1 minimization (preprint)
6. D'Aspremont, A., El Ghaoui, L., Jordan, M.I., Lanckriet, G.R.G.: A direct formulation for sparse PCA using semi-definite programming. *SIAM Rev.* 49, 434–448 (2007)
7. Duong, V.: Dynamic models for airborne air traffic management capability: State-of-the-art analysis (Internal report). Eurocontrol Experimental Centre, Bretigny (1996)
8. Fazel, M., Hindi, H., Boyd, S.: A rank minimization heuristic with application to minimum order system approximation. In: Proceedings of the American Control Conference, Arlington, VA., vol. 6, pp. 4734–4739 (2001)
9. Hotelling, H.: Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* 24, 417–441 (1933)
10. Jagannathan, R., Ma, T.: Risk reduction in large portfolios: Why imposing the wrong constraints helps. *Journal of Finance* 58, 1651–1684 (2003)
11. Jeffers, J.: Two case studies in the application of principal components. *Appl. Statist.* 16, 225–236 (1967)
12. Jolliffe, I.T.: Rotation of principal components: Choice of normalization constraints. *J. Appl. Statist.* 22, 29–35 (1995)
13. Jolliffe, I.T., Trendafilov, N.T., Uddin, M.: A modified principal component technique based on the LASSO. *J.Comput. Graphical Statist.* 12, 531–547 (2003)
14. Jolliffe, I.T.: Principal component analysis. Springer, New York (2002)
15. Lemarechal, C., Oustry, F.: Semi-definite relaxations and lagrangian duality with application to combinatorial optimization. Rapport de recherche 3710, INRIA, France (1999)
16. Lovasz, L., Schrijver, A.: Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.* 1, 166–190 (1991)
17. Moghaddam, B., Weiss, Y., Avidan, S.: Spectral Bounds for Sparse PCA: Exact & Greedy Algorithms. In: Advances in Neural Information Processing Systems, vol. 18, pp. 915–922. MIT Press, Cambridge (2006)
18. Nesterov, Y.: Smoothing technique and its application in semi-definite optimization. *Math. Program.* 110, 245–259 (2007)

19. Pearson, K.: On lines and planes of closest fit to systems of points in space. *Phil. Mag.* 2, 559–572 (1901)
20. Sturm, J.: Using SEDUMI 1.0x, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.* 11, 625–653 (1999)
21. Toh, K.C., Todd, M.J., Tutuncu, R.H.: SDPT3 - a MA TLAB software package for semi-definite programming. *Optim. Methods Softw.* 11, 545–581 (1999)
22. Vines, S.: Simple principal components. *Appl. Statist.* 49, 441–451 (2000)
23. Zass, R., Shashua, A.: Non-negative Sparse PCA. In: *Advances In Neural Information Processing Systems*, vol. 19, pp. 1561–1568 (2007)
24. Zhang, Z., Zha, H., Simon, H.: Low-rank approximations with sparse factors I: Basic algorithms and error analysis. *SIAM J. Matrix Anal. Appl.* 23, 706–727 (2002)
25. Zhang, Z., Zha, H., Simon, H.: Low-rank approximations with sparse factors II: Penalized methods with discrete Newton-like iterations. *SIAM J. Matrix Anal. Appl.* 25, 901–920 (2004)
26. Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. *J. Comput. Graphical Statist.* 15, 265–286 (2006)

Sentence Compression by Removing Recursive Structure from Parse Tree

Seiji Egawa¹, Yoshihide Kato², and Shigeki Matsubara³

¹ Graduate School of Information Science, Nagoya University

² Graduate School of International Development, Nagoya University

egawa@el.itc.nagoya-u.ac.jp

³ Information Technology Center, Nagoya University

Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan

Abstract. Sentence compression is a task of generating a grammatical short sentence from an original sentence, retaining the most important information. The existing methods of removing the constituents in the parse tree of an original sentence cannot deal with recursive structures which appear in the parse tree. This paper proposes a method to remove such structure and generate a grammatical short sentence. Compression experiments have shown the method to provide an ability to sentence compression comparable to the existing methods and generate good compressed sentences for sentences including recursive structures, which the previous methods failed to compress.

Keywords: sentence compression, text summarization, phrase structure, recursive structure, maximum entropy method.

1 Introduction

Sentence compression is a task of summarizing a single sentence. It is useful for automatic text summarization and other applications such as generating subtitles or reducing messages for mobile devices.

Several sentence compression algorithms have been proposed so far. These algorithms produce a summary of a single sentence, which is called *compression*. Compression should satisfy the following conditions:

- It should be grammatical.
- It should retain the most important information of the original sentence.

In previous works, the problem of sentence compression have been simplified to removing redundant words or phrases from the original sentence. To generate a compression, the algorithms utilize syntactic information, such as phrase structure, dependency structure, part-of-speech and so on. Most of the algorithms only remove some redundant words or phrases, then the compression is a subsequence of the original sentence.

Knight and Marcu have proposed a probabilistic method of removing redundant constituents from the parse tree of the original sentence[2]. The probabilities

of removing constituents are estimated from a compression parallel corpus consisting of the pairs of original sentences and the corresponding compressions. Turner and Charniak have proposed an alternative method to approximate such probabilities without compression parallel corpora to overcome the lack of compression corpora[7]. Unno et al. have proposed a method of using maximum entropy method[1] so that more various features are dealt with, while Knight and Marcu have used only simple PCFG[8]. Vandeghinste and Pan have proposed a method of combining a probabilistic approach like above and a rule-based approach to avoid generating ungrammatical sentences[9].

These methods have only one operation of removing a constituent from a parse tree. However, the operation is not enough to compress any kind of sentences. The parse trees of some compressions have quite different structure from those of the original sentences so that it is impossible to obtain the compressed version of parse trees by removing constituents.

To solve the problem, this paper proposes an operation of transforming parse trees for sentence compression. We focus on recursive structures, which frequently appear in parse trees and represent adjuncts, coordinations, embedded sentences and so on. The operation removes recursive structures from the parse tree while preserving its grammaticality. Our method models sentence compression as a process of removing constituents and recursive structures from the parse tree of an original sentence. The model is probabilistic and learned from a compression parallel corpus. Our method can compress sentences including recursive structures.

Experimental results have shown that our method is comparable with the existing methods and that removing recursive structure from parse trees is effective for compressing certain sentences.

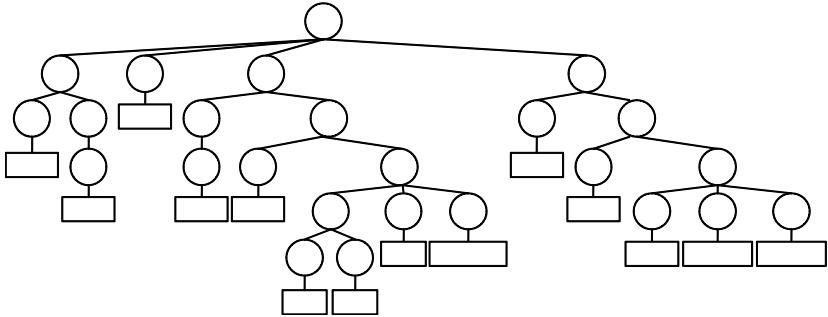
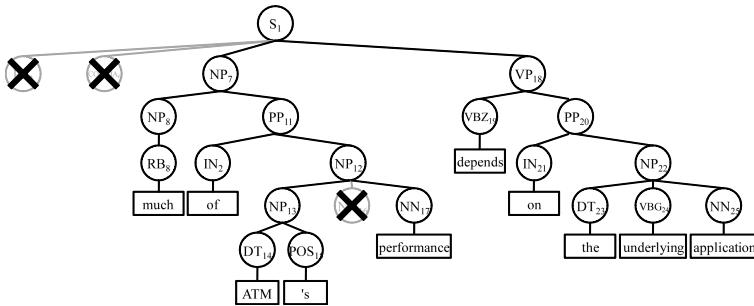
The organization of this paper is as follows: We review the previous methods of removing constituents in section 2. Section 3 describes our method which deal with recursive structures in parse trees for sentence compression. Section 4 presents some experimental evaluation of our method compared to the previous methods. Section 5 concludes this paper and presents future works.

2 Sentence Compression by Removing Constituents

In previous works, given an input sentence l , a compression s is formed by removing words from l . No rearranging words or no adding new words take place. The $2^{|l|}$ compression candidates exist and the problem of sentence compression can be formalized as determining which candidate is the best compression.

Knight and Marcu[2] tackled this problem by presenting a noisy channel model. The method finds the compression s which maximizes the conditional probability $P(s|l)$. The model $P(s|l)$ is decomposed into two models: the source model $P(s)$ and the channel model $P(l|s)$. That is, the compression s' is defined as follows:

$$s' = \operatorname{argmax}_s P(s|l) = \operatorname{argmax}_s P(s)P(l|s)$$

**Fig. 1.** Parse tree of sentence (1)**Fig. 2.** Parse tree of sentence (2) created from (1) by Knight and Marcu's method

The source model $P(s)$ evaluates the grammaticality of s . The channel model $P(l|s)$ determines which parts in l are redundant.

$P(s)$ and $P(l|s)$ are calculated based on the parse tree. As an example, let us consider the following original sentence (1) and its compression (2):

- (1) Like facelift, **much** of ATM's screen **performance depends on the underlying application.**
- (2) **Much of ATM's performance depends on the underlying application.**

The original sentence (1) is parsed into a tree shown in Fig. 1. The parse tree of compression (2) is created by removing some constituents from the original tree, that is, removing nodes “**PP₂**”, “**COMMA₆**” and “**NN₁₆**”. These nodes respectively correspond to word sequences “Like facelift”, “,” and “screen” which do not appear in compression (2). In this case, the probability $P(s)$ is high because the parse tree of (2) is grammatical.

$P(l|s)$ is learned from a compression parallel corpus consisting of pairs of sentences and compressions. A parse tree is assigned to every sentence and

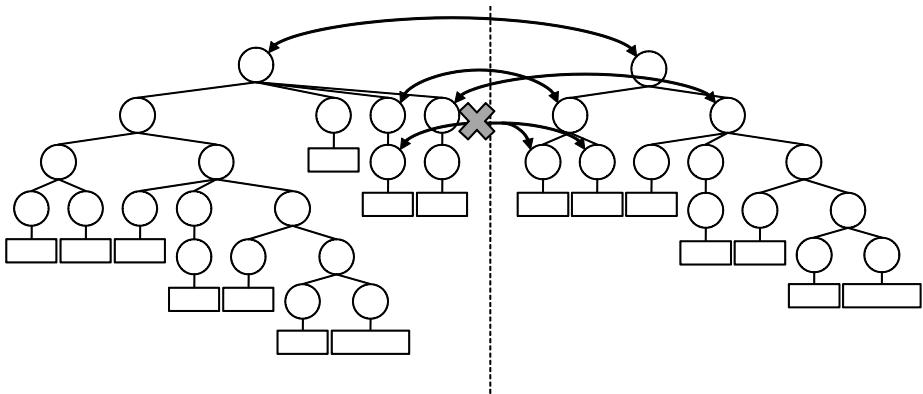


Fig. 3. Mismatch between parse trees of original sentence and compression

every compression. The method finds the correspondence between the nodes in the original parse tree and the compressed one in top-down fashion, and identifies the constituents removed from the original parse tree. For example, there exists a correspondence between the parse trees shown in Fig. 1 and Fig. 2, and nodes **PP₂**, **COMMA₆** and **NN₁₆** are identified with removed constituents.

As the following example shows, however, there is certain cases where the method cannot find the correspondence between original and compressed parse trees. (see Fig. 3).

- (3) **The user can then abort the transmission**, he said .
- (4) **The user can then abort the transmission**.

In this example, the method first finds the correspondence between “**S** → **NP VP**” in the compressed parse tree and “**S** → **S COMMA NP VP**” in the original parse tree. In the next stage, the method finds no correspondence because the child “**PRP**” of **NP** in the original parse tree does not match the children “**DT**” and “**NN**” in the compressed parse tree.

On the contrary, Unno et al.[8] have proposed a method for finding correspondences between both parse trees in a bottom-up fashion. The method parses only original sentences and extracts compressed parse trees from the original parse trees as in Fig. 4. Even though finding the correspondence always succeeds, the compressed parse trees become sometimes ungrammatical. Unno et al. directly estimate probabilities of removing constituents and do not evaluate the grammaticality of the compression.

As an example, let us consider a sentence (5) and its compression (6).

- (5) It is likely that a Macintosh version will be available soon.
- (6) **A Macintosh version will be available.**

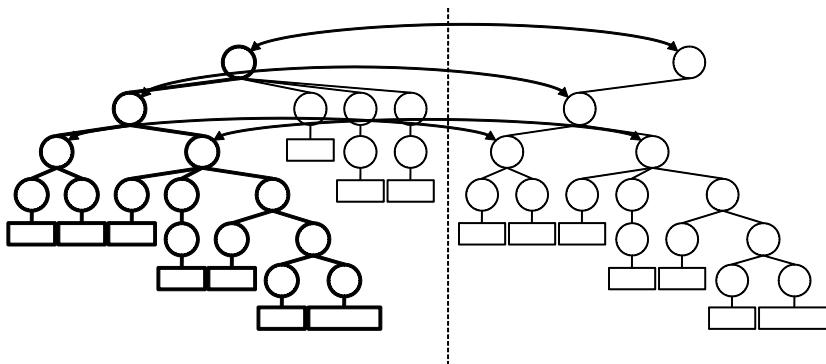


Fig. 4. Matching parse trees of original sentence and compression

The parse tree of (5) is shown in Fig.5. To obtain the compression, the method should remove nodes **NP₂**, **AUX₅**, **JJ₇**, **IN₉** and **ADVP₂₁**. Since the removal operations are assumed to be independent, it is difficult to compress such sentence. The same can be said for a sentence (7) and its compression (8).

- (7) **The CAKE in CAKEware** is an acronym which **stands for computer-assisted knowledge engineering**.
- (8) **The CAKE in CAKEware stands for computer-assisted knowledge engineering.**

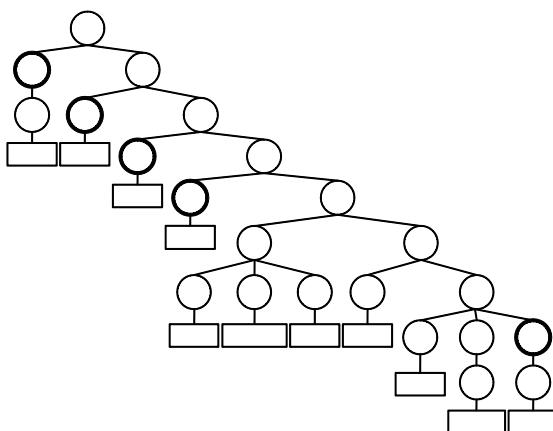


Fig. 5. Parse tree difficult to compress by previous methods

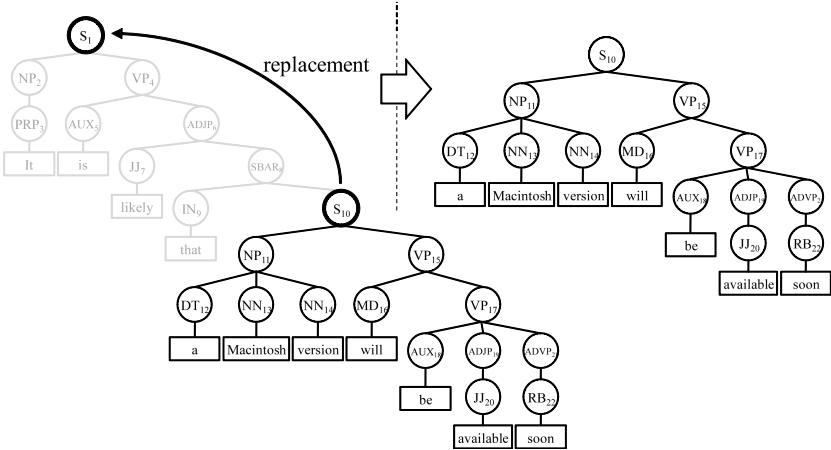


Fig. 6. Removing recursive structure for sentence compression

3 Method of Removing Recursive Structures

This section describes our algorithm for sentence compression. We introduce a new operation: removing recursive structures from parse trees. At first, we describe the basic idea of our approach.

As an example, let us consider the parse tree shown in Fig. 5. The parse tree has a recursive structure in which node S_1 includes node S_{10} . If we replace S_1 with S_{10} , we obtain a parse tree (see Fig. 6). The parse tree is grammatical because S_{10} plays the same syntactic role as S_1 . Our method introduces such operation. This operation can capture the dependence among removed constituents. For example, the operation captures the dependence between NP_2 , AUX_5 , JJ_7 and IN_9 removals, because they are removed by one operation.

In order to confirm the validity of this method, we investigated how often such operation occurs in human compression. It occurs 579 times in compressing 943 sentences which are included in the compression parallel corpus used in Knight and Marcu[2]. 110 operations out of 579 are particularly difficult to emulate for previous methods because there are some other nodes on the path from the ancestor node to the descendant node with the same syntactic category and the multiple nodes have a dependence.

Although the problem still remains whether important information of the original sentence is retained or not, it can be solved by training probabilities of removal operations from a compression parallel corpus.

3.1 Elementary Unit

This section gives some definitions for explanations of our method.

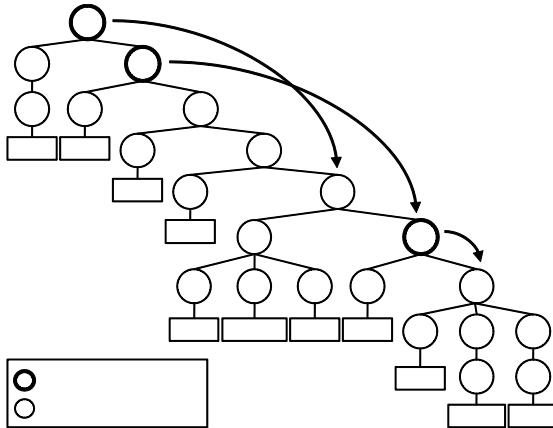


Fig. 7. Recursive node and non-recursive node

Definition 1 (Recursive Node). Let T be a parse tree, η be a node in T and X be the label of η . We call η recursive if there exists a node η' satisfying the following conditions:

1. η' is a descendant of η .
2. The label of η' is X .

We call η non-recursive if η is not recursive.

For example, there are three recursive nodes, S_1 , VP_4 and VP_{15} in Fig. 7. The node η' which is the nearest to η is called *foot*. The path from η to η' is called *minimal recursive path (MRP)*. We say that the root of the MRP is η . An MRP corresponds to a recursive structure in a parse tree. Fig. 8 shows the MRP whose root is S_1 .

3.2 Removing Elementary Units from Parse Tree

Our proposed algorithm removes constituents and MRPs from the parse tree of an input sentence to generate the compression. We use two types of operations:

removeConst operations remove a non-recursive node η and all descendants of η from parse tree.

removeMRP operations remove a recursive node η and all descendants of η , and replace the position of η with the foot of η

By applying these operations to the parse tree of the input sentence, we can obtain the compressed version of it. However, we need to choose the operations to generate a compression which is grammatically correct and preserves the important information of the original sentence. For this purpose, our method learns the process of applying operations from a compression parallel corpus.

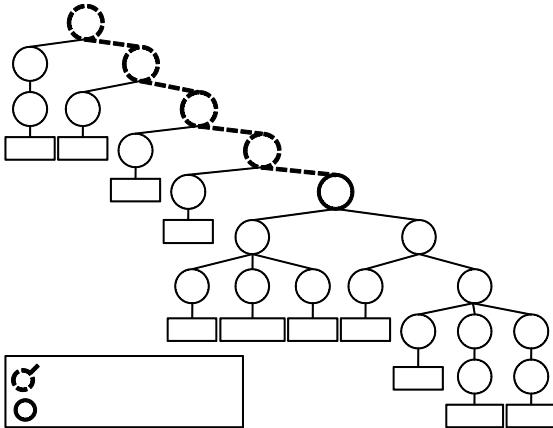


Fig. 8. Minimal recursive path

The compression parallel corpus consists of pairs of original sentences and their compressions. Our method first assigns the parse tree only to original sentences. For each pair of original parse tree and its compression, we determine which operations are applied to the parse tree. Next, we count the frequency of applying operations and estimate the probabilities.

Our method determines which operations are applied as follows: For each node in the parse tree, the operation is applied, if it does not remove any words in the compression. The operations are applied in a top-down fashion.

As an example, let us consider the input sentence (5) and its compression (6). Fig. 9 shows the parse tree of (5). For each terminal node, it is marked with bold line if it exists in the compression (6).

At first, the procedure tries to apply removeMRP since the root **S₁** is recursive. Because the word sequence “It is likely that”, which is removed by the operation, do not overlap the compression, so this operation is applied to **S₁**. Note that nodes **NP₂, PRP₃, …, IN₉** are removed by the operation to **S₁**. Next **S₁₀** is non-recursive. Applying removeConst, all words in the original tree are removed. Because some of these words exist in compression (6), this operation is not applied. For each node from **NP₁₁** to **NN₁₄**, removeConst operation is not applied for the same reason. **VP₁₅** is recursive. The removeMRP operation is not applied for this node because this operation deletes the word “will”, which appears in the compression (6). For each node from **MD₁₆** to **JJ₂₀**, which are non-recursive, the removeConst operations are not applied. **ADVP₂₁** is non-recursive. The removeConst operation is applied, since its descendant, “soon”, does not exist in the compression (6).

As the above, applying each corresponding operation to **S₁** and **ADVP₂₁**, we obtain the parse tree of (6). This tree (shown in Fig. 10) is grammatical as opposed to the one generated by the previous method.

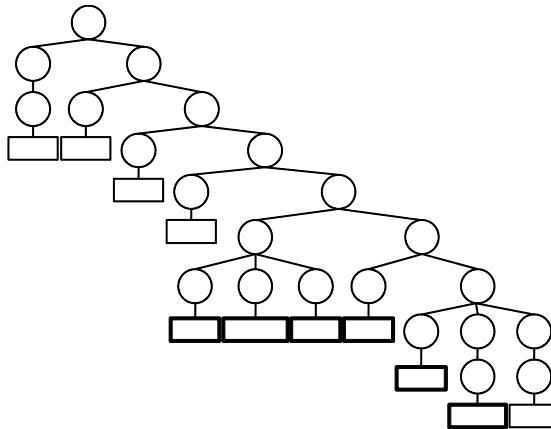


Fig. 9. Parse tree of sentence (5)

After determining, for each node, whether the operation is applied or not, we estimate its probability by maximum entropy method using the features:

- a. the removal operation type (removeConst or removeMRP)
 - b. the current node label
 - c. the parent node label
 - d. the daughter node labels
 - e. the left sibling node labels and which siblings are removed only if the operation type is removeConstj
 - f. the node labels on MRP
 - g. the daughter node labels of nodes on MRP
 - h. the foot node label

3.3 Probabilistic Sentence Compression Model

This section describes how to calculate the compression probability by using removal probabilities.

We define the probability of compressing a long sentence l to a short sentence s as the probability of generating the compressed version of the parse tree from the parse tree of l by removal operations. The probability is calculated by the product of the removing probabilities, that is,

$$P(s|l) = \prod_{\eta \in N} P(a_\eta | \eta, l)$$

where N is the set of nodes remaining in the parse tree of s or to which operations are applied. N does not have any node which is removed by applying a removal operation to an ancestor. a_η is 1 if an operation is applied to η and 0 if not.

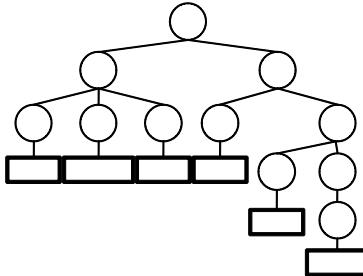


Fig. 10. Parse tree of sentence (6) by our method

For example, the probability of the compressing sentence (5) to the sentence (6) is $P(1|S_1) P(0|NP_{11}) P(0|DT_{12}) P(0|NN_{13}) P(0|NN_{14}) P(0|VP_{15}) P(0|MD_{16}) P(0|VP_{17}) P(0|AUX_{18}) P(0|ADJP_{19}) P(0|JJ_{20}) P(1|RB_{21})$. For simplicity, we abbreviate l .

3.4 Computing Scores

Using the model described in the previous section, we compute the compression score for every compression candidate s .

$$Score(s) = \text{length}(s)^\alpha \cdot \log P(s|l)$$

This score is proposed by Unno et al. and the compression model $P(s|l)$ is replaced with ours. α is a length parameter which controls the average length of outputs. Our method formalize the sentence compression problem as finding the compression which maximizes the score.

4 Experiments

To evaluate our algorithm, we conducted experiments. We use the compression parallel corpus used in Knight and Marcu[2]. This corpus consists of sentence pairs extracted from the Ziff-Davis corpus, which includes news articles on computer products. 32 sentence pairs in this corpus are used for evaluation in Knight and Marcu's experiment. We also use these sentences as a test set. Our model is trained on 943 sentences pairs, where each word in a compression corresponds to only one word in the original sentence, in the rest of the compression corpus.

4.1 Comparison with Original Results

At first, we compared our model with noisy-channel model in Knight and Marcu[2]. We evaluated both methods using four measures, compression rate,

Table 1. Comparison with Knight and Marcu

Method	compression	F-measure	bigram F-measure	BLEU
Knight and Marcu	70.4%	71.9%	58.5%	48.9%
Our method	50.7%	68.4%	58.5%	52.8%
Human	53.3%			

Table 2. Examples of compressions

Original	The user can then abort the transmission, he said.
Human	The user can then abort the transmission.
Knight	The user can abort the transmission said.
Unno	The user can then abort the transmission.
Our method	The user can then abort the transmission.
Original	It is likely that both companies will work on integrating multimedia with database technologies.
Human	Both companies will work on integrating multimedia with database technologies.
Knight	It is likely that both companies will work on integrating.
Unno	It is will work on integrating multimedia with database technologies.
Our method	Both companies will work on integrating multimedia with database technologies.
Original	A file or application "alias" similar in effect to the MS-DOS path statement provides a visible icon in folders where an aliased application does not actually reside.
Human	A file or application alias provides a visible icon in folders where an aliased application does not actually reside.
Knight	A similar in effect to MS-DOS statement provides a visible icon in folders where an aliased application does reside.
Unno	A file or application statement provides a visible icon in folders where an aliased application does not actually reside.
Our method	A file or application "alias" similar in effect to the MS-DOS path statement provides a visible icon in folders.

word F-measure, word bigram F-measure and BLEU score[6]. These measures except compression rate represent the similarity between sentences and we evaluate a compression with the degree of similarity to human compression. The BLEU score is a measure for machine translation quality. We used from unigram to 4-gram precisions for the BLEU score as in Unno et al.[8]. The value of the length parameter α for our method is determined by using 50 sentence pairs randomly extracted from training set. In this experiment, $\alpha = -0.43$.

The results are shown in Table 1. Our method achieved comparable accuracy with Knight and Marcu's method.

4.2 Examples of Compressions

Table 2 shows three sentences with compressions by human, the previous methods and our method. These sentences are used in the literature [8]. The first sentence is accurately compressed by a bottom-up method of Unno et al. while Knight and Marcu failed. Our method has also generated a correct compression. The second sentence has some recursive structures in its parse tree and both previous methods can not correctly compress it. Removing one of the recursive structures, Our method generated proper compression. Although all of the compressions generated for the third sentence are different from the one generated by human, our method seems to be superior to the others from the viewpoints of grammaticality and meaning.

5 Conclusion

We proposed a probabilistic method for sentence compression to remove recursive structures in the parse trees of original sentences. While recursive structures frequently appear in the parse tree, the previous methods do not deal with such the structure. Our method accurately compress such sentences applying a removal operation of the recursive structure. The experimental results show that our model has comparable power for sentence compression with other methods, and correctly compresses certain sentences which those methods cannot deal with. Evaluating our method only using three measures in this paper, we will evaluate our method by human judgments in terms of grammaticality and retention of important information.

Acknowledgements. The authors would like to thank Prof. Kevin Knight and Prof. Daniel Marcu for providing their parallel corpus and the experimental results. This research was partially supported by the Grant-in-Aid for Scientific Research (B) (No. 20300058) of JSPS.

References

1. Berger, A.L., Della Pietra, V.J., Della Pietra, S.A.: A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics* 22(1), 39–71 (1996)
2. Knight, K., Marcu, D.: Statistics-Based Summarization – Step One: Sentence Compression. In: AAAI/IAAI 2000, pp. 703–710. MIT Press, Cambridge (2000)
3. Knight, K., Marcu, D.: Summarization beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression. *Artificial Intelligence* 139, 91–107 (2002)
4. Mani, I.: Automatic Summarization. John Benjamins, Philadelphia (2001)
5. Nguyen, M.L., Horiguchi, S., Shimazu, A., Ho, T.B.: Example-Based Sentence Reduction Using the Hidden Markov Model. *ACM Trans. on Asian Language Information Processing* 3(2), 146–158 (2004)
6. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A Method for Automatic Evaluation of Machine Translation. In: ACL 2001, pp. 311–318. ACL, Morristown (2001)

7. Turner, J., Charniak, E.: Supervised and Unsupervised Learning for Sentence Compression. In: ACL 2005, pp. 290–297. ACL, Morristown (2005)
8. Unno, Y., Ninomiya, T., Miyao, Y., Tsujii, J.: Trimming CFG Parse Trees for Sentence Compression Using Machine Learning Approaches. In: COLING/ACL 2006, pp. 850–857. ACL, Morristown (2006)
9. Vandeghinste, V., Pan, Y.: Sentence Compression for Automated Subtitling: A Hybrid Approach. In: ACL 2004 Workshop on Text Summarization, pp. 89–95. ACL, Morristown (2004)

An ATP of a Relational Proof System for Order of Magnitude Reasoning with Negligibility, Non-closeness and Distance^{*}

Joanna Golińska-Pilarek^{1, **}, Angel Mora², and Emilio Muñoz-Velasco²

¹ Institute of Philosophy, Warsaw University
National Institute of Telecommunications, Poland
j.golinska@uw.edu.pl

² Dept. Matemática Aplicada. Universidad de Málaga, Spain
{amora,emilio}@ctima.uma.es

Abstract. We introduce an Automatic Theorem Prover (ATP) of a dual tableau system for a relational logic for order of magnitude qualitative reasoning, which allows us to deal with relations such as negligibility, non-closeness and distance. Dual tableau systems are validity checkers that can serve as a tool for verification of a variety of tasks in order of magnitude reasoning, such as the use of qualitative sum of some classes of numbers. In the design of our ATP, we have introduced some heuristics, such as the so called *phantom variables*, which improve the efficiency of the selection of variables used un the proof.

1 Introduction

Qualitative reasoning (QR) is the area of AI which provides an intermediate level between discrete and continuous models in order to develop representations for continuous aspects of the world, such as space, time, and quantity, without the kind of precise quantitative information needed by conventional analysis techniques [20].

A form of QR is to manage numerical data in terms of orders of magnitude, that is, to stratify values according to some notion of scale [7, 14, 16, 19]. Two approaches to order of magnitude reasoning have been identified in [20]: absolute order of magnitude, which is represented by a partition of the real line \mathbb{R} where each element of \mathbb{R} belongs to a qualitative class and relative order of magnitude, introducing a family of binary order of magnitude relations which establish different comparison relations in \mathbb{R} (e.g., *comparability*, *negligibility*, and *closeness*). In general, both models need to be combined to capture the relevant information.

The introduction of the logic formalism in QR tries to solve the problem about the soundness of the reasoning supported by the formalism and to give some

* Partially supported by Spanish projects TIN2006-15455-C03-01 and P6-FQM-02049.

** The author is a recipient of the 2007 and 2008 Grant for Young Scientists of the Foundation for Polish Science and is partially supported by the Polish Ministry of Science and Higher Education grant N N206 399134.

answers about the efficiency of using that. Several logics have been developed in different contexts, e.g., spatial and temporal reasoning [1, 17, 21]. In particular, logics dealing with order of magnitude reasoning have been developed in [3, 4, 5] by combining the absolute and relative approaches, that is, by defining different qualitative relations using the intervals provided by a specific absolute order of magnitude model.

In this paper, we focus our attention on the multimodal propositional logic $\mathcal{L}(OM)^{NCD}$ (from now on, OM for short) presented in [3], which introduces a sound and complete axiom system to deal with relations such as negligibility, non-closeness and distance.

We introduce an ATP for a relational proof system in the style of dual tableaux for the relational logic associated with OM, given in [10]. This system can be used as a tool for verification of a variety of tasks in order of magnitude reasoning, such as the use of qualitative sum of some classes of numbers. We emphasize, that the interaction between the theoretical study and the implementation of the ATP has contributed in a new style of proving the formulas by using deduction natural.

Our relational system, is based on the Rasiowa-Sikorski system for the first-order logic [18] extended to the classical relational logic originated in [15], following the ideas presented in [11]. Another approach to relational logics for order of magnitude reasoning has been given in [6].

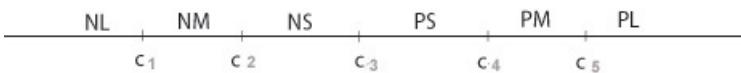
An implementation of the proof system for the classical relational logic is described in [8]. In [9] an implementation of translation procedures from non-classical logics to relational logic is presented. Moreover, in [2], there is an implementation of the system presented in [6].

The paper is organized as follows: In Section 2, we give a short presentation of the syntax, semantics, and the axiomatization of the logic OM, for more details see [3]. In Section 3, we give a survey of the relational logic appropriate for OM and its dual tableau system, presented in [10]. In Section 4, we show the details about the ATP with some examples and, finally, in Section 5, some conclusions and prospects future work are commented.

2 The Multimodal Logic OM

In this section, we summarize the logic OM introduced in [3]. We consider a strict linearly ordered set $(\mathbb{S}, <)^1$ divided into seven equivalence classes using five landmarks chosen depending on the context [20]. The cases with a different number of classes could be treated similarly.

The system corresponds to the following schematic representation, where $c_i \in \mathbb{R}$, being $i \in \{1, 2, 3, 4, 5\}$ such that $c_j < c_{j+1}$, for all $j \in \{1, 2, 3, 4\}$:



¹ For practical purposes, this set could be the real line.

The labels correspond, respectively, to the qualitative classes “negative large”, “negative medium”, “negative small”, “zero”, “positive small”, “positive medium”, and “positive large”.

The concepts of order of magnitude, non-closeness, distance and negligibility we consider in this paper introduce the ‘relative part’ of the approach, which builds directly on the ‘absolute part’ just presented.

First of all, we define the relation \vec{d}_α to give the intuitive meaning of a *constant distance*, called α . Let $(\mathbb{S}, <)$ be a strict linearly ordered set which contains the constants c_i , for $i \in \{1, 2, 3, 4, 5\}$ as defined above. Given $n \in \mathbb{N}$, we define \vec{d}_α^n as a relation on \mathbb{S} such that, for every $x, y, z, x', y' \in \mathbb{S}$ the following hold:

- (i) If $x \vec{d}_\alpha y$, then $x < y$
- (ii) $c_j \vec{d}_\alpha c_{j+1}$, for $j \in \{1, 2, 3, 4\}$
- (iii) If $x \vec{d}_\alpha y$ and $x \vec{d}_\alpha z$, then $y = z$
- (iv) If $x \vec{d}_\alpha y$, $x' \vec{d}_\alpha y'$ and $x < x'$ then $y < y'$

In the definition above, we assume for simplicity that every two consecutive constants are at the same distance, called α . This choice arises from the idea of taking α as the basic pattern for measuring. It could be easily generalized by assuming that the distance between two consecutive constants should be a multiple of α .

Now we define the remaining relations on \mathbb{S} . For every $x, y \in \mathbb{S}$ we define: $x \text{OM} y$ if and only if $x, y \in \text{EQ}$, where EQ denotes a qualitative class, that is, an element in the set $\{\text{NL}, \text{NM}, \text{NS}, \text{C}_0, \text{PS}, \text{PM}, \text{PL}\}$. Analogously, we define: $x \text{OM} y$ whenever x, y do not belong to the same class. The relations of *non-closeness* \vec{N} and *distance* \vec{D} , are defined as follows:

$$\begin{aligned} x \vec{N} y \text{ if and only if either } x \text{OM} y \text{ and } x < y \\ \text{or there exists } z \in \mathbb{S} \text{ such that } z < y \text{ and } x \vec{d}_\alpha z \\ x \vec{D} y \text{ if and only if there exist } z, z' \in \mathbb{S} \text{ such that } z < z' < y \text{ and } x \vec{d}_\alpha^2 z. \end{aligned}$$

Notice that $\vec{d}_\alpha^2 = \vec{d}_\alpha \circ \vec{d}_\alpha$, being \circ the usual composition of relations.

If we assume that \mathbb{S} is a set of real numbers, the intuitive interpretation of non-closeness relation is that x is non-close to y if, and only if, either x and y have not the same order of magnitude, or y is obtained from x by adding a medium or large number. On the other hand, x is distant from y if and only if y is obtained from x by adding large number.

In order to define the negligibility relation, note that it seems to be reasonable that if $x \neq c_3$ is *negligible* with respect to y , then x is distant to y .

Now, we can give the following definition for all $x, y \in \mathbb{S}$: x is *negligible* with respect to y (denoted by $x \vec{N} y$) if and only if either of the following holds:

$$(i) \quad x = c_3 \quad (ii) \quad x \in \text{NS} \cup \text{PS} \text{ and, either } y \vec{D} c_2 \text{ or } c_4 \vec{D} y.$$

Note that item (i) above corresponds to the intuitive idea that zero is negligible with respect to any real number and item (ii) corresponds to the intuitive idea

that a *sufficiently small* number is negligible with respect to any *sufficiently large* number, independently of the sign of these numbers. This definition ensures that if $x \neq c_3$ and $x \vec{N} y$, then either $y \vec{D} x$ or $x \vec{D} y$.

The relations of *non-closeness*, *distance* and *negligibility* can be defined in terms of $<$, \vec{d}_α , their inverses, and the constants c_i , for $i \in \{1, 2, 3, 4, 5\}$, for this reason, we only consider in our logic, connectives associated to these relations.

The syntax and semantics of OM are defined as usual in modal logics. We consider modal connectives $\vec{\Box}$, $\Box_{\vec{d}_\alpha}$ and $\vec{\Box}_{\vec{d}_\alpha}$, $\Box_{\vec{d}_\alpha}$ associated to the accessibility relations $<$, \vec{d}_α and their inverses, respectively. The intuitive meaning of the constants c_i is that c_i is true only in the constant c_i . The sound and complete axiom system of OM consists of all tautologies of classical propositional logic together with the following axiom schemata, being $i \in \{1, \dots, 5\}$ and $j \in \{1, \dots, 4\}$:

$$\begin{aligned} \mathbf{K1} \quad & \vec{\Box}(A \rightarrow B) \rightarrow (\vec{\Box}A \rightarrow \vec{\Box}B) \quad \mathbf{K2} \quad A \rightarrow \vec{\Box}\vec{\Diamond}A \quad \mathbf{K3} \quad \vec{\Box}A \rightarrow \vec{\Box}\vec{\Box}A \\ \mathbf{K4} \quad & (\vec{\Box}(A \vee B) \wedge \vec{\Box}(\vec{\Box}A \vee B) \wedge \vec{\Box}(A \vee \vec{\Box}B)) \rightarrow (\vec{\Box}A \vee \vec{\Box}B) \\ \mathbf{C1} \quad & \vec{\Diamond}c_i \vee c_i \vee \vec{\Diamond}c_i \quad \mathbf{C2} \quad c_i \rightarrow (\vec{\Box}\neg c_i \wedge \vec{\Box}\neg c_i) \quad \mathbf{d1} \quad \vec{\Box}A \rightarrow \Box_{\vec{d}_\alpha}A \quad \mathbf{d2} \quad \Diamond_{\vec{d}_\alpha}A \rightarrow \Box_{\vec{d}_\alpha}A \\ \mathbf{d3} \quad & (\Diamond_{\vec{d}_\alpha}A \wedge \vec{\Diamond}\Diamond_{\vec{d}_\alpha}B) \rightarrow \vec{\Diamond}(A \wedge \vec{\Diamond}B) \quad \mathbf{d4} \quad c_j \rightarrow \Diamond_{\vec{d}_\alpha}c_{j+1} \\ \mathbf{d5} \quad & \Box_{\vec{d}_\alpha}(A \rightarrow B) \rightarrow (\Box_{\vec{d}_\alpha}A \rightarrow \Box_{\vec{d}_\alpha}B) \quad \mathbf{d6} \quad A \rightarrow \Box_{\vec{d}_\alpha}\Diamond_{\vec{d}_\alpha}A \end{aligned}$$

The corresponding mirror images of **K1–K4** and **d1–d6** are also considered as axioms. We also consider the rules of inference as usual in modal logic.

3 Relational Formalization of OM

This section summarizes the more important concepts about relational logics needed to obtain the relational formalization of our logic, for more details, see [10, 11, 15].

The language of the logic RL_{OM} appropriate for expressing OM-formulas consists of the following pairwise disjoint sets of symbols:

$$\begin{aligned} \text{OV} &= \{x, y, z, \dots\} - \text{a countably infinite set of object variables;} \\ \text{OC} &= \{c_i : i \in \{1, \dots, 5\}\} - \text{the set of object constants;} \\ \text{RV} &= \{P, Q, \dots\} - \text{a countably infinite set of binary relational variables;} \\ \text{RC} &= \{1, 1', <, \vec{d}_\alpha\} \cup \{\Psi_i : i \in \{1, \dots, 5\}\} - \text{the set of relational constants }^2; \\ \text{OP} &= \{-, \cup, \cap, ;, ^{-1}\} - \text{the set of relational operation symbols.} \end{aligned}$$

The set of *relational terms* RT is the smallest set of expressions including the set $\text{RV} \cup \text{RC}$ of *atomic* terms and closed with respect to the operation symbols from OP . The set of RL_{OM} -formulas (or, simply formulas if it is clear from the context), consists of expressions of the form xPy where $x, y \in \text{OS} = \text{OV} \cup \text{OC}$ and $P \in \text{RT}$.

The semantics of RL_{OM} can be given as usual in relational logic, by using the previous definitions of our accessibility relations and constants. The respective

² 1 and 1' represent, respectively, the universal and equality relations.

semantics of OM and RL_{OM} give us the concepts of OM-validity and RL_{OM} -validity. Now, we define a translation function in order to have a relationship between these concepts. The translation of OM-formulas into relational terms starts with a one-to-one assignment of relational variables to the propositional variables, called τ' . Then the translation τ of OM-formulas is defined inductively as follows, being ; and – the composition and opposite of relations, respectively:

$$\tau(p) = \tau'(p); 1, \text{ for every propositional variable } p.$$

$$\tau(c_i) = \Psi_i; 1, \text{ for every } i \in \{1, \dots, 5\}$$

τ extends to all compound OM-formulas as follows ³:

$$\begin{array}{lll} \tau(\neg\varphi) = -\tau(\varphi) & \tau(\varphi \vee \psi) = \tau(\varphi) \cup \tau(\psi) & \tau(\varphi \wedge \psi) = \tau(\varphi) \cap \tau(\psi) \\ \tau(\varphi \rightarrow \psi) = -\tau(\varphi) \cup t(\psi) & \tau(\overrightarrow{\Box}\varphi) = -(<; -\tau(\varphi)) & \tau(\Box_{\overrightarrow{d_\alpha}}\varphi) = -(d_\alpha; -\tau(\varphi)) \end{array}$$

The following theorem shows the semantical relationship between OM and RL_{OM} :

Theorem 1. *For every OM-formula φ and for all object variables x and y , φ is OM-valid iff $x\tau(\varphi)y$ is RL_{OM} -valid.*

Dual tableau systems are determined by axiomatic sets of formulas and rules which apply to finite sets of formulas. The axiomatic sets take the place of axioms. There are two groups of rules: the *decomposition rules* which reflect definitions of the standard relational operations and the *specific rules* which reflect the properties of the specific relations assumed in RL_{OM} -models. The rules are of the form $\frac{\Phi}{\Phi_1 | \dots | \Phi_n}$, where Φ_1, \dots, Φ_n are finite non-empty sets of formulas, $n \geq 1$, and Φ is a finite (possibly empty) set of formulas. Φ is called the *premise* of the rule, and Φ_1, \dots, Φ_n are called its *conclusions*. A rule is said to be *applicable* to a set X of formulas whenever $\Phi \subseteq X$. As a result of an application of a rule to a set X , we obtain the sets $(X \setminus \Phi) \cup \Phi_i$, $i = 1, \dots, n$.

We say that an object variable in a rule is *new* whenever it appears in a conclusion of the rule and does not appear in its premise.

Decomposition rules of RL_{OM} -dual tableau have the following forms, for all object symbols $x, y \in \text{OS}$ and for all relational terms $P, Q \in \text{RT}$, where z is any object symbol and w is a new object variable:

$$\begin{array}{lllll} (\cup) & \frac{x(P \cup Q)y}{xPy, xQy} & (-\cup) & \frac{x-(P \cup Q)y}{x-Py \mid x-Qy} & (\cap) & \frac{x(P \cap Q)y}{xPy \mid xQy} & (-\cap) & \frac{x-(P \cap Q)y}{x-Py, x-Qy} \\ (-) & \frac{x--Py}{xPy} & (-^1) & \frac{xP^{-1}y}{yPx} & (-^{-1}) & \frac{x-P^{-1}y}{y-Px} \\ (;) & \frac{x(P; Q)y}{xPz, x(P; Q)y \mid zQy, x(P; Q)y} & z (-;) & \frac{x-(P; Q)y}{x-Pw, w-Qy} \end{array}$$

³ The translation of the inverse formulas is trivial.

Specific rules of RL_{OM} -dual tableau have the following forms, for all object symbols $x, y \in \text{OS}$, for every atomic relational term R , and for every $i \in \{1, \dots, 5\}$, where z, v are any object symbols:

$$\begin{array}{c}
 (1'1) \quad \frac{xRy}{xPz, xPy \mid y1'z, xPy} \quad (1'2) \quad \frac{xRy}{x1'z, xPy \mid zPy, xPy} \\
 (\text{Irref}<) \quad \underline{\underline{x < x}} \qquad \qquad (\text{Tran}<) \quad \frac{x < y}{x < y, x < z \mid x < y, z < y} \\
 (C_i1) \quad \frac{}{x\Psi_i y \mid x - \Psi_i y} \qquad (C_i2) \quad \frac{x\Psi_i y}{x\Psi_i y, x1'c_i} \qquad (C_i3) \quad \frac{x - \Psi_i y}{x - \Psi_i y, x - 1'c_i} \\
 (D1) \quad \frac{x < y}{xd_\alpha y, x < y} \qquad (D2) \quad \frac{x1'y}{zd_\alpha x, x1'y \mid zd_\alpha y, x1'y} \\
 (D3) \quad \frac{x < y}{zd_\alpha x, x < y \mid vd_\alpha y, x < y \mid z < v, x < y}
 \end{array}$$

A finite set of RL_{OM} -formulas is said to be an RL_{OM} -axiomatic set whenever it includes either of the following subsets, for any $x, y \in \text{OS}, R \in \text{RT}, i \in \{1, \dots, 4\}$:

$$(Ax1) \quad \{x1'x\}, (Ax2) \quad \{x1y\}, (Ax3) \quad \{xRy, x - Ry\}, (Ax4) \quad \{c_i d_\alpha c_{i+1}\}, (Ax5) \quad \{x < y, y < x, x1'y\}.$$

An RL_{OM} -proof tree for a formula xPy is a tree with the following properties:

- xPy is at the root of this tree;
- each node except the root is obtained by an application of an RL_{OM} -rule to its predecessor node;
- a node does not have successors whenever it is an RL_{OM} -axiomatic set.

Due to the forms of the rules for atomic formulas, if a node of an RL_{OM} -proof tree contains an RL_{OM} -formula xPy or $x - Py$, for some atomic P , then all of its successors contain this formula as well.

A branch of an RL_{OM} -proof tree is said to be *closed* whenever it contains a node with an RL_{OM} -axiomatic set of formulas. A *closed tree* is an RL_{OM} -proof tree such that all of its branches are closed. A formula xPy is RL_{OM} -provable whenever there is a closed proof tree for xPy , which is then referred to as an RL_{OM} -proof of xPy .

The following main result ensures the correspondence between OM-validity and RL_{OM} -provability.

Theorem 2 (Soundness and Completeness). *Let φ be an OM-formula. Then for all object variables x and y , φ is OM-valid iff $x\tau(\varphi)y$ is RL_{OM} -provable.*

4 The ATP

We show in broad strokes the implementation realized in Prolog⁴ of an ATP for obtained an automatic Rasiowa-Sikorski proof system associated to the relational translation RL_{OM} of the multimodal logic of qualitative order of magnitude reasoning OM.

⁴ See <http://www.matap.uma.es/~emilio/omr.zip>, for a revision of the ATP.

We have represented the formula $x_m R_i y_n$ as the Prolog fact: $rel([1], R_i, x_m, y_n)$. Node [1] denotes the root of the proof tree that the Prolog tool develops when it applies the rules of the RL_{OM} .

Example 1. The union of expressions $xRy \cup x - (\vec{d}_\alpha; -(a; 1))y$ is translated to the following facts in Prolog:

```
rel([1], r, x, y).
rel([1], opposite(comp(dalpha, opposite(comp(a, universal)))), x, y).
```

Prolog knows the leaf in which it must apply any rule, because the Prolog predicate $\text{leaves}([[1, \dots, 1], \dots, [1, \dots, k]])$ stores the leaves that the tool must close. Prolog will try to satisfy the relations in the leaf nodes. If the tool can close all the leaves in the tree, then *formula* is true.

The rules in RL_{OM} have the following general form: $\frac{\Phi}{\Phi_1 | \dots | \Phi_n}$ where Φ_1, \dots, Φ_n are non-empty set of formula and Φ is a finite (possibly finite) set of formula.

Let X a set of formulas, and if $\Phi \subseteq X$ then, as said before, the system transform Φ in $X \setminus \Phi \cup \Phi_i, i = 1 \dots, n$. That's to say , if X is represented in the leaf $[i_1, i_2, \dots, i_k]$, the system divides the the leaf in n new leaves, labeled as $[i_1, i_2, \dots, i_k, i_{k+1}], \dots [i_1, i_2, \dots, i_k, i_{k+n}]$ and copies $(X \setminus \Phi) \cup \Phi_1$ to the node $[i_1, i_2, \dots, i_k, i_{k+1}]$, and copies $X \setminus \Phi \cup \Phi_2$ to the node $[i_1, i_2, \dots, i_k, i_{k+2}]$ (see Figure 1).

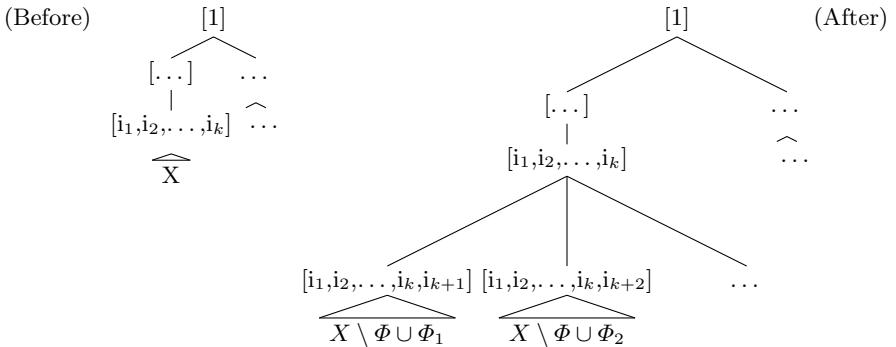


Fig. 1. Division of a leaf of the tree

We have translated the rules for RL_{OM} to clauses in Prolog. For example, for the union rule (\cup):

```
uni(Leaf):-  
    rel(Leaf,uni(R,S),X,Y),  
    new_rule_deduced([rel(Leaf,R,X,Y),rel(Leaf,S,X,Y)]),  
    \+rule_used(Leaf,uni,[rel(uni(R,S),X,Y)]),  
    write_rule('Union',[rel(Leaf,uni(R,S),X,Y)],  
              [rel(Leaf,R,X,Y),rel(Leaf,S,X,Y)]),  
    add_list_of_relations([rel(Leaf,R,X,Y),rel(Leaf,S,X,Y)]).
```

Any rule of RL_{OM} in Prolog checks the preconditions $x(R \cup S)y$ in which the rule is applicable. If the rule fulfils these conditions, we control if the relations deduced by the rule are new (`new_rule_deduced predicate`) and the rule has not previously applied (`rule_used predicate`), then we write the rule in the display and store the rule applied. Finally, we apply the rule, that normally adds some facts to the adequate leaf.

The (D2) rule divides the node labeled *Leaf* in two new leaves and copy all formulas of *Leaf* to the two new ones, by using the predicate `divideInLeaves`. The predicate `copyToLeaves` adds $zd_\alpha x$ to the first leaf and adds $zd_\alpha y$ to the second leaf.

```
d2(Leaf):-  
    rel(Leaf,equal,X,Y),  
    \+rule_used(Leaf,d2,[rel(equal,X,Y)]),  
    any_variable('d2 (equal)',Leaf,[rel(Leaf,equal,X,Y)],Z),!,  
    divideInLeaves(Leaf,2),  
    copyToLeaves(Leaf,1,[[rel(Leaf,dalpha,Z,X)]  
                            , [rel(Leaf,dalpha,Z,Y)]],[],ListNewLeaves),  
    remove_leaf_after_divide(Leaf),  
    write_and_rule('d2 (equal)',[rel(equal,X,Z)],  
                  [[rel(Leaf,dalpha,Z,X)],[rel(Leaf,dalpha,Z,Y)]]  
                  ,ListNewLeaves),!.
```

Now, we show the engine of the ATP. The main predicate in the inference engine is `run_engine` that examine the first leaf of the tree that the proof system needs to check and tries to apply the rules to the relations that contains this leaf. The engine tries first to apply the rules that no divide the leaves and then the rules that divide the leaves.

```
new_run_engine:-  
    leaves([FirstLeaf|Leaves]),  
    new_apply_rules_in_leaf(FirstLeaf),  
    new_run_engine,!.  
new_run_engine:-  
    write(' OK. There are no Leaves in the proof tree. '),  
    write(' VALID. '),!.  
new_apply_rules_in_leaf(FirstLeaf):-  
    new_one_rule_no_divide(FirstLeaf),!.  
new_apply_rules_in_leaf(FirstLeaf):-  
    new_one_rule_divide(FirstLeaf),!.  
new_one_rule_no_divide(FirstLeaf):-  
    uni(FirstLeaf)-> axiomatic_set;  
    notinter(FirstLeaf)-> axiomatic_set;  
    ...  
new_one_rule_divide(FirstLeaf):-  
    notuni(FirstLeaf)-> axiomatic_set;  
    d2(FirstLeaf)-> axiomatic_set;  
    ...
```

While the tree has opened leaves, `new_run_engine` is recursively called. If all leaves are closed in the proof system, then system informs to the user that the proof is finished and it is possible to trace (`used_rules predicate`) what rules have been used in the proof process. The engine of the ATP use the mechanism of pattern machine of Prolog to detect if exists, in a leaf of the tree, an axiomatic set, then deletes the corresponding leaf and informs to the user.

```
axiomatic_set:-  
    rel(NumLeaf,equal,X,X),  
    nl,  
    remove_leaf(NumLeaf,[rel(NumLeaf,equal,X,X)]),!.  
....
```

In this point, we introduce an important idea which improves the efficiency of our system. Some rules of the logic need to introduce *any object symbol*, that is, either a constant or any of the previously used variables. The ATP delays the substitution by any of the possible object symbols and introduces a *phantom variable*. The system replaces the phantom variable by any of the possible objects, only when obtains an axiomatic set with this substitution and then it closes the tree.

We emphasize that this mechanism avoids the process of selecting a possible variable and checking the validity of the formula in this leaf with this variable. In that case, it would be necessary to expand the leaf in a enormous sub-tree and, if the formula could not be proved, to return to the previous leaf by selecting another variable. The process would be repeated for all possible variables.

The *phantom variables* prune the search tree in a efficient way. The instantiation of the phantom variable is delayed until the ATP is able to obtain an axiomatic set. In this moment, the unification of the correct variable is done in the tree and some sub-trees are closed.

In the following example, we outline how the ATP works and emphasize the use of the phantom variables for detecting axiomatic sets in an automatic way.

Example 2. In this example, we execute the ATP to prove the axiom **d2** of the system OM. We represent it as follows:

```
rel([1],opp(comp(dalpha, comp(p, universal))),x,y).  
rel([1],opp(comp(dalpha, opp(comp(p, universal)))), x, y).
```

This example is satisfied by the ATP with the Prolog predicate:

```
?tad('axioms \ axiomd2.pl','logaxiomd2.txt').
```

The following report in `logaxiomd2.txt` file is returned:

```
----->Input file: axioms\axiomd2.pl  
leaves([[1]]).  
--->Opposite composition Rule  
[rel([1],opp(comp(dalpha,comp(p,universal))),x,y)]  
-----  
[rel([1],opp(dalpha),x,z),rel([1],opp(comp(p,universal)),z,y)]  
...  
...
```

```
[rel([1],comp(p,universal),t,y)]
-----
rel([1,1],p,t,t1) | rel([1,2],universal,t1,y)

Found axiomatic set. Leaf: [1,2]
- Axiomatic set: [rel([1,2],universal,t1,y)]
- Deleted relations in Leaf: [1,2]
...
[rel([1,1,1,1,1,2,2],opp(p),z,u)]
-----
rel([1,1,1,1,1,1,2,2,1],equal,z,t8) | rel([1,1,1,1,1,1,2,2,2],opp(p),t8,u)

Substitute in all relations variable phantom:t8 by t
Substitute in all relations variable phantom:t1 by u
Found axiomatic set. Leaf: [1,1,1,1,1,2,2,2]
- Axiomatic set: [rel([1,1,1,1,1,2,2,2],opp(p),t8,u),
                  rel([1,1,1,1,1,2,2,2],p,t,t1)]
- Deleted relations in Leaf: [1,1,1,1,1,2,2,2]
....
[rel([1,1,1,1,1,2,2,1],equal,z,t)]
-----
rel([1,1,1,1,1,1,2,2,1,1],dalp,ta9,z)|rel([1,1,1,1,1,1,2,2,1,2],dalp,ta9,t)

Substitute in all relations variable phantom:ta9 by x
Found axiomatic set. Leaf: [1,1,1,1,1,2,2,1,2]
- Axiomatic set: [rel([1,1,1,1,1,2,2,1,2],opp(dalp),x,t),
                  rel([1,1,1,1,1,2,2,1,2],dalp,ta9,t)]
- Deleted relations in Leaf: [1,1,1,1,1,2,2,1,2]
Found axiomatic set. Leaf: [1,1,1,1,1,2,2,1,1]
- Axiomatic set: [rel([1,1,1,1,1,2,2,1,1],opp(dalp),x,z),
                  rel([1,1,1,1,1,2,2,1,1],dalp,x,z)]
- Deleted relations in Leaf: [1,1,1,1,1,2,2,1,1]
OK. There are no Leaves in the proof tree. VALID.
```

Notice that the substitution of the phantom variable t_1 has been delayed until the appearance of variable t_8 , because in this moment, the leaf can be closed by replacing t_1 and t_8 by u and t , respectively. In this case, the last two leaves of the tree are closed with this unification process.

Finally, we remark that the system has some abduction mechanism. It is capable to give explanations about what rules have been used to prove a set of relations. We have the predicate `used_rules` that store knowledge about the reasoning process of the inference engine. We give below a trace in inverse order of the proof process:

```
used_rules([1,1,1,1,1,2,2,1],d2,[rel(equal,z,t)]).
used_rules([1,1,1,1,1,2,2],equality2,[rel(opp(p),z,u)]).
...
used_rules([1],notcomp,[rel(opp(comp(dalp,comp(p,universal))),x,y)]).
```

5 Conclusions and Future Work

In this paper, we have implemented an ATP for the relational proof system in the style of dual tableaux for the relational logic associated with the multimodal propositional logic for order of magnitude qualitative reasoning OM. This system can be used as a tool for verification of a variety of tasks in order of magnitude reasoning, such as the use of qualitative sum of some classes of numbers.

Nowadays, we are working in a more intelligent engine for the ATP and implementing a mechanism that selects what is the better rule by analyzing the relations and the variables in the tree. Also, we are improving the use of phantom variables to obtain an ATP more efficient.

The ATP works with depth-first search, at the moment. We are going to programme a more intelligent engine for the ATP, that combines the depth-first search with breadth-first search, depending on the analysis of the knowledge obtained from the formulas.

The goal for the future is to generalize this implementation for different logics (not only for order of magnitude reasoning). The idea is to develop an ATP more general that receives as input the description of the logic: constants, rules, constraints, etc. and renders the translation in a relational system. Moreover, the ATP will be allowed to prove the validity of any set of formulas of this logic.

Other future works are related to the study of decidability of this logic and, in the case of positive answer, to obtain decision procedures, by using some of the ideas presented in this paper. Last, but not least, it is planned to extend our ATP, in order to be used for model checking and verification of entailment.

References

1. Bennett, B., Cohn, A.G., Wolter, F., Zakharyaschev, M.: Multi-Dimensional Modal Logic as a Framework for Spatio-Temporal Reasoning. *Applied Intelligence* 17(3), 239–251 (2002)
2. Burrieza, A., Mora, A., Ojeda-Aciego, M., Orlowska, E.: Implementing a relational system for order-of-magnitude reasoning. Technical Report (2008)
3. Burrieza, A., Muñoz-Velasco, E., Ojeda-Aciego, M.: A Logic for Order of Magnitude Reasoning with Negligibility, Non-closeness and Distance. In: Borrajo, D., Castillo, L., Corchado, J.M. (eds.) CAEPIA 2007. LNCS (LNAI), vol. 4788, pp. 210–219. Springer, Heidelberg (2007)
4. Burrieza, A., Muñoz, E., Ojeda-Aciego, M.: Order of magnitude qualitative reasoning with bidirectional negligibility. In: Marín, R., Onaindía, E., Bugarín, A., Santos, J. (eds.) CAEPIA 2005. LNCS (LNAI), vol. 4177, pp. 370–378. Springer, Heidelberg (2006)
5. Burrieza, A., Ojeda-Aciego, M.: A multimodal logic approach to order of magnitude qualitative reasoning with comparability and negligibility relations. *Fundamenta Informaticae* 68, 21–46 (2005)
6. Burrieza, A., Ojeda-Aciego, M., Orlowska, E.: Relational approach to order-of-magnitude reasoning. In: de Swart, H., Orlowska, E., Schmidt, G., Roubens, M. (eds.) TARSKI 2006. LNCS (LNAI), vol. 4342, pp. 105–124. Springer, Heidelberg (2006)

7. Dague, P.: Symbolic reasoning with relative orders of magnitude. In: Proc. 13th Intl. Joint Conference on Artificial Intelligence, pp. 1509–1515. Morgan Kaufmann, San Francisco (1993)
8. Dallien, J., MacCaull, W.: RelDT: A relational dual tableaux automated theorem prover, <http://www.logic.stfx.ca/reldt/>
9. Formisano, A., Orlowska, E., Omodeo, E.: A PROLOG tool for relational translation of modal logics: A front-end for relational proof systems. In: Beckert, B. (ed.) TABLEAUX 2005 Position Papers and Tutorial Descriptions, Fachberichte Informatik No 12, Universitaet Koblenz-Landau, pp. 1–10 (2005), <http://www.di.univaq.it/TARSKI/transIt/>
10. Golińska-Pilarek, J., Muñoz-Velasco, E.: Relational approach for a logic for order of magnitude qualitative reasoning with negligibility, non-closeness and distance. Technical Report (2008)
11. Golińska-Pilarek, J., Orlowska, E.: Relational logics and their applications. In: de Swart, H., Orlowska, E., Schmidt, G., Roubens, M. (eds.) TARSKI 2006. LNCS (LNAI), vol. 4342, pp. 125–161. Springer, Heidelberg (2006)
12. MacCaull, W., Orlowska, E.: Correspondence results for relational proof systems with application to the Lambek calculus. *Studia Logica* 71, 279–304 (2002)
13. Mavrovouniotis, M.L., Stephanopoulos, G.: Reasoning with orders of magnitude and approximate relations. In: Proc. 6th National Conference on Artificial Intelligence. The AAAI Press/The MIT Press (1987)
14. Mavrovouniotis, M.L.: A belief framework for order-of-magnitude reasoning and other qualitative relations. *Artificial Intelligence in Engineering* 11(2), 121–134 (1997)
15. Orlowska, E.: Relational interpretation of modal logics. In: Andréka, H., Monk, D., Nemeti, I. (eds.) Algebraic Logic, Col. Math. Soc. J. Bolyai., vol. 54, pp. 443–471. North Holland, Amsterdam (1988)
16. Raiman, O.: Order of magnitude reasoning. *Artificial Intelligence* 51, 11–38 (1991)
17. Randell, D., Cui, Z., Cohn, A.: A spatial logic based on regions and connections. In: Proc. of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR 1992), pp. 165–176 (1992)
18. Rasiowa, H., Sikorski, R.: The Mathematics of Metamathematics. Polish Scientific Publishers, Warsaw (1963)
19. Sánchez, M., Prats, F., Piera, N.: Una formalización de relaciones de comparabilidad en modelos cualitativos. *Boletín de la AEPIA* (Bulletin of the Spanish Association for AI) 6, 15–22 (1996)
20. Travé-Massuyès, L., Ironi, L., Dague, P.: Mathematical Foundations of Qualitative Reasoning. *AI Magazine*, American Asociation for Artificial Intelligence, 91–106 (2003)
21. Wolter, F., Zakharyaschev, M.: Qualitative spatio-temporal representation and reasoning: a computational perspective. In: Lakemeyer, G., Nebel, B. (eds.) Exploring Artificial Intelligence in the New Millennium. Morgan Kaufmann, San Francisco (2002)

A Heuristic Data Reduction Approach for Associative Classification Rule Hiding

Juggapong Natwichai¹, Xingzhi Sun², and Xue Li³

¹ Computer Engineering Department, Faculty of Engineering
Chiang Mai University, Chiang Mai, Thailand

juggapong@eng.cmu.ac.th

² IBM Research Laboratory
Beijing, China

sunxingz@cn.ibm.com

³ School of Information Technology and Electrical Engineering
The University of Queensland, Brisbane, Australia
xueli@itee.uq.edu.au

Abstract. When data are to be shared between business partners, there could be some sensitive patterns which should not be disclosed to the other parties. On the other hand, the “quality” of the data must also be preserved. This creates an interesting question: how can we maintain the shared data that are guaranteed to have the quality, and the certain types of sensitive patterns be removed or “hidden”? In this paper, we address such the problem of sensitive classification rule hiding by using data reduction approach, i.e. removing the whole selected tuples in the given dataset. We focus on a specific type of classification rules, i.e. associative classification rules. In our context, a sensitive rule is hidden when its support falls below a minimal support threshold. Meanwhile, the impact on the data quality of the dataset is represented in term of a number of false-dropped rules, and a number of ghost rules. We present a few observations on the data quality with regard to the data reduction processes. From the observations, we can represent the impact by each reduction precisely without any re-applying the classification algorithm. Subsequently, we propose a heuristic algorithm to hide the sensitive rules based on the observations. Experimental results are presented to show the effectiveness and the efficiency of the proposed algorithm.

1 Introduction

Data mining can provide powerful tools for extracting useful patterns, or knowledge, from given data. However, the data may contain sensitive private information of individuals. This situation has raised privacy concerns to data mining research community. Moreover, as data sharing between cooperating organizations becomes a common business practice in order to utilize the collected data, the problem seems to be even escalated. To prevent the disclosure of the sensitive data, the techniques such as data transformation to conform k-anonymity standard [1] and their variants can be applied. Besides the privacy concern for

sensitive data, there exists another form of threat, i.e. the disclosure of sensitive patterns discoverable from the shared data.

In [2], the authors present a motivating example which sensitive patterns can damage the reputation of the individuals in the data. In this example, suppose that a dataset is shared publicly. A rule “**(PostCode = 5409) \wedge (Age = 18 to 25) \wedge (Gender = Male) \rightarrow HepBStatus = Yes**” is discovered from the dataset, suppose that the postcode 5409 referred to an indigenous community or the national parliament. This rule may be considered an offense to the population in the area and should be removed or “hidden” before the dataset is shared.

In data sharing scenario, in order to hide sensitive patterns, the given dataset needs to be modified so that the sensitive pattern becomes uninteresting against the pre-specified “interestingness” thresholds. On the other hand, the “quality” of the shared data must also be preserved as much as possible, i.e. data modification algorithms should also maintain the characteristics of the given dataset required by the sharing purpose. Apparently, failing to preserve data quality means that the data sharing is useless.

Typically, the existing data modification algorithms [3,4,5,6] apply data perturbation approach, i.e. changing some data values in a given dataset from an original value to another value. Although such the approach could hide sensitive patterns and possibly maintain data quality, it has the following drawbacks. First, some of the data values in the perturbed dataset are not “original” values. Further, there is no method to distinguish the real data and the modified data within a perturbed dataset. This drawback could reduce the creditability of modified datasets. Finally, the perturbation may modify some tuples and cause some uninteresting patterns to become interesting.

In this paper, we address the problem of sensitive pattern hiding by data reduction approach, i.e. removing the whole selected tuples from the dataset. Comparing with the data perturbation approach, all data values in a modified data set are original. So, this approach produces credible data sets in detail level. Also, if some uninteresting patterns become interesting by a data reduction, it is always the case that for these patterns, at least one of their interestingness measures have reached the threshold at the first place, but some tuples may block the patterns from being interesting against the other interestingness measure(s). This is different from the perturbation approach which this situation may be created artificially.

The pattern type addressed in this paper is associative classification rules [7]. For the hidden condition of sensitive rules, a sensitive rule is hidden successfully if its support is fallen below the pre-specified support threshold. Because the support of a rule is its statistical interestingness, so, a rule is worth consideration if its support is higher than the minimum threshold. Meanwhile, the impact on data quality is represented in term of a number of false-dropped rules, and a number of ghost rules which are well-known quality definitions. False-dropped rules are non-sensitive rules which their support fall below minimal support threshold, or their confidence fall below minimal confidence threshold by data modification unintentionally. Ghost rules are falsely generated by data

modification. To maintain data quality, data modification algorithms must keep the two numbers as low as possible.

As sensitive pattern hiding problem is proven to be NP-hard problem in [8], heuristic algorithms are usually proposed to address the problem. Typically, after a heuristic algorithm is applied to a dataset, we will obtain a modified dataset which is free from sensitive patterns. Subsequently, the quality of the dataset will be evaluated by re-applying the classification algorithm. In this paper, we present a few observations on the data quality with regard to the data reduction processes. From the observations, we can represent the impact on the data quality by each reduction precisely without any re-applying the classification algorithm. Subsequently, we propose a heuristic algorithm to hide the sensitive rules based on the observations. As we can avoid re-applying the classification algorithm, it means our algorithm can use the proposed heuristic with low computational cost. Moreover, unlike the other algorithms which have output as only modified datasets, our algorithm outputs both modified datasets and rules on the datasets. Therefore, we can also skip a final re-applying to evaluate the solution.

The organization of this paper is as follows. The problem is defined Section 2. Section 3 presents the observations on the data quality with regard to the data reduction processes. A proposed algorithm to hide sensitive rule based on the observations is presented in Section 4, and its experimental result is reported in Section 5. Finally, we conclude our work in Section 6.

2 Problem Definition

In this section, firstly, we introduce the basic notation required for our consideration: Dataset and Classification.

Definition 1 (Dataset). Let a dataset D be a collection of tuples defined on a schema \mathbf{A} , $D = \{d^1, d^2, \dots, d^n\}$. For each attribute $A_j \in \mathbf{A}$, its domain is denoted as $\text{dom}(A_j) \subseteq \mathcal{N}$, where \mathcal{N} is the set of natural number. For each $d^i \in D$, $d^i(\mathbf{A}) = (d^i(A_1), d^i(A_2), \dots, d^i(A_k))$, denoted as $(d_1^i, d_2^i, \dots, d_k^i)$. Note here that tuples in a table is not necessary to be unique.

Let C be a set of class labels, such that $C = \{c_1, c_2, \dots, c_o\}$, each $c_m \in C$ is a natural number. The class label of a tuple d^i is denoted as $d^i.\text{Class}$.

The label is just an identifier of a class. A class which is labelled as c_m defines a subset of tuples which is described by data assigned to the class. The classification problem is to establish a mapping from D to C .

Please note that we are defining the general dataset for the traditional associative classification problem. In the dataset, we allow duplication (i.e. two data entries are identical in terms of tuple and class label) and conflict (i.e. two data entries can have the same tuple information but with different class label).

Definition 2 (Classification). A literal p is a pair, consisting of an attribute A_j and a value v in $\text{dom}(A_j)$. A tuple d^i will **satisfy** the literal $p(A_j, v)$ iff $d_j^i = v$.

For all $l \in L$, $r_l : \bigwedge p \rightarrow c_m$, where p is the literal, and c_m is a class label. The left hand side (LHS) of the rule r_l is the conjunction of the literals, denoted as $r_l.LHS$. The right hand side (RHS) is a class label of the rule r_l , denoted as $r_l.RHS$.

A tuple d^i **satisfies** the classification rule r_l iff it satisfies all literals in $r_l.LHS$, and has a class label c_m as $r_l.RHS$.

A tuple d^i which satisfies the classification rule r_l is called **supporting tuple** of r_l . The **support** of the rule r_l , denoted as $Sup(r_l)$, is the ratio between the number of supporting tuples of r_l and the total number of tuples. The **confidence** of rule r_l , denoted as $Conf(r_l)$, is the ratio between $Sup(r_l)$ and the total number of tuples which satisfy all literals in LHS of r_l . Given a dataset D , a set of class labels C , a minimal support threshold $minsup$, and a minimal confidence threshold $minconf$, a set of classification rules $R = \{r_1, r_2, \dots, r_q\}$ can be derived.

Typically, a number of classification rules which satisfy minimal support and confidence values can be large [7]. The set of rules should be pruned by removing some “redundant” rules before being applied in the classification for the target dataset. So, in the context of sensitive rule hiding problem, we should deal with only unpruned rules. In this paper, rule hiding and data quality are addressed on the concept of “general rules” as follows.

Definition 3 (General Rule). Given a dataset D , a set of classification rules R satisfying minimal support $minsup$, and minimal confidence $minconf$. A classification rule $r_l \in R$ is a general rule if there does not exist a classification rule $r'_l \in R$ which $r_l.RHS = r'_l.RHS$ and $r_l.LHS \supset r'_l.LHS$.

Before we discuss further, we present here an example dataset. It will be used through this paper. Suppose we are dealing with the 3-attributes dataset, and two classes as shown in Table 1a).

With the minimal support and minimal confidence set at 2 and 90% respectively, we can derive a set of general rules from the example data set by an associative classification algorithm as shown in Table 1b). We can see that non-general rules are not listed, for example, a rule $(A_2 = 0) \wedge (A_3 = 0) \rightarrow 1$ which has support 2 and 100 % confidence is not listed because a rule $r_1, (A_3 = 0) \rightarrow 1$ is more general.

Table 1. An example dataset and its set of rules

a)

Tuple ID	A_1	A_2	A_3	C
1	1	0	1	0
2	1	1	1	0
3	1	0	1	0
4	0	1	1	1
5	0	0	0	1
6	0	1	1	1
7	1	1	0	1
8	1	0	0	1
9	0	1	0	1
10	1	1	0	1

b)

Rule No.	Content	Support	Confidence
1.	$(A_3 = 0) \rightarrow 1$	5	100%
2.	$(A_1 = 0) \rightarrow 1$	4	100%
3.	$(A_1 = 1) \wedge (A_3 = 1) \rightarrow 0$	3	100%
4.	$(A_2 = 0) \wedge (A_3 = 1) \rightarrow 0$	2	100%

Subsequently, we define the “hidden” condition for a sensitive rule as follows.

Definition 4 (Hidden Rule). *Given a dataset D with a set of class labels C , let R be the set of general classification rules from D satisfying a minimal support threshold minsup , and a minimal confidence threshold minconf . Let $R_s \subset R$ be a set of sensitive classification rules. A sensitive rule r_s is hidden if its support in D' , the modified dataset, less than minsup .*

The impact on data quality is represented in terms of a number of false-dropped rules and a number of ghost rules in the modified data set. They are defined as follows.

Definition 5 (False-dropped Rules). *A false-dropped rule is a non-sensitive general rule in $R - R_s$ in the original data set D which can not be derived from the modified data set D' by using minimal support minsup and minimal confidence minconf . The number of false-dropped rules, from hiding the sensitive rule r_s by removing of d^i , is denoted as $fd(s, i)$.*

Definition 6 (Ghost Rules). *A ghost rule is a rule which can not be derived from the original dataset D by using minimal support minsup and minimal confidence minconf , but can be derived from the modified data set D' . The number of ghost rules when the tuple d^i is removed in order to hide the rule r_s is denoted as $gh(s, i)$.*

Remember that in the associative classification problem, we only consider the most general interesting classification rules as the mining result. This will lead to some additional circumstances in which false-dropped rules and ghost rules can be generated during the hiding process. First, the false-dropped rules can also be caused by the confidence increase of the previously uninteresting rules. For example, suppose that r_0 is an uninteresting rule, and during the hiding process, it becomes interesting due to the increase of its confidence (that is, r_0 is a ghost rule). If r_0 is more general than some interesting non-sensitive rules, these less general rules should be removed from the result set and therefore, become the false-dropped rules. Similarly, if a non-sensitive rule r_1 becomes uninteresting due to the decrease of its confidence (i.e. r_1 is a false-dropped rule), some rules which are less general than r_1 but with the support and confidence above the thresholds could appear in the result set because they are now the most general interesting rules. According to our definition, these rules are ghost rules.

From the above definitions, we formalize sensitive associative classification rule hiding problem by data reduction approach as follows.

Problem 1. Given a data set D with set of class labels C , let R be the set of associative classification rules from D and for any rule $r \in R$, $\text{Sup}(r) > \text{minsup}$ and $\text{Conf}(r) > \text{minconf}$, where minsup and minconf are two given thresholds. In addition, let $R_s \subset R$ be a set of sensitive classification rules. The problem is to transform D into D' by removing some tuples from D such that 1) any rule $r_s \in R_s$ is invalid in D' in terms of the threshold minsup and 2) the impact, i.e. the summation of the number of false-dropped rules and the number of ghost rules, of removal is minimum.

Note here that the impact is defined as the summation for simplicity. It could be adjusted according to the application. For example, in medical domain, ghost rules could lead to the wrong treatment [9], so it should be weighted as the higher impact on data quality, then data modification algorithms will prefer to generate false-dropped rules.

3 Impact on Data Quality

In this section, we present our observations of the impact on data quality with regard to the data reduction processes. Subsequently, we propose a heuristic algorithm based on them in the next section.

In this paper, we propose to use a geometric model to help improving illustration of the impact. Note that the geometric model is only applied to facilitate our discussion. Essentially our proposal on hiding the classification rules is not necessary to be presented based the geometric model. However, applying the geometric model can help to explain some key concepts and observations better.

First, from our running example in Table 1, since we have three attributes, so we can represent the dataset in three dimensional geometric model as shown in Figure 1a). Each tuple is represented as a point. The number of tuples with the same attribute values, or duplicate tuples, is represented by the number label. For example at the coordinate $(1, 0, 1)$ which is the duplication of d^1 and d^3 , it has label “ $2 : d^1, d^3$ ”. For the rules, in our running example, we can represent rules by points, lines, or faces for the rules with three, two, and one literal respectively. From the rules in Table 1b), we can represent them as shown in Figure 1b).

3.1 False-Dropped Rules

We begin with the discussion of false-dropped rules. When we remove a supporting tuple, for the non-sensitive rules which are supported by the tuple, both their support and confidence are decreased. If the support or confidence value

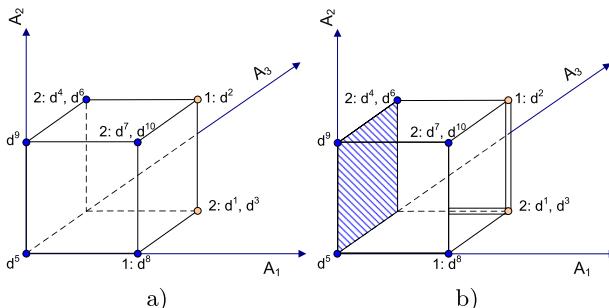


Fig. 1. The geometric model

of any non-sensitive rule is less than the threshold, the rule can not be derived in the modified data set and becomes a false-dropped rule.

Example 1. Suppose that a data owner wants to hide sensitive rule $r_3 : (A_1 = 1) \wedge (A_3 = 1) \rightarrow 0$ in the running example, which has supporting tuples $\{d^1, d^2, d^3\}$. If tuple d^1 (or d^3) is selected, we can see from the model in Figure 1b) that the selected tuple also supports non-sensitive rule $r_4(A_2 = 0) \wedge (A_3 = 1) \rightarrow 0$. Moreover, the support of r_4 is exactly equal to the minimal support threshold. After the tuple is removed, rule r_4 is lost. So, the number of false-dropped rules from removal of d_1 (or d^3) to hide rule r_3 , $fd(3, 1)$, is 1.

Observation 1. Let d^i be a tuple to be removed. In addition, Let R_FD be the set of false-dropped rules caused by the removal of d^i . For any rule $r_l \in R_FD$, it must satisfy the following conditions: 1) $r_l \in R - R_s$, 2) d^i is a supporting tuple of r_l , i.e. $d^i \in D_l$, and 3) $sup(r_l) = minsup$ or $(sup(r_l) - 1) < minconf * (sup(r_l.LHS) - 1)$. The number of false-dropped rules by removing d^i , $fd(s, i) = |R_FD|$.

3.2 Ghost Rules

For the impact in term of ghost rules, it can be considered opposite to the impact of false-dropped rules. In a dataset, there may exist some rules whose support is greater than $minsup$, but confidence below $minconf$. When a supporting tuple of a sensitive rule r_l is removed, it may increase confidence of this type of rules if the tuple satisfies the LHS of the rules, but the rules have different class label. If the increasing confidence of a rule can satisfy the minimal confidence threshold, the rule will become a ghost rule.

Example 2. Suppose that the data owner wants to hide a sensitive rule $r_1 : (A_3 = 0) \rightarrow 1$ in the running example, its supporting tuples which we can remove are $\{d^5, d^7, d^8, d^9, d^{10}\}$. If tuple d^8 is removed, we can see from the model in Figure 1b) that d^8 satisfies their all literals of rules $(A_2 = 0) \rightarrow 0$, $(A_1 = 1) \rightarrow 0$, and $(A_1 = 1) \wedge (A_2 = 0) \rightarrow 0$, but have different class. However, they are not derived in the first place because the confidence values of these rules are less than minimal confidence threshold. Considering the dataset, d^8 removal will cause the confidences of the rule $(A_1 = 1) \wedge (A_2 = 0) \rightarrow 0$ to increase and satisfy the minimal confidence threshold. After the removal, such the ghost rule $(A_1 = 1) \wedge (A_2 = 0) \rightarrow 0$ is generated. The number of ghost rules from removal of d_8 to hide the rule r_1 , $gh(1, 8)$, is 1.

Observation 2. To find all ghost rules, we need to maintain all rules whose support is greater than $minsup$ (regardless of their confidence). Let such the set of rules be denoted as R_C . Note that R_C is not the set of general rules, however, we can derive the general rules from R_c . Given a tuple to be removed d^i for hiding a sensitive rule r_s , let R_GH_C be the set of ghost rules caused by the removal of d^i . For any rule $r_l \in R_GH_C$, it must satisfy the following conditions: 1) $r_l \in R_c$, 2) d^i satisfies all the literals in $r_l.lhs$ and $r_l.RHS \neq r_s.RHS$, and 3)

$\text{sup}(r_l) \leq \text{minconf} * \text{sup}(r_l.\text{LHS})$ and $\text{sup}(r_l) > \text{minconf} * (\text{sup}(r_l.\text{LHS}) - 1)$. The number of ghost rules by removal d^i , $\text{gh}(s, i) = |R_{\text{GH}}|$, where R_{GH} is the set of general rules derived from R_{GH_C} .

4 Algorithm

In this section, we present a heuristic algorithm to solve the problem of associative classification rule hiding which applies the intuitive observations from the previous section. In order to select a tuple to be removed to hide a sensitive rule, the algorithm will select to remove the tuple which generates minimal impact potentially.

```

Input:
 $D$  (a dataset),  $\text{minsup}$  (a support threshold), and  $\text{minconf}$  (a confidence threshold)
 $R$ : the set of associative classification rules in  $D$  (satisfying  $\text{minsup}$  and  $\text{minconf}$ )
 $R_s$ : the set of sensitive classification rules,  $R_s \subset R$ 
Output:
 $D'$ : the output dataset, from which  $R_s$  can not be derived
 $R'$ : the set of associative classification rules in  $D'$ 
Method:
1 for each  $r_s \in R_s$  ordered by the numbers of the literals of the rules in  $R_s$  ascendingly do
2    $R = R - \{r_s\}$ .
3   Find  $D_s$ .
4   for  $(i = 0; i < \text{Sup}(r_s) - \text{minsup} + 1; i++)$  do
5     Initialize minimum impact.
6     for each  $d^i \in D_s$  do
7       Find  $R_{FD}$ .
8       Determine  $fd(s, i)$ .
9       Find  $R_{GH}$ .
10      Determine  $gh(s, i)$ .
11      Determine impact,  $Impact(s, i) = fd(s, i) + gh(s, i)$ .
12      if  $Impact(s, i) < MinImpact$ 
13        Mark  $d^i$  as the tuple to be removed,
14        keep  $R_{FD}$  and  $R_{GH}$ .
15      select to remove a tuple  $d^i$  with minimum  $Impact(s, i)$ .
16      if  $fd(s, i) \neq 0$ 
17        delete false-dropped rules from  $R$ .
18      if  $gh(s, i) \neq 0$ 
19        add ghost rules to  $R$ .
20      Update dataset  $D$ .
21 Output  $D' = D, R' = R$ .

```

Fig. 2. The proposed heuristic algorithm

Figure 2 shows the pseudo code of the proposed algorithm. For any sensitive rule r_s , we first find the set D_s of tuples that support r_s . The key step is to select $(\text{Sup}(r_s) - \text{minsup} + 1)$ tuples with minimal impact. To do this, for each iteration of removal, we evaluate the impact for every tuple in current D_s . That is, for each tuple, we compute the set of false-dropped rule R_{FD} and the set of ghost rule R_{GH} based on the observations given in Section 3. Once the tuple with minimal impact is determined, we remove the tuple from D_s , update the current interesting classification rules based on R_{FD} and R_{GH} , and update

the dataset. The worst case time complexity of this algorithm is $O(q_s \times q_c \times n^3)$ where n is the number of the tuples in D , q_s is the number of sensitive classification rules, and q_c is the number of rules whose support is greater than the $minsup$.

Typically, when a set of sensitive patterns is given, sensitive pattern-hiding algorithms will consider hiding the rules on a one-by-one basis. In this paper we apply a proven heuristic presented in [5] to generate a sequence of sensitive rules to be hidden, i.e. rank the sensitive rules by the number of their literals ascendingly. Additionally, we improve the efficiency of the supporting tuple retrieval by applying an inverted file index as implemented in [3].

5 Experimental Results

In this section, we evaluate the effectiveness and efficiency of the proposed algorithm. The experiments are conducted on an 3 GHz Intel Pentium 4 with 1024 megabytes main memory running Microsoft Window XP. We compare the proposed algorithm with an algorithm which generates optimal solutions (minimum impact) by exploring the whole search space to hide sensitive rules. Both the proposed algorithm and the optimal algorithm are implemented by using JDK 5.0 based on Weka Data Mining Software.

The experiment is performed on three real-life datasets from UCI repository, i.e. mushroom, credit screening, and voting datasets. All datasets are transformed into binary datasets. Tuples with missing values are removed. The features of the datasets used in experiments are summarized in Table 2. The rule summary under given parameter settings on minimal support and minimal confidence are also presented. Note that the supports listed in the table is the ratio of support values to the total number of tuples.

5.1 Effectiveness Evaluation

Firstly, we investigate the effectiveness of the proposed heuristic algorithm, i.e. the data quality of modified datasets by two factors: numbers of sensitive rules to be hidden ($|R_s|$), and support range of sensitive rules (the range of $sup(r_s)$). When we consider the effect of $|R_s|$, the range of $sup(r_s)$ will be fixed. In the same way, we fix $|R_s|$, when we consider the effect of $sup(r_s)$. In each experiment, for five times, we randomly select sensitive rules according to a specified setting of $|R_s|$ and $sup(r_s)$. Then, for each random selection, we apply both algorithms to hide the selected sensitive rules. Finally, we report the five-time-average impact

Table 2. Features of datasets

Dataset	#Tuples	#Attributes	$minsup$	$minconf$	#General Rules	#All Rules	Support Range	Average #Literals
Voting	232	15	0.4	0.5	14	71	0.40-0.51	1.55
Credit Screening	653	16	0.15	0.5	12	52	0.16-0.44	3.44
Mushroom	5644	22	0.1	0.4	9	66	0.1-0.33	4.67

Table 3. Impact on data quality

a) In term of $ R_s $							b) In term of $sup(r_s)$						
Dataset	$ R_s $	Heuristic Algorithm		Optimal Algorithm			$sup(r_s)$	Heuristic Algorithm		Optimal Algorithm			
		FD	GH	FD	GH			FD	GH	FD	GH		
Voting	1	1.2	0.0	1.0	0.0		0.40-0.41	0.2	0.0	0.2	0.0		
	2	1.6	0.0	1.4	0.0		0.42-0.43	1.6	0.0	1.4	0.0		
	3	2.0	0.0	2.0	0.0		0.44-0.45	1.8	0.0	1.6	0.0		
	4	1.2	0.0	1.2	0.0		0.46-0.47	3.0	0.0	2.6	0.0		
	5	0.4	0.0	0.4	0.0		0.48-0.50	3.2	0.0	2.8	0.0		
Credit Screening	1	0.4	1.4	0.4	1.0		0.20-0.24	0.2	0.2	0.2	0.2		
	2	0.6	1.4	0.6	1.2		0.25-0.29	1.0	1.0	0.8	0.8		
	3	1.0	2.0	0.8	2.0		0.30-0.34	1.2	1.2	1.0	1.2		
	4	1.2	1.0	1.0	1.0		0.35-0.39	1.6	1.6	1.2	1.4		
	5	0.0	1.0	0.0	1.0		0.40-0.45	2.0	2.2	1.8	1.6		
Mushroom	1	1.6	0.0	1.6	0.0		0.10-0.14	0.0	0.0	0.0	0.0		
	2	2.0	0.0	1.8	0.0		0.15-0.19	1.0	0.0	1.0	0.0		
	3	1.4	0.0	1.4	0.0		0.20-0.24	2.0	0.0	1.6	0.0		
	4	1.0	0.0	1.0	0.0		0.25-0.29	2.2	0.0	2.0	0.0		
	5	0.4	0.0	0.4	0.0		0.30-0.34	2.2	0.0	2.2	0.0		

on data quality, i.e., the average number of false-dropped rules and ghost rules for the given setting.

Table 3a) shows the effect of $|R_s|$ to the data quality. Numbers of false-dropped rules and Numbers of ghost rules are denoted here as “FD” and “GH” respectively. The fixed ranges of $sup(r_s)$ are 0.42-0.44, 0.18-0.25, and 0.18-0.27 for voting, credit screening, and mushroom datasets respectively.

From Table 3a) it can be seen that both algorithms perform very well. We can see that the optimal algorithm can perform only slightly better than our proposed heuristic algorithm, although it explores the whole search space. We also observe that when the number of sensitive rules is increased, the number of false-dropped rules is also increase for some period, then, it starts to drop. The reason is because the numbers of derivable general rules in different datasets are the certain constant numbers (14, 12, 9 for voting, credit screening, and mushroom respectively). Therefore, the more general rules to be hidden, the less number of non-sensitive rules will be false-dropped. We also observe that there is no ghost rule generated from voting and mushroom dataset. For voting dataset, the reason is because the high minimal support range is used (0.42-0.44), when we hide sensitive rules, many more tuples will be removed. This means the potential ghost rules are removed. While mushroom dataset is very sparse, so, it is hard to find a new ghost rule when data reduction approach is used.

In Table 3b), the effect of $sup(r_s)$ to the data quality is shown. The numbers $|R_s|$ are fixed at 2 for all datasets. From the result, we can see that the impact on data quality increases when we increase $sup(r_s)$ in both algorithms, though, it is still considered relatively low. Since we fix the number of sensitive rules at 2, from the results when we hide the rules with highest ranges of support, the heuristic still can preserve the majority of the non-sensitive general rules, i.e. the average numbers of preserved non-sensitive rules are 10.0 out of 12, 8.8 out of 10, and 5.5 out of 7 rules for voting, credit screening, and mushroom data sets respectively.

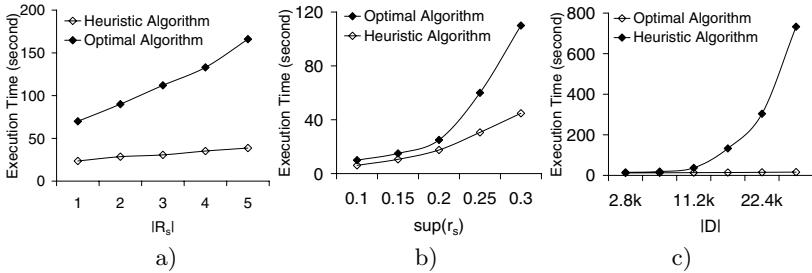


Fig. 3. Efficiency Evaluation

5.2 Efficiency Evaluation

In this section, the efficiency of the proposed algorithm, i.e. the execution time is considered. We investigate the efficiency in term of $|R_s|$, $\text{sup}(r_s)$, and the size of input datasets ($|D|$). In this experiment, only the results on mushroom dataset is presented due to the space limitations. When the effect of $|R_s|$ is considered, we fix the range of $\text{sup}(r_s)$ at 0.20-0.25. And, we fix $|R_s|$ at 2, when we consider the effect of $\text{sup}(r_s)$. To evaluate the efficiency in term of $|D|$, we fix $|R_s|$ and $\text{sup}(r_s)$ at 2 and 0.20-0.25 respectively.

From Figure 3a) and 3b), we can see that our proposed heuristic algorithm uses much less execution time compared with the optimal algorithm. This is because the optimal algorithm must explore the whole search space to find the solutions. From Figure 3b), the execution time of the optimal algorithm increases almost exponentially when the support of sensitive rules increases. If we consider the efficiency of both algorithms along with the effectiveness, we can see that our proposed algorithm can perform almost as effective as the optimal algorithm, but uses much less time. In Figure 3c), we can see that the execution time of our proposed algorithm is very small comparatively when the number of tuples in the dataset increases.

6 Conclusions

To summarize, in this paper, we address the problem of hiding sensitive associative classification rules by data reduction approach. The focus of the problem is on minimizing the impact on the data quality, which is modelled in terms of the number of false-dropped rules and ghost rules. Based on our observations, we can determine the impact on data quality precisely without any re-applying the classification algorithm. Accordingly, an algorithm is proposed to hide the sensitive rules. The experimental results show that the proposed algorithm is effective, i.e. it can preserve data quality very well compared with the optimal algorithm. Additionally, the proposed algorithm is also efficient. In our future work, we will target on addressing the problem where the given datasets have the attributes with richer domains, e.g. categorical or continuous attributes.

References

1. Sweeney, L.: k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* 10, 557–570 (2002)
2. Fule, P., Roddick, J.F.: Detecting privacy and ethical sensitivity in data mining results. In: ACSC 2004: Proceedings of the 27th Australasian Conference on Computer Science, pp. 159–166. Australian Computer Society, Inc. (2004)
3. Oliveira, S.R.M., Zaïane, O.R.: Privacy preserving frequent itemset mining. In: Proceedings of the IEEE international conference on Privacy, security and data mining, pp. 43–54. Australian Computer Society, Inc. (2002)
4. Verykios, V.S., Elmagarmid, A.K., Bertino, E., Saygin, Y., Dasseni, E.: Association rule hiding. *IEEE Transactions on Data and Knowledge Engineering* 16, 434–447 (2004)
5. HajYasien, A., Estivill-Castro, V.: Two new techniques for hiding sensitive itemsets and their empirical evaluation. In: Proceedings of 8th International Conference on Data Warehousing and Knowledge Discovery, pp. 302–311. Springer, Heidelberg (2006)
6. Moustakides, G.V., Verykios, V.S.: A max-min approach for hiding frequent itemsets. In: Workshops Proceedings of the 6th IEEE ICDM International Conference on Data Mining, pp. 502–506. IEEE Computer Society Press, Los Alamitos (2006)
7. Li, W., Han, J., Pei, J.: Cmar: Accurate and efficient classification based on multiple class-association rules. In: Proceedings of the 2001 IEEE ICDM International Conference on Data Mining, Washington, DC, USA, pp. 369–376. IEEE Computer Society Press, Los Alamitos (2001)
8. Atallah, M., Elmagarmid, A., Ibrahim, M., Bertino, E., Verykios, V.: Disclosure limitation of sensitive rules. In: KDEX 1999: Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange, Washington, DC, USA, pp. 45–52. IEEE Computer Society Press, Los Alamitos (1999)
9. Wu, Y.H., Chiang, C.M., Chen, A.L.P.: Hiding sensitive association rules with limited side effects. *IEEE Transactions on Knowledge and Data Engineering* 19, 29–42 (2007)
10. Natwichai, J., Orlowska, M.E., Sun, X.: Hiding sensitive associative classification rule by data reduction. In: Alhajj, R., Gao, H., Li, X., Li, J., Zaïane, O.R. (eds.) ADMA 2007. LNCS (LNAI), vol. 4632, pp. 310–322. Springer, Heidelberg (2007)

Evolutionary Computation Using Interaction among Genetic Evolution, Individual Learning and Social Learning

Takashi Hashimoto and Katsuhide Warashina

School of Knowledge Sciene,
Japan Advanced Institute of Science and Technology (JAIST)
1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan
{hash,warashina}@jaist.ac.jp
<http://www.jaist.ac.jp/~hash/index-e.html>

Abstract. This paper studies the characteristics of interaction among genetic evolution, individual learning and social learning using an evolutionary computation system with NK fitness landscape, both under static and dynamic environments. We show conditions for effective social learning: at least 1.5 times lighter cost of social learning than that of individual learning, beneficial teaching action, low epistasis and dynamic environment.

Keywords: Evolutionary computation, Genetic evolution, Individual learning, Social learning, NK fitness landscape.

1 Introduction

Biologically inspired computation algorithms, such as neural networks mimicking the brains and genetic algorithm simply implementing genetic evolution, are often utilized in many adaptive and intelligent systems, optimization and system designing. Recently, adaptive algorithms using interaction between evolution and learning have been studied [1,2,3,4]. In this paper, we also study such adaptive algorithm, especially we pay attention to the interaction among genetic evolution, individual and social learnings.

Learning is classified into individual and social. The former is change of individual characters through individual experiences, such as enhancement of muscles through exercises and gain of knowledge and skills by trial and error. The latter is transmission of knowledge and skills through direct and indirect interactions between individuals. The social learning is mediated by imitation or teaching. While the individual learning is often seen in many organisms, the social learning is found in only some animals with sociality.

The representative of such social animals is some primates including humans. We claims that the ability of social learning is one of the key features enabling the humans to adapt to various environments. Thanks to this ability, the humans can discovers new knowledge accumulatively and utilize the knowledge of predecessors [5]. Such knowledge accumulated forms “culture”. The ability of social

learning works effectively when the ability of individual learning is adequately combined with it. The both abilities had evolved through genetic evolutionary processes. That is, the humans had acquired the characters realizing fruitful cultures and the culture itself is thought to have been evolved through interaction among genetic evolution, individual learning and social learning. We may be able to utilize such adaptive strategy for intelligent systems and optimization.

In this paper, we study the characteristic of evolutionary algorithm in which genetic evolution, individual learning and social learning interact with each other. Especially, we focus on conditions that enable effective social learning. Social learning is useful, as we have said, but not ubiquitous in biological species. This may be because that obtaining the ability of social learning is difficult. This fact lead us a prediction that the condition realizing the social learning is stern.

We adopt NK fitness landscape [6] as a model of environment for individuals to fit. The NK landscape models originally fitness function taking the interaction among genes, called epistasis, into consideration. Many combinatorial optimization problems can be reduced to the NK landscape. Actually, the NP-completeness had been proven [7]. This model has also been used as important test beds for search and optimisation techniques, especially, evolutionary computation algorithms. We investigate the characteristic of the present algorithm under static and dynamic environments, in the latter, the NK fitness landscape changes with generations.

This paper is structured as follows. We introduce the model to incorporate genetic evolution, individual and social learnings in section 2. The simulation results in static and dynamic NK landscapes are described in section 3. We discuss the results in section 4 from the viewpoint of the difficulty of social learning. The paper is concluded by section 5 to deliver conditions favorable to the social learning.

2 Model

We model a population of agents which are engaged in genetic evolution, individual learning and social learning on a NK fitness landscape. The structure of the model is schematically shown in Fig. 1. One generation consists of three phases, the individual learning, the social learning and the genetic evolution (reproduction), in turn.

2.1 Structure of Agent

Each agent has three types of genetic elements: a genotype G which is a bit string with length N , the maximum time of individual learning operations, $IL_{MAX} = 0 \sim L_{MAX}$, and the social learning factor, $SLFactor = \{t, s, i\}$. The genotype determines the agent's innate fitness, denoted by F_{gt} , on the predefined NK fitness landscape. This string has a circular structure with a head in order for all genes to have the same number of neighbors. The capacity of learning operations is limited by L_{MAX} , given as a common parameter to all agents. The capacity is apportioned to the individual and social learning operations, IL_{MAX} and

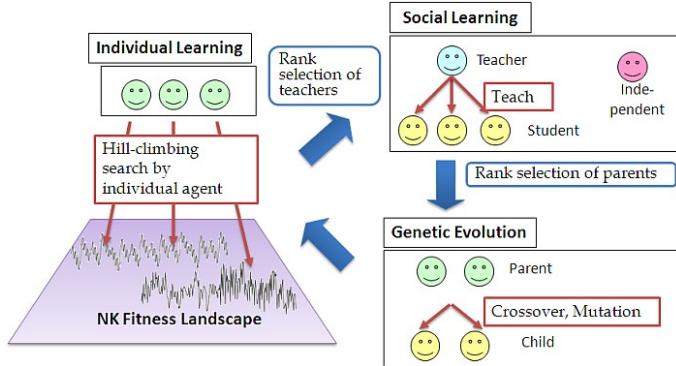


Fig. 1. The structure of the model, consisting of three phases corresponding to the three adaptive algorithms

SL_{MAX} ($L_{MAX} = IL_{MAX} + SL_{MAX}$). Each agent is doomed to be a teacher, a student or none of each genetically at the social learning phase. This role is represented by $SLFactor = \{t, s, i\}$, respectively.

Each agent has the other characters that change through learning: a phenotype P , which is the same as the genotype G at the moment of birth, counters for the individual learning, social learning and teaching operations, IL , SL and TL , respectively. The SL and TL are, respectively, used only by student agents having the student factor, $SLFactor = s$, and by teacher agents having the teacher factor, $SLFactor = t$.

The initial population is generated by the following procedure.

1. Generate agents having the same genotype which is randomly determined. The number of agents is Num .
2. Flip the genotype of each agent with a provability $1/N$ per each bit.
3. Determine IL_{MAX} between 0 and L_{MAX} and $SLFactor$ from $\{t, s, i\}$ using uniform random numbers.

2.2 Individual Learning

The individual learning of each agent proceeds as the following process:

1. Copy genotype G of the agent to a phenotype bit string P and set the individual learning counter IL to 0.
2. If any one bit flip of P does not increase the NK fitness of P , $F_{NK}(P)$, go to the 4th procedure; otherwise, make a bit string P' in which one random bit of P is flipped.
3. When $IL < IL_{MAX}$ and $F_{NK}(P') > F_{NK}(P)$, copy P' to P , increment IL and go to the 1st procedure; otherwise go to the 4th procedure.
4. Stop the individual learning phase of the agent and set the fitness after individual learning to $F_{indi} = F_{NK}(P)$.

2.3 Social Learning

In the social learning phase, the teacher agents transmit their phenotypes, that is, the learning results, to the student agents. At first, all teachers are ranked according to their fitness after individual learning, $F_{indi}(P^t)$, and their teaching counters TL are set to 0. The following two procedures are repeated for all the student agents.

1. Each student agent selects one teacher agent using the rank selection, that is, in terms of the probability proportional to the teacher ranking. If the teacher has higher fitness after individual learning than the student, $F_{indi}(P^t) > F_{indi}(P^s)$, then the teacher is adopted, otherwise the student does not learn socially, $F_{social} = F_{indi}(P^s)$.
2. Set the social learning counter SL of the student agent to 0. The student compares each bit of its phenotype P^s with the teacher's P^t . If the bit has different value, the student copies the teacher's bit and increments its social learning counter, SL . At the same time, the teacher increments its teaching counter, TL . When $SL = SL_{MAX}$ or $P^s = P^t$ during this copy process, the student stops copying and sets its fitness after social learning to $F_{social} = F_{NK}(P^s)$. Note that the teacher's phenotype may be partially copied to the student's due to the limitation of SL_{MAX} .

2.4 Fitness and NK Landscape

By means of the learned results, the lifetime fitness of the agent is calculated by

$$F_{lt} = F_b - C_{lt} , \quad (1)$$

$$F_b = \begin{cases} F_{indi} & \text{for } SLFactor = t \text{ or } i , \\ F_{social} & \text{for } SLFactor = s , \end{cases} \quad (2)$$

$$C_{lt} = C_{indi} \cdot IL + C_{student} \cdot SL + C_{teacher} \cdot TL , \quad (3)$$

where C_{indi} , $C_{student}$ and $C_{teacher}$ are the costs of the individual learning, social learning and teaching, respectively, given as parameters common to all agents.

The environment is modeled by Kauffman's NK fitness landscape [6]. An NK fitness landscape is specified by the length of genotype, N , and the strength of epistatic interactions among genes, K . The parameter K controls the ruggedness of the fitness landscape. Larger K brings the more number of local optima.

A landscape is defined by the N number of tables with 2^{K+1} uniform random numbers between 0.0 and 1.0. An example table is shown in Fig. 2. The i -th table determines the fitness of the i -th gene, $f_{NK}(i)$, by making correspondence between the $K+1$ bits patterns to the random numbers. The NK fitness of a genotype G is the average of the fitness of all genes, $F_{NK}(G) = (1/N) \sum_{i=1}^N f_{NK}(i)$. The same method and the same tables are used to calculate the fitness of phenotype.

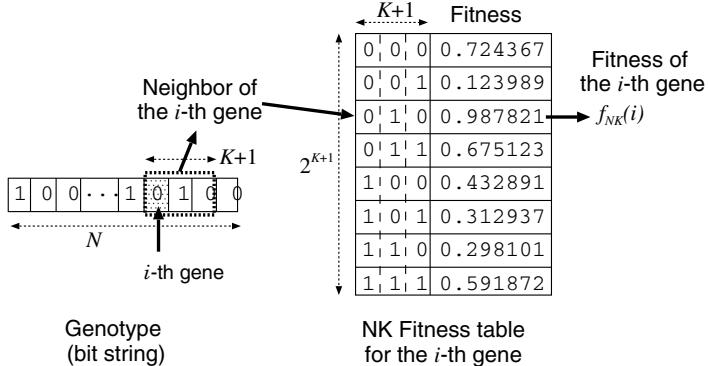


Fig. 2. An example of the NK fitness table for $K = 2$

2.5 Reproduction

The next generation consisting the same number of agents is produced through crossover and mutation.

At first, two parental agents are selected using rank selection according to the lifetime fitness, F_{lt} . Two genotypes from those of the parental agents are made with one-point crossover. Note that we use the genotypes of the parental agents, not their phenotypes, to prevent the inheritance of acquired characters. One of the new genotypes is randomly adopted as the genotype of an offspring agent. This agent inherits the maximum time of individual learning, IL_{MAX} , and the social learning factor, $SLFactor$, from one of the parental agents randomly determined.

Mutation of the genotypic structure consists of bit flips of the genotype, increment/decrement of IL_{MAX} , and change of $SLFactor$ with the mutation rate μ . If the result of mutation on IL_{MAX} exceeds the maximum value, L_{MAX} , or the minimum, 0, the increment/decrement operation is canceled. The result of mutation on $SLFactor$ may coincide with that before mutation, since one value from $\{t, s, i\}$ is adopted with equal probabilities.

3 Simulation Results

We conducted computational experiments under static and dynamic environments. In static environment, the NK fitness landscape is fixed at initially defined. In dynamic environment, the landscape changes with generations. We investigated from the viewpoint that under which conditions the social learning is effectively used or is superior to the individual learning.

The fixed parameters used in the experiments are, the number of agents, $Num = 100$, the length of genotype/phenotype bit strings, $N = 20$, the total capacity of learning operations, $L_{MAX} = 5$. All graphs shows the average data of 10 runs, unless specially indicated.

3.1 Static Environment

We show the dynamics of various fitness achievement rates averaged over all agents in Fig. 3. The achievement rate is the ratio of the fitness value to the optimal value in the landscape, indicated by bars on F 's. The parameter setting is the following: the individual learning cost $C_{indi} = 0.01$, the social learning cost, $C_{student} = 0.001$, the teaching cost, $C_{teacher} = -0.001$, the mutation rate, $\mu = 0.02$, the epistasis, $K = 2$. We use a moderate value of the mutation rate smaller than the error threshold ($\mu = 1/N = 0.05$), since some information should be passed over generations to estimate the effect of three evolutionary algorithms. Until around the 20th generation, the individual learning has much larger effectiveness than the social learning. After this generation, fitness raised by the social learning is larger than by the individual learning.

Figure 4 represents the dynamics of the average learning operations. While the individual learning is used at the initial stage, it comes to be unused. The acquired results through the individual learning seems to be genetically assimilated, since the individual learning is costly when $C_{indi} = 0.01$. Actually, as seen in Fig. 3, the innate fitness catches up with that after individual learning until the 55th generation. In contrast, the social learning operations less decreases relatively than the individual learning, as the social learning cost is ten times smaller than the individual learning cost. The value of IL and SL at the stable point depends on the parameter settings as shown in the following paragraphs.

The individual and social learning operations vary with the individual and social learning costs as shown in Fig. 5. This graph uses the average values of IL and SL at the 20th generations. Rightfully, the social learning is used than the individual learning at the region of larger individual learning cost and smaller social learning cost. The cross section of the IL and SL planes forms roughly a straight line, $C_{indi} \approx 1.5C_{social} + 0.0055$. This implies that both types of learning

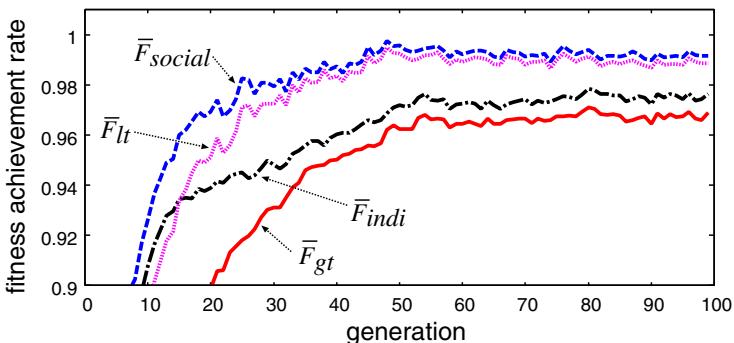


Fig. 3. The transition of fitness achievement rate averaged over all agents with generations in a static environment. The solid line is the achievement rate of the innate fitness \bar{F}_{gt} , the chain line is that after individual learning, \bar{F}_{indi} , the dashed line is that after social learning, \bar{F}_{social} , and the broken line is of the lifetime fitness achievement rate, \bar{F}_{lt} .

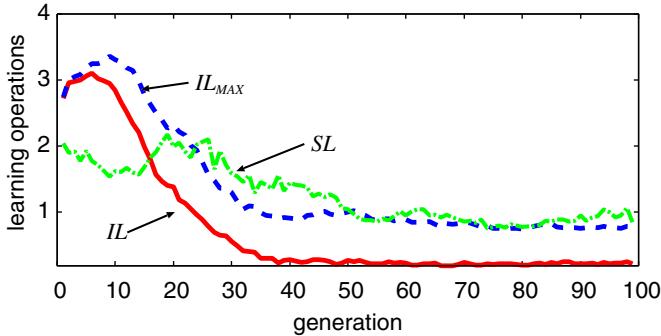


Fig. 4. The transition of the times of learning operations with generations in a static environment. The solid line is the times of individual learning, IL , and the dashed line is the maximum times of individual learning, IL_{MAX} . These are averaged over all agents. The chain line is the times of social learning, SL , averaged over all the student agents at each generation.

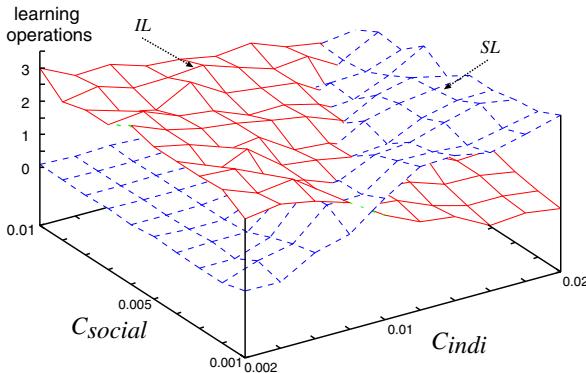


Fig. 5. The times of individual (IL) and social (SL) learning operations v.s. the individual (C_{indi}) and social (C_{social}) learning costs. The plane with solid and dashed lines represent IL and SL , respectively.

are used comparably when the cost of individual learning exceeds 1.5 times than that of social learning under the present parameter setting. We confirmed that if the individual learning cost is larger than 0.02, IL_{MAX} comes to nearly 0. This means that the individual learning is avoided under such large cost.

We investigated how the other important parameters, epistasis K and mutation rate μ , affect the learning operations. The times of learning operations changes with the both parameters as shown in Fig. 6. The individual and social learning costs are $C_{indi} = 0.01$, and $C_{social} = 0.001$. Larger K and μ , as overall effect, increase IL and decrease SL . The IL plane is roughly symmetrical with respect to the diagonal line of $K - \mu$ space. This implies that the epistasis and the mutation affect similarly the individual learning. As for the social learning,

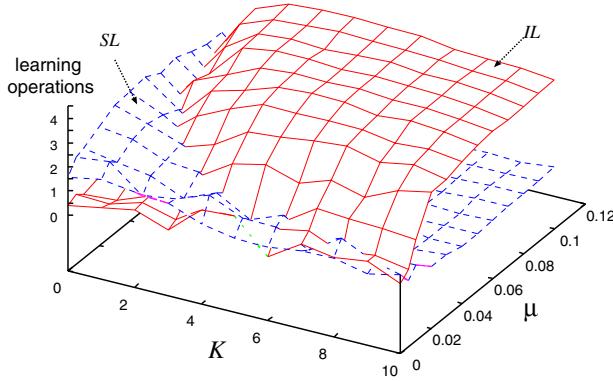


Fig. 6. The times of individual (*IL*) and social (*SL*) learning operations v.s. the epistasis (*K*) and the mutation rate (μ). The plane with solid and dashed lines represent *IL* and *SL*, respectively.

their effects are different. The change of *SL* with μ is smaller than that with *K*. *SL* takes the highest at $0.02 \lesssim \mu \lesssim 0.04$ in small *K* region.

The *IL* and *SL* planes are nearly flat at $IL = 4 \sim 4.4$ and at $SL = 0$, respectively, in the region $K \gtrsim 4$ and $\mu \gtrsim 0.05$ (coincide with the error threshold). In such rugged (complex) fitness landscape and unstable genetic circumstance, the agents use most of their learning capacity for the individual learning and the social learning does not work. Actually, we confirmed that, in such region, the difference between the innate fitness and the fitness after individual learning, $F_{indi} - F_{gt}$, is larger than in the region of smaller *K* and μ , and the fitness after social learning F_{social} virtually the same as F_{indi} . These two planes cross at the small *K* and μ . The cross section is approximately described by $\mu \cdot K \approx 0.04$.

In the above results, we used a negative teaching cost ($C_{teacher} = -0.001$). Namely, teaching behavior is not costly but beneficial, which is favorable for social learning. When the teaching cost is set at positive value, the social learning is very unstable (Fig. 7). The teacher only momentary lives, since selective pressure affects to exclude the teacher factors. The social learning operations sharply rises and falls stochastically.

3.2 Dynamic Environment

While organisms adapt genetically to stable environments, learnable organisms can adapt to changing environments. For changes with an intermediate time scale, cultural evolution through social learning may work well. In this section, we study how our hybrid evolutionary algorithm works in dynamic environments, since good adaptive algorithm for dynamic environment has not been invented so much. We are interested in the division of roles corresponding to time scales.

The way to change the environment is to remake one NK fitness table corresponding to one bit randomly selected. This models environmental change that affect the fitness of one gene. The fittest gene varies by this change, therefore the

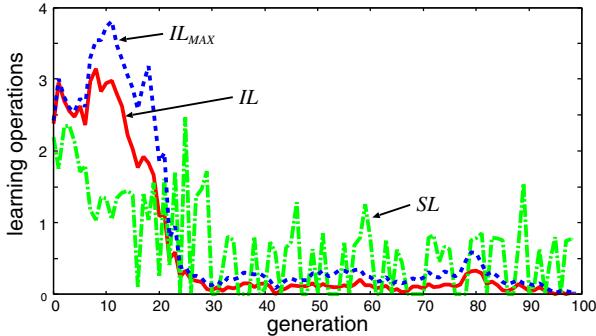


Fig. 7. The transition of the times of learning operations with generations in a static environment. The teaching cost is $C_{teacher} = 0.001$. The solid line is the times of individual learning, IL , and the dashed line is the maximum times of individual learning, IL_{MAX} . These are averaged over all agents. The chain line is the times of social learning, SL , averaged over all the student agents at each generation. This graph show the result of one typical run.

surrounding genes also indirectly affected, if $K > 0$. In our experiments, the environment changes every 5 generations. The parameter values are, $C_{indi} = 0.01$, $C_{social} = 0.001$ and $C_{teacher} = -0.001$ and $K = 2$, which are the same as the case in the static environment shown in Fig. 3 and 4.

In contrast to the static environment (Fig. 3), the innate fitness F_{gt} stays, as shown in Fig. 8, at lower value and does not increase substantially after the 25th generation. The rapid environmental changes brings this result, since the rapid change make genetic assimilation of learning result impossible. On the other hand, the social learning promotes the average fitness relatively than in the case of static environment.

Figure 9 compares the average times of individual and social learning operations in the static and dynamic environments. The social learning remains

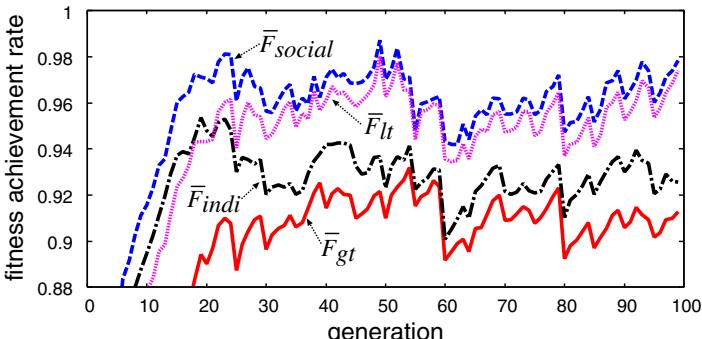


Fig. 8. The transition of the times of learning operations with generations both in static and dynamic environments. The legends of lines are the same as Fig.3.

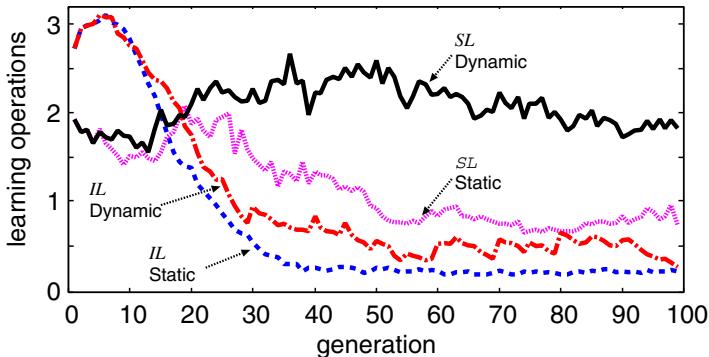


Fig. 9. The transition of the times of learning operations with generations both in static and dynamic environments. The dashed and chain lines are the times of individual learning, IL , for the static and dynamic environments, respectively. The broken and solid lines are the times of social learning, SL , for the static and dynamic environments, respectively.

until later generations at around 2.0 operations per each student agent when the environment is dynamic, while it decreases in the static environment with generations. The individual learning decays in the both environmental conditions.

4 Discussion

In our model, although one operation in both the individual and social learnings are one bit flip, the individual learning operation outnumbers that of social learning under the same cost level. In order to make the social learning superior, there must be cost difference of 1.5 times at least and the teaching must not be cost but benefit.

These severe conditions are brought by the several constraints for social learning in the present model. The individual learning always precedes the social learning. Unless the individual learning rises the fitness of some individuals, the only way for the social learning to improve the fitness of population is to propagate the innate superiority. Low diversity in the population prevents the social learning, since it is difficult for students to find good teacher. Further, the students cannot always copy the whole phenotypes of the teachers, since the times of social learning operation is limited by $SL_{MAX} (= L_{MAX} - IL_{MAX})$. In epistatic landscapes, incomplete copy of teacher phenotype often degrades the students' fitness, no matter how high the teacher's fitness is. Namely, the diversity in the population is indispensable but large diversity becomes harmful for the social learning to be effective. Such difficulties of social learning is not limited to our present model but essential feature of social leaning.

The social learning operations is the highest at the range of mutation rate $0.02 \lesssim \mu \lesssim 0.04$ in the low epistasis region. As we mentioned, on the one hand, the mutation rate must not be too small in order to supply diversity for effectual

social learning. On the other hand, too high mutation rate makes the genetic assimilation impossible. Actually, in the region of high mutation rate, the innate fitness, F_{gt} , hardly grows with generations, despite that individual learning rises the fitness, F_{indi} . Therefore, the maximum times of individual learning IL_{MAX} stays nearly at L_{MAX} . Namely, the learning capacity is devoted mostly to the individual learning, and the agents cannot reserve the capacity for social learning, even though the teacher and student agents exist.

We show that the learning capacity is used only for the individual learning also in strong epistasis $K \gtrsim 4$. Mayley suggests that individual learning does not work well when epistasis is too strong [2]. Based on Mayley's suggestion, how the individual learning works under strong epistasis in our model should be studied in more detail.

We indicate that in a dynamic environment the social learning is used continually than the individual learning and can contribute to rise the fitness, while the times and the effectiveness of individual learning are the same as in the case of static environment. However, the experiments and analysis of the dynamic environment are considerably insufficient, although the observation shown in section 3.2 is typical in that setting. We need intensive investigation about the phenomena concerning the way of environmental change, the degree of epistasis and costs.

5 Conclusion

We study a new type of evolutionary computation in which three adaptive algorithms, genetic evolution, individual learning and social learning, interact with each other. In this model, the three adaptive algorithms interact as follows. A population of individuals search higher fitness in a rugged landscape as hill-climbing using the individual learning. Then, the results of the learning are transmitted to the population from teachers to students using the social learning. Finally, the results of individual and social learnings are genetically assimilated due to the selective pressure posed by learning costs.

We investigated the conditions which favor the social learning. The conditions are qualitatively as follows: The individual learning cost is larger than the social learning cost. Teaching is beneficial for teachers. Mutation rate is low. Epistasis is low (The fitness landscape is not so complex). Environmental change occurs. In the present model, the individual learning cost should be at least 1.5 times than the social one; the mutation rate should be less than 0.04 per each gene; more than 3 genes should not interact.

As we discussed, the social learning in the present model has many constraints. The social learning is really so constrained that it is hard to establish biologically. Therefore, it is difficult to study the essential interaction of social learning with genetic evolution and individual learning. One of the most missing points concerning the social learning in our model is generation overlapping which is important to realize accumulative knowledge creation and transmission, We show only phenomenological findings in this paper. Although some conditions are

reasonable and some are discussed, we should pursue the understandable mechanism and causal relationship between the model and the conditions in order to understand the interaction among the three adaptive algorithms and to utilize their interaction. In order to discuss our algorithm from the view point of computational complexity, it would be interesting to analyze the time of evaluations required to verify if some changes on genotypes and phenotypes improve the fitness of agents or not.

Acknowledgement

The authors thank Hajime Jimmy Yamauchi for his important discussions. This work is supported by Grant-in-Aid for Scientific Research (No. 17680021) of Japan Society for the Promotion of Science (JSPS). We are grateful to reviewers for their valuable comments for improving our manuscript.

References

1. Hinton, G.E., Nowlan, S.J.: How learning can guide evolution. *Complex Systems* 1, 495–502 (1987)
2. Mayley, G.: Landscapes, learning costs and genetic assimilation. *Evolutionary Computation* 4(3), 213–234 (1996)
3. Best, M.L.: How culture can guide evolution: an inquiry into gene/meme enhancement and opposition. *Adaptive Behavior* 7(3-4), 289–306 (1999)
4. Arita, T., Suzuki, R.: Interactions between Learning and Evolution – Outstanding Strategy generated by the Baldwin Effect. In: *Proceedings of Artificial life VII*, pp. 196–205 (2000)
5. Tomasello, M.: *Cultural Origin of Human Cognition*. Harvard University Press (1999)
6. Kauffman, S.: Adaptation on rugged fitness landscapes. In: Stein, D. (ed.) *Lectures in the Sciences of Complexity*, pp. 527–618. Addison-Wesley, Reading (1989)
7. Wright, A.H., Thompson, R.K., Zhang, J.: The computational complexity of N-K fitness functions. *IEEE Transactions on Evolutionary Computation* 4(4), 373–379 (2000)

Behavior Learning Based on a Policy Gradient Method: Separation of Environmental Dynamics and State Values in Policies

Seiji Ishihara¹ and Harukazu Igarashi²

¹ Kinki University, 1 Takaya-umenobe, Higashi-hiroshima, Hiroshima, 739–2116, Japan
ishihara@hiro.kindai.ac.jp

² Shibaura Institute of Technology, 3–7–5 Toyosu, Koto-ku, Tokyo 135–8548, Japan

Abstract. Policy gradient methods are very useful approaches in reinforcement learning. In our policy gradient approach to behavior learning of agents, we define an agent’s decision problem at each time step as a problem of minimizing an objective function. In this paper, we give an objective function that consists of two types of parameters representing environmental dynamics and state-value functions. We derive separate learning rules for the two types of parameters so that the two sets of parameters can be learned independently. Separating these two types of parameters will make it possible to reuse state-value functions for agents in other different environmental dynamics, even if the dynamics is stochastic. Our simulation experiments on learning hunter-agent policies in pursuit problems show the effectiveness of our method.

1 Introduction

Environmental dynamics is usually represented by state-transition probabilities in reinforcement learning. However, it is not always necessary to know or learn environmental dynamics in advance when an agent learns its policy using reinforcement learning. For example, in Q -learning [1], which is a representative type of reinforcement learning, this information is included in action-value functions $Q(s, a)$ ($s \in S$, $a \in A$) and learned together with behavior knowledge for solving a task given to an agent. Therefore, both dynamics around an agent and knowledge for solving the task are learned simultaneously in action-value functions by Q -learning. The optimal policy for an agent is calculated by a greedy search of the action-value functions. This situation pertains even in TD -learning [1], where state-value functions $V(s)$ ($s \in S$) depend on environmental dynamics as $Q(s, a)$ in Q -learning.

Then what defines the environmental dynamics of an agent? There are at least two types of factors that affect environmental dynamics. The first is the behavior characteristics of each agent, such as the moving characteristics of a real mobile robot. The second is environmental conditions like a muddy field or a strong wind. As an example, let us consider a pursuit problem in which robot agents pursue and catch a prey robot. In such a problem, it may happen that a real robot has its own moving characteristics and the floor is very slippery. Then state transitions of the environment around learning agents are stochastic. If we change a hunter robot to another robot having different

moving characteristics or conduct a pursuit experiment in another room with a different type of floor material, we can no longer use the state-value functions obtained by previous learning experiments. If environmental dynamics and behavior knowledge were separated in an agent policy, it would be possible for either of them to be reused in other pursuit experiments where the dynamics or the task is changed. Fortunately, the former knowledge, i.e., the moving characteristics of a real robot, can be measured in advance or observed in a learning process. Of course, it can be learned together with the latter knowledge in our learning framework described in the next sections. Behavior knowledge independent of environmental dynamics can be learned by simulation where only robot agents with standard deterministic moving characteristics are used. If both types of knowledge were obtained and used as initial values of parameters in an agent policy, this would greatly help to reduce iteration times in learning experiments with real robots in the real world.

2 Policy Gradient and Learning Rule

2.1 Reinforcement Learning Based on Policy Gradient

Policy gradient methods originate from the REINFORCE algorithm of Williams [2]. In the REINFORCE algorithm, an agent policy includes parameters. The parameters are updated using policy gradient vectors to increase the expectation of rewards given to an agent. REINFORCE was extended to POMDPs (Partially Observable Markov Decision Processes) by Kimura [3]. In these methods, an agent policy is learned directly without calculating state-value function $V(s)$ or action-value function $Q(s, a)$. Consequently, there was a large discrepancy between value-based reinforcement learning algorithms and the primitive policy gradient methods. However, methods where policy gradient vectors are represented and calculated by action-value functions have been proposed assuming MDPs (Markov Decision Processes) [4][5]. We showed that the REINFORCE algorithm can be applied without requiring Markov properties on state transitions, rewards [6] and policies. Moreover, we derived the learning rule proposed in Ref.[4] and Ref.[5] from the learning rule of REINFORCE using statistical properties of characteristics eligibilities [7], and we showed that a policy gradient method can be applied to learning in multi-agent systems as pursuit problems [8]. We approximated the policy function controlling all agents by the product of policy functions of each agent [9]. Therefore, the primitive policy gradient method, REINFORCE by Williams, has been extended to learning agents in POMDPs, MDPs, non-MDPs, and multi-agent systems.

2.2 Objective Function and Policy

We use a Boltzmann-type stochastic policy. An objective function is used as energy in a Boltzmann distribution function. The objective function $E(a; s)$ evaluates action a of an agent in environmental state s , and it includes parameters. The parameters are learned to maximize the expectation of the reward given to an agent. In our previous paper, we applied a policy gradient method with a Boltzmann-type stochastic policy to

pursuit problems [8], where some deterministic environmental dynamics is given and all state-transition probabilities take the value of one or zero. This paper deals with stochastic environmental dynamics to verify whether the policy gradient method can be applied under environments that have stochastic dynamics.

We define policy $\pi(a; s, \{\theta\})$ as

$$\pi(a; s, \{\theta\}) \equiv \frac{e^{-E(a; s, \{\theta\})/T}}{\sum_{b \in A} e^{-E(b; s, \{\theta\})/T}}, \quad (1)$$

where $E(a; s, \{\theta\})$ is an objective function that evaluates action a of an agent in state s . When an agent was in a deterministic environment, a two-dimensional table,

$$E(a; s, \{\theta\}) = -\theta(s, a) \quad (2)$$

was used in our previous work on pursuit problems [8]. We propose a new type of objective function for stochastic dynamics in Section 3.

2.3 Learning Rule

In this paper, we consider only episodic learning. An episode consists of $\{a(t)\}$ and $\{s(t)\}$, which are time-series data on actions and states that an agent actually took and occupied in the episode. Let θ be a parameter in objective function $E(a; s, \{\theta\})$. The gradient vector of the expectation of reward r given in an episode can be written as [8]

$$\frac{\partial E[r]}{\partial \theta} = E \left[r \sum_{t=0}^{L-1} e_\theta(t) \right], \quad (3)$$

where L is time length of an episode from start to goal and $e_\theta(t)$ is characteristic eligibility [2] defined by

$$e_\theta(t) = \frac{\partial}{\partial \theta} \ln \pi(a(t); s(t), \{\theta\}). \quad (4)$$

If an agent policy is given by Boltzmann-type action selection in Eq. (1) [8],

$$e_\theta(t) = -\frac{1}{T} \left[\frac{\partial E(a(t); s(t), \{\theta\})}{\partial \theta} - \left\langle \frac{\partial E}{\partial \theta} \right\rangle_\pi \right]. \quad (5)$$

Operation $\langle \dots \rangle_\pi$ means taking an expectation weighted by the probability density function $\pi(a; s, \{\theta\})$ in Eq. (1).

We use the following learning rule according to the property of the gradient vector in Eq. (3) [7][8],

$$\Delta \theta = \varepsilon \cdot r \cdot \sum_{t=0}^{L-1} e_\theta(t). \quad (6)$$

$\varepsilon (> 0)$ is a learning ratio and parameter θ is updated at the end of each episode by Eq. (6). $e_\theta(t)$ means amount of eligibility that parameter θ should be reinforced by reward r .

3 Separation of Environmental Dynamics and State Values in Policy

3.1 Objective Function in Stochastic Environments

For stochastic environments, we propose objective function $E(a; s, \{\theta(s)\})$ defined by

$$E(a; s, \{\theta(s)\}) \equiv - \sum_{s'} \omega(s, s'; a) \theta(s') \quad (7)$$

where parameter $\theta(s)$ is task-dependent knowledge that evaluates state s and parameter $\omega(s, s'; a)$ represents a transition probability from state s to state s' when an agent takes action a . An agent does not always know the accurate state-transition probabilities given by the agent environments in advance of learning. If an agent has the exact information or can learn it by observation in the process of learning an agent policy, the agent can use the information as $\omega(s, s'; a)$ in Eq. (7) appropriately to select its action at each moment.

Objective function $E(a; s, \{\theta(s)\})$ in Eq. (7) means that action a of an agent in state s should be evaluated by state value $\theta(s')$ of state s' to which an agent is moved from state s by action a . Parameters $\omega(s, s'; a)$ represent stochastic properties of the agent environments. Then state value $\theta(s')$ weighted by $\omega(s, s'; a)$ will give better evaluation of action a when state transition is caused stochastically.

3.2 Characteristic Eligibilities and Learning Rules

If Eq. (7) is substituted into Eq. (5), we obtain

$$e_{\theta(s')}(t) = \frac{1}{T} [\omega(s(t), s'; a(t)) - \langle \omega(s(t), s'; a(t)) \rangle_{\pi}] \quad (8)$$

and

$$e_{\omega(s, s'; a)}(t) = \frac{1}{T} [\delta_{a, a(t)} - \pi(a; s(t))] \theta(s') \delta_{s, s(t)}, \quad (9)$$

where function $\delta_{s, s'}$ takes 1 if $s = s'$ else 0. Substituting eligibilities in Eq. (8) and Eq. (9) into Eq. (6) derives easily the following learning rules for stochastic environments:

$$\Delta \theta(s') = \frac{\varepsilon_{\theta} \cdot r}{T} \sum_{t=0}^{L-1} [\omega(s(t), s'; a(t)) - \langle \omega(s(t), s'; a(t)) \rangle_{\pi}] \quad (10)$$

$$\Delta \omega(s, s'; a) = \frac{\varepsilon_{\omega} \cdot r}{T} \sum_{t=0}^{L-1} [\delta_{a, a(t)} - \pi(a; s(t))] \theta(s') \delta_{s, s(t)}, \quad (11)$$

where ε_{θ} and ε_{ω} are learning ratios that take a small positive value.

According to Eq. (10), state-value parameter $\theta(s')$ of state s' to which an agent can be moved from state $s(t)$ by action $a(t)$ is updated in proportion to the deviation of

$\omega(s(t), s'; a(t))$ from its expectation value. State $s(t)$ and action $a(t)$ are a state of environments and an agent action. They appear at time t in an episode. Note that the expectation value $\langle \omega(s(t), s'; a(t)) \rangle_\pi$ does not depend on action $a(t)$ at time t , since it is defined by

$$\langle \omega(s(t), s'; a(t)) \rangle_\pi \equiv \sum_a \omega(s(t), s'; a) \pi(a : s(t)). \quad (12)$$

Parameter $\omega(s, s'; a)$, which corresponds to the transition probability $P_{s,s'}^a$, is updated by calculating the right-hand side of Eq. (11). Eq. (11) updates $\omega(s(t), s'; a(t))$, which represents a probability of transition from state $s(t)$ to state s' by action $a(t)$. $s(t)$ and $a(t)$ are a state and an action that an agent takes at time t . Parameter $\omega(s(t), s'; a)$ is increased by Eq. (11) when $a = a(t)$ in proportion to state-value parameter $\theta(s')$, because what is inside of the brackets $[]$ in Eq. (11) does not take any negative value. On the other hand, $\omega(s(t), s'; a)$ for action a that is not $a(t)$ is decreased so that the transition from state $s(t)$ to state s' caused by action a is suppressed, because what is inside of the brackets, i.e., $-\pi(a; s(t))$, takes a negative value. The degree of increasing/decreasing is reinforced in Eq. (10)/Eq. (11) by reward r given to an agent at the end of each episode.

Vector $(E[\Delta\theta], E[\Delta\omega])$, whose elements are expected amounts of change given by Eq. (10) and Eq. (11), is directed to the gradient vector $(\nabla_\theta E[r], \nabla_\omega E[r])$, because the inner product of the two vectors is non-negative, i.e.,

$$\begin{aligned} & (E[\Delta\theta], E[\Delta\omega]) \cdot (\nabla_\theta E[r], \nabla_\omega E[r])^T \\ &= \sum_{s'} \varepsilon_\theta \left[\frac{\partial E[r]}{\partial \theta(s')} \right]^2 + \sum_{s, s', a} \left[\frac{\partial E[r]}{\partial \omega(s, s'; a)} \right]^2 \geq 0. \end{aligned} \quad (13)$$

Eq. (13) indicates that update vector $(\Delta\theta, \Delta\omega)$ given by the learning rules in Eq. (10) and Eq. (11) moves in a direction where expectation reward $E[r]$ is increased on average, but not at every step.

4 Advantage of Separation of Dynamics and State-Values in Policy

Let us consider a pursuit problem using real mobile robots as an example. Fig. 1 shows a schematic diagram of the separation of environmental dynamics and behavior knowledge in a pursuit problem. As mentioned in Introduction, there are two types of factors that affect environmental dynamics in this example. The first one is the behavior characteristics of each hunter robot. The second one is environmental conditions such as the surface condition of the floor on which the hunter robots run to pursue a prey robot. In all cases, we must deal with the stochastic dynamics of environments. Stochastic dynamics is usually expressed as state-transition probability $P_{s,s'}^a$ in reinforcement learning. We represent the stochastic dynamics by parameter $\omega(s, s'; a)$ in agent policy $\pi(a; s)$ through objective function $E(a; s)$ in Eq. (7). The environmental dynamics of a real robot is different from that of other robots. However, it can be measured in advance. As a matter of course, it can be learned by the learning rule in Eq. (11); however, this costs much computation time.

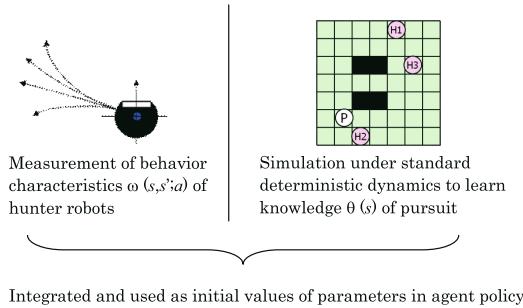


Fig. 1. Schematic diagram on the separation of environmental dynamics and behavior knowledge in a pursuit problem

In pursuit simulation with deterministic environmental dynamics, proper agent policy, which includes state-value parameters $\theta(s')$, can be learned by a policy gradient method described in Section 2 [8]. These are suitable for use as initial values of $\theta(s)$ in objective function Eq. (7), regardless of the environmental dynamics of each hunter robot. If we measured environmental dynamics in advance and obtained state-value parameters by simulation with standard deterministic dynamics, and used both types of prior knowledge as initial values of parameters in agent policies, we could considerably reduce iteration times in learning experiments with real robots.

5 Simulation

5.1 Pursuit Problem with Stochastic Dynamics

We conducted experiments to verify whether dynamics parameters and state-value parameters in an agent policy can be learned by the learning rules in Eq. (10) and Eq. (11). Our previous work dealt with simulation experiments for pursuit problems with deterministic environmental dynamics [8]. Here, we changed the dynamics to a stochastic one and applied our method proposed in Section 3 to simulation experiments of pursuit problems.

Let us consider a 2D grid world. It has a torus structure and consists of 7 by 7 squares. There are two hunter agents and a prey agent. Hunters pursue the prey until all hunters are adjacent to the square that the prey occupies. Only one agent can occupy the same square at one time. Agents move in a given order. The prey agent moves at random and does not learn anything. An episode ends if hunters catch the prey. The initial positions of hunters and prey are given randomly at the start of every episode. At each time step in an episode, a hunter takes and selects an action among five actions, which are "stop" and moves from the current square that it occupies to one of the four adjacent squares.

An agent's dynamics is stochastic. The agent's next position is determined by state transition probabilities $P_{s,s'}^a$. In the experiments of this section, we do not use real robots but realize stochastic dynamics of hunter agents in pursuit simulation by $P_{s,s'}^a$.

5.2 Experimental Conditions

The goal of the pursuit problems described in the previous section is to catch a prey as soon as possible. To solve these problems by reinforcement learning, we give hunter agents reward r as $r = 1/L^2$ at the end of each episode. L is the length of an episode.

As described in the next sections, we conducted learning experiments when the environmental dynamics was deterministic (Section 5.3) and stochastic (Section 5.4). In each learning experiment, we updated parameters $\omega(s, s'; a)$ and $\theta(s')$ one hundred thousand times and repeated each experiment ten times. Initial values of parameters $\omega(s, s'; a)$ are set to 0.2 and those of $\theta(s')$ are selected at random from $[0, 0.1]$ in each experiment. Temperature parameter T is set to 0.2. Learning ratios $\varepsilon_{\theta(s')}$ and $\varepsilon_{\omega(s, s'; a)}$ are set to 0.1 and 0.01, respectively. After each learning experiment, we conducted an evaluation experiment for $\omega(s, s'; a)$ and $\theta(s')$, where T is set to 0.01 and pursuit simulation is repeated for ten thousand episodes.

5.3 Experiment with Deterministic Dynamics

In Expt. 1, we assumed that hunter agents could move to any adjacent square where they intended to move. State transition probabilities $P_{s, s'}^a$ take 1 or 0. We learned the values of $\theta(s')$ when $\omega(s, s'; a)$ was set to $P_{s, s'}^a$. The average episode length observed in the last episode of ten learning experiments was 5.6. We also conducted an experiment to evaluate $\theta(s')$ obtained by learning. In the evaluation experiment, we set $T=0.01$ to simulate a greedy selection of action and repeated pursuit simulation ten thousand times. A greedy policy with correct value functions gives the optimal policy in value-based reinforcement learning algorithms such as TD-learning and Q-learning. The average episode length obtained in the evaluation experiment was 5.3. This is equal to the value obtained in our previous work, where Q-learning was applied to the same problems [8].

5.4 Experiment with Stochastic Dynamics

In this section, we assume stochastic environmental dynamics in which a hunter agent is moved with probability p in the direction that is 90 degrees right to the direction in which the agent intends to move: An agent's moving course is stochastically diverted 90 degrees to the right with probability p . Then state transition probabilities $P_{s, s'}^a$ take one of the values of 0, p or $1 - p$. Under this stochastic environment, we conducted four types of experiments, Expt. 2.1 to Expt. 2.4.

In Expt. 2.1 and Expt. 2.2, we learn $\theta(s')$ of hunter agents when $\omega(s, s'; a)$ is set to $P_{s, s'}^a$. Initial values of $\theta(s')$ are selected at random from $[0, 1]$ in Expt. 2.1. In Expt. 2.2, one set of state-value parameters $\{\theta(s')|s' \in S\}$ is selected at random from ten sets of $\{\theta(s')|s' \in S\}$ obtained when deterministic dynamics is assumed in Expt. 1. The set of state-values is used as initial values of $\theta(s')$ in Expt. 2.2. Expt. 2.3 learns $\omega(s, s'; a)$ with $\theta(s')$ fixed to the set of state-values. Expt. 2.4 learns both $\theta(s')$ and $\omega(s, s'; a)$ simultaneously. In Expt. 2.4, we updated $\theta(s')$ and $\omega(s, s'; a)$ five hundred thousand times in each learning experiment and used deterministic dynamics for initial values of $\omega(s, s'; a)$ to accelerate the learning.

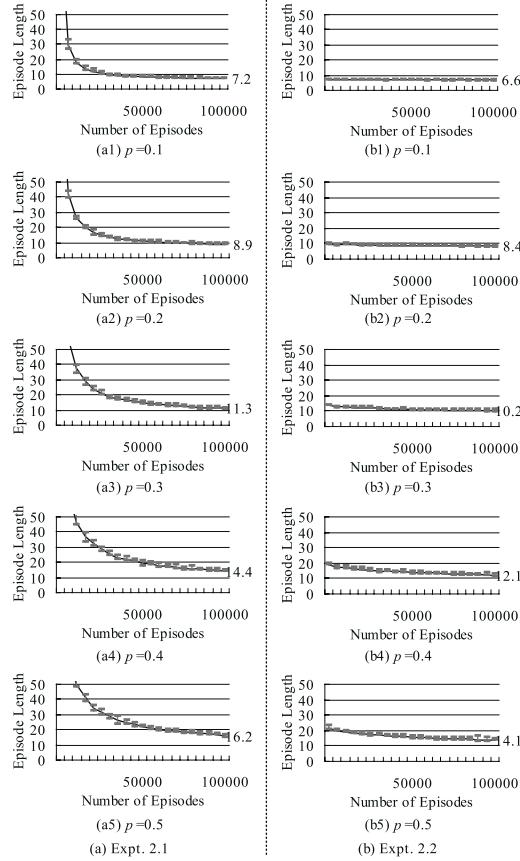


Fig. 2. Learning curves observed in Expt. 2.1 and Expt. 2.2

5.5 Experimental Results

Changes in episode length L averaged over 5000 episodes are shown in Fig. 2. Figures 2a and 2b are the learning curves obtained in Expt. 2.1 and Expt. 2.2, respectively. In Fig. 2, learning curves are averaged over ten trials for each experiment. The minimum and maximum among the ten trials are also depicted by error bars in Fig. 2.

Fig. 2a suggests that state values $\theta(s')$ in Eq. (7) can be learned by a policy gradient method described in Section 3 if correct dynamics $P_{s,s'}^a$ is given to $\omega(s, s'; a)$ in Eq. (7). From Fig. 2b, we can conclude that state values $\theta(s')$ obtained by learning under environments with deterministic dynamics can be used as initial values of $\theta(s')$ even when environmental dynamics is stochastic. This reuse as initial values reduces much computation time in learning.

The mean square averages between $\omega(s, s'; a)$ (i.e., the average of ten sets of $\omega(s, s'; a)$) obtained in Expt. 2.3 and the actual stochastic dynamics $P_{s,s'}^a$ are shown in Table 1 for different values of p . The table also lists the average length of the last episode

Table 1. Episode length and mean square average between $\omega(s, s'; a)$ obtained in Expt. 2.3 and the actual stochastic dynamics $P_{s,s'}^a$

	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$
Mean square difference between ω and P	0.023	0.019	0.017	0.021	0.025
Length of the last episode at learning stage	8.8	11.3	15.1	19.9	26.4
Average length over 10,000 episodes in the evaluation experiments ($T = 0.01$)	6.7	8.7	11.4	15.6	19.7

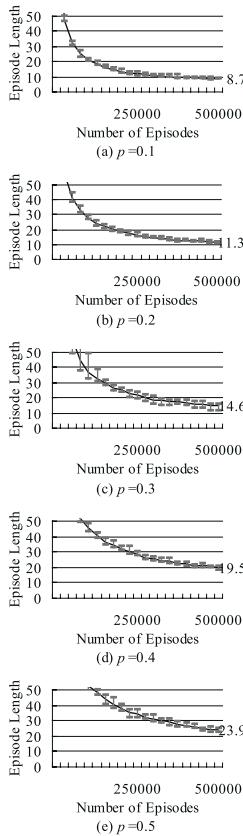


Fig. 3. Learning curves observed in Expt. 2.4

observed in ten learning experiments and the average length over ten thousand episodes in ten evaluation experiments ($T = 0.01$) for each value of p . Results of average length in the evaluation experiments suggest that an agent dynamics represented by $\omega(s, s'; a)$

Table 2. Episode length and mean square averages between $\omega(s, s'; a)$ and $P_{s,s'}^a$ observed in Expt. 2.4

	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$
Mean square difference between ω and P	0.022	0.013	0.011	0.009	0.012
Length of the last episode at learning stage	8.7	11.3	14.6	19.5	23.9
Average length over 10,000 episodes in the evaluation experiments ($T = 0.01$)	5.9	7.1	9.4	11.5	13.1

in Eq. (7) can be learned if state-value parameters $\theta(s')$ are given properly in Eq. (7). However, all episode lengths obtained are larger than those in Expt. 2.2, where $\theta(s')$ are learned with correct dynamics $\omega(s, s'; a)$. It seems that learning stochastic dynamics $\omega(s, s'; a)$ is more difficult than learning behavior knowledge $\theta(s')$ in the pursuit problems considered in this paper. Episode length seems to be affected by errors in agent dynamics more strongly than by errors in behavior knowledge.

Learning curves observed in Expt. 2.4 are shown in Fig. 3. The mean square averages between $\omega(s, s'; a)$ and $P_{s,s'}^a$, the average length of the last episode in ten learning experiments, and the average length over ten thousand episodes in ten evaluation experiments ($T = 0.01$) are listed in Table 2 for each value of p . Fig. 3 suggests that simultaneous learning of $\omega(s, s'; a)$ and $P_{s,s'}^a$ takes more computation time than learning only one of them. However, the results of evaluation experiments shown in Table 2 indicate that simultaneous learning of all parameters in Eq. (7) can be obtained by a policy gradient method described in Section 3 if initial values of $\omega(s, s'; a)$ are selected properly. In all experiments, episode length L becomes larger as probability p becomes larger from 0.1 to 0.5. This means that the task of pursuing a prey becomes more difficult as the environmental dynamics becomes more stochastic.

6 Conclusion

In this paper, we proposed a separation of parameters representing environmental dynamics and parameters representing state-value functions. Both types of parameters are included in an objective function, which is used as the policy function of an agent at each time step in policy gradient methods. We also derived learning rules for the parameters. Such a separation makes it possible to reuse state-value parameters for agents in different environmental dynamics. Conversely, the dynamics parameters of an agent can be reused in different tasks. Moreover, if we measured the environmental dynamics for each agent in advance and obtained state-values by simulation with standard deterministic dynamics, and used both values as initial parameter values in an agent's objective function, we could considerably reduce iteration times in learning both types of parameters, even when the dynamics is stochastic.

We considered pursuit problems where two hunter agents pursue a prey agent that moves at random in a 7 by 7 grid world. Our experiments show that the dynamics parameters and state-value parameters in each agent policy function can be learned even if agents cannot move deterministically but are diverted 90 degrees to the right stochastically. In the future, we will try our method using real mobile robots for pursuit problems in a real-world environment.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning. MIT Press, Cambridge (1998)
2. Williams, R.J.: Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning* 8, 229–256 (1992)
3. Kimura, H., Yamamura, M., Kobayashi, S.: Reinforcement Learning by Stochastic Hill Climbing on Discounted Reward. In: Proceedings of the 12th International Conference on Machine Learning, pp. 295–303 (1995)
4. Sutton, R.S., McAllester, D., Singh, S., Mansour, Y.: Policy Gradient Methods for Reinforcement Learning with Function Approximation. In: Advances in Neural Information Processing Systems (Proc. NIPS 1999 Conf.), vol. 12, pp. 1057–1063 (2000)
5. Konda, V.R., Tsitsiklis, J.N.: Actor-Critic Algorithms. In: Advances in Neural Information Processing Systems (Proc. NIPS 1999 Conf.), vol. 12, pp. 1008–1014 (2000)
6. Baird, L., Moore, A.: Gradient Descent for General Reinforcement Learning. In: Advances in Neural Information Processing Systems (Proc. NIPS 1998 Conf.), vol. 11, pp. 968–974 (1999)
7. Igarashi, H., Ishihara, S., Kimura, M.: Reinforcement Learning in Non-Markov Decision Processes—Statistical Properties of Characteristic Eligibility. *IEICE Transactions on Information and Systems* J90-D(9), 2271–2280 (2007) (in Japanese)
8. Ishihara, S., Igarashi, H.: Applying the Policy Gradient Method to Behavior Learning in Multi-agent Systems: The Pursuit Problem. *Systems and Computers in Japan* 37(10), 101–109 (2006)
9. Peshkin, L., Kim, K.E., Meuleau, N., Kaelbling, L.P.: Learning to cooperative via policy search. In: Proc. of 16th Conference on Uncertainty in Artificial Intelligence (UAI 2000), pp. 489–496 (2000)

Developing Evaluation Model of Topical Term for Document-Level Sentiment Classification

Yi Hu, Wenjie Li, and Qin Lu

Department of Computing,
The Hong Kong Polytechnic University,
Hong Kong, China

{csyihu, cswjli, csluqin}@comp.polyu.edu.hk

Abstract. Sentiment classification is used to identify whether the opinion expressed in a document is positive or negative. In this paper, we present an evaluation modeling approach to document-level sentiment classification. The motivation of this work stems from the observation that the global document classification can benefit greatly by learning how a topical term is evaluated in its local sentence context. Two sentence-level sentiment evaluation models, namely positive and negative models, are constructed for each topical term. When analyzing a document, the evaluation models generate divergence to support sentence classification that in turn can be used to decide on the whole document classification collectively. When evaluated on a public available movie review corpus, the experimental results are comparable with the ones published. This is quite encouraging to us and motivates us to further investigate how to develop more effective evaluation models in the future.

Keywords: Sentiment Classification; Evaluation Model; Topical Term; Maximum Spanning Tree.

1 Introduction

Sentiment classification is a recently rapidly growing sub-discipline of text classification concerned with the opinion expressed in a text rather than its topic. Document-level sentiment classification is targeted to classify a document according to the positive or negative polarity of its opinion. Automatically labeling documents such as product reviews or movie reviews with their sentiment polarities can be useful enough in many business intelligent systems and recommending systems.

Conventional topic-oriented classification models normally represent a document as a set of words in which topic sensitive words are important. In contrast, polarity words, such as “excellent” and “worst”, are considered essential to sentiment-oriented classification. However, we argue that sentiment structures in sentence context are more expressive than individual polarity words. Take the following sentence selected from a movie review as an example.

It is evidence that the **film** is **satisfying** all audience.

Clearly, the key polarity word “satisfy” in the context of “film” is a strong clue to indicate its positive opinion orientation towards the subject matter, i.e. to praise the film. Now let us look at another example.

The film never satisfies any audience beyond its visuals.

Although “satisfy” is a positive perspective in the first example, the negation word “never” in the second example, however, transforms it into a negative one. This implies that the same polarity word may deliver different perspectives when it appears in different contexts. The polarity transformation issue has been traditionally addressed by defining negation rules [5]. Unfortunately, rules are not always robust in NLP applications.

In our consideration, more helpful evidence should come from a broader, associated and structured context, such as “never satisfy any audience”, that can be represented by two pairs, i.e. <never, satisfy> and <satisfy, audience>. Notice that when the object of “satisfy” (i.e. “audience”) is replaced with “carper”, the polarity of “never satisfy any carper” becomes positive. Such a polarity transformation could not be achieved by rules alone, but the pair <satisfy, carper> might provide the necessary information. In order to better capture sentiment structures, we propose to develop evaluation models of topical terms.

In our definition, “topical terms” are those entities or aspects of entities in a particular domain, such as “film” and “actor”. They can be extracted according to domain termhood and can be used to approximately characterize the document topics. A sentence containing at least one topical term is called a “subjective sentence” in this paper. It is supposed to express some opinion to the topical term. The sentiment structure of a topical term in a subjective sentence can then be represented by some kind of structures within a specified context surrounding the topical term, such as the maximum spanning tree (MST for short) where the root node is the topical term. When mining the polarity of a sentence, the generation probabilities of its MST structures regarding to the positive or negative polarities are used for classification. In this connection, we need to train two so-called evaluation models for each topical term in order to support probability calculation. The evaluation models manage to capture the habitual use of a language within the context of a topical term.

The rest of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 introduces the proposed evaluation models and Section 4 presents the model parameter estimations and the smoothing techniques. Section 5 conducts evaluations. Finally, Section 6 concludes the paper.

2 Related Work

As one of the opinion mining tasks, sentiment classification has attracted tremendous research attention for its broad applications in many domains such as movie reviews and customer feedback reviews.

A large body of research focuses on identifying semantic orientation of individual words or phrases by employing linguistic heuristics [5][13]. Usually, a measure of the strength of the sentiment polarity is developed to determine how strongly a word or phrase is positive or negative. For example, Turney and Littman [14] determine semantic orientation by phrase Pointwise Mutual Information (PMI) based on

pre-defined seed words, such as “excellent” and “poor”. Kennedy [5] uses negations and intensifiers to adjust the semantic polarity of a particular word.

When machine learning approaches become predominant in many text classification tasks, they are also applied to recognize sentiment polarity. By comparing different learning algorithms, Pang et al [9] conclude that SVM in general obtain better results with unigram features. Later on, Pang and Lee [10] further advance their previous study by training a sentence subjectivity classifier and then determining document sentiment polarity relying on the identified subjective sentences. In the work of [1], Bayesian belief networks are used to represent a Markov Blanket, i.e. a directed acyclic graph on which each vertex represents a word and the edge corresponds to the parent/child relationship between the words. Very recently, the mutual influence of document-level and sentence-level sentiment polarities are stressed by [7]. They investigate a joint sentiment classification framework which incorporates the classifications at both sentence and document levels. As a matter of fact, numerous research articles focusing on document classification have utilized sentence analysis. This implies that the sentence information is advantageous to document classification.

On the other side, efforts of mining opinions from sentences have been made on the task of extracting the templates <who feels how about which aspect of what product> from unstructured text data. For example, In contrast to sentiment classification, opinion extraction aims to produce discrete information and requires an in-depth analysis of opinion [4][11][15].

3 Evaluation Models for Document Sentiment Classification

We introduce an evaluation modeling approach to document sentiment classification in this section. The evaluation models are developed to capture and evaluate the links among a topical term and its context words from both the positive and the negative perspectives. The motivations of this approach are two fold. First, the context helps to determine the sentiment structure of a topical term. Second, the generation probabilities of a certain sentiment structure in “positive” and “negative” evaluation models are likely to be substantially different. When analyzing a document, the evaluation models generate divergence that supports sentence classification that in turn can be used collectively to decide on the whole document classification.

3.1 A General Probability Framework for Sentence Sentiment Classification

The sentiment structure “ Str ” of a topical term t in a subjective sentence “ s ” can be formulated by any general probabilistic framework. In this paper, we choose the following log-ratio decision function f to determine the polarity of sentence s .

$$\begin{aligned} f(s) &= \log \left(\frac{\Pr(POS|s)}{\Pr(NEG|s)} \right) = \log \left(\frac{\Pr(POS|t, Str)}{\Pr(NEG|t, Str)} \right) \\ &= \log \left(\frac{\left(\frac{\Pr(POS|t) \Pr(Str|POS, t)}{\Pr(Str|t)} \right)}{\left(\frac{\Pr(NEG|t) \Pr(Str|NEG, t)}{\Pr(Str|t)} \right)} \right) = \log \left(\frac{\Pr(Str|POS, t)}{\Pr(Str|NEG, t)} \right) \end{aligned} \quad (1)$$

where s is further described by a tuple of t and Str of t , i.e., $s = (t, Str)$. POS indicates the positive and NEG the negative. In (1), it is not necessary for t to have its own polarity. Therefore the probability $Pr(POS|t)$ equals to $Pr(NEG|t)$. Both of them can be ignored from the function. The probabilistic framework illustrated in (1) allows the generation probabilities of Str to be used for the sentiment classification of s .

It should be noted that Str here is a general representation of the sentiment structure. Referring back to the examples in Section 1, the sentiment structure Str of the context “never satisfy any audience” can be $\langle \text{never}, \text{satisfy} \rangle$ and $\langle \text{satisfy}, \text{audience} \rangle$, though the ideal structure might be a higher order “ $\langle \text{never}, \langle \text{satisfy}, \text{audience} \rangle \rangle$ ”. The higher order structure is able to capture more accurate information for classification, but it also suffers from more severe data sparseness problem.

3.2 Maximum Spanning Tree (MST) Based Sentiment Structure

As just mentioned, the sentiment structure Str can be represented by any suitable formal structure. In this paper, we exploit the Maximum Spanning Tree (MST) structure to discover the links among t and its context words. For the MST choice, we have the following assumptions:

- (1) It is hard to decide which words in the context of t ought to be included or excluded from the sentiment structure. So all the words in the context are taken into consideration. Based on this model, a word will link to at least one other word, and unconnected words are not allowed.
- (2) Calculating all the links between the words is unnecessary because useful links are concealed in a redundant link set. On the other hand, calculating the redundant links is time-consuming. So only the significant links are preserved.

Obviously, the tree style of MST can cover all the words and has the least number of links [12]. Therefore, when the links between the nodes form a clique, we simply prune the weakest one in the clique to guarantee a tree. The partial MST of the sentence “It is evidence that the **film** is **satisfying** all **audience**” is illustrated in Figure 1. The topical term “film” is the root node in this MST.

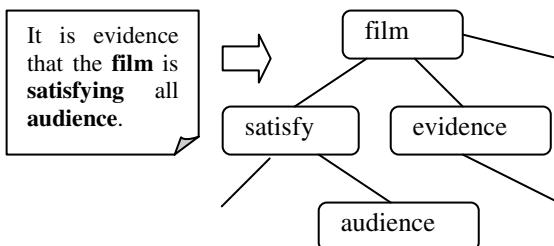


Fig. 1. The Partial MST of the instance

In constructing the MST structure, the weight of the link between a pair ($\langle \text{satisfy}, \text{audience} \rangle$ for example) is measured by the PMI [14], e.g.

$$\text{weight}(\text{satisfy}, \text{audience}) = \log\left(\frac{\Pr(\text{satisfy}, \text{audience})}{\Pr(\text{satisfy})\Pr(\text{audience})}\right) \quad (2)$$

where $\Pr(\text{satisfy})$, $\Pr(\text{audience})$ and $\Pr(\text{satisfy, audience})$ are obtained from a background corpus (including both subjective and non-subjective sentences) by using Maximum Likelihood Estimation. For a sentence with n words, its MST must have $n-1$ links because of the acyclic aspect of a tree. The $n-1$ links with the highest weights can link all the n words and build the MST of the sentence.

We assume that the MST generation probability of the sentence in the positive evaluation model is $\Pr(\text{MST|POS}, t)$, and the generation probability in the negative evaluation model is $\Pr(\text{MST|NEG}, t)$. If $\Pr(\text{MST|POS}, t)$ is larger than $\Pr(\text{MST|NEG}, t)$, the sentence is determined as positive, and vice versa.

It should be noted that a sentence may contain more than one topical term or to say a sentence may create more than one (t, Str) tuple. For any t of concern, the other terms are all deemed as the ordinary words. Each (t, Str) from the same sentence has its own generation probability, and the largest one decides the sentence polarity, i.e.

$$\Pr(s) = \max_{t \in s} \Pr(\text{Str} | Y, t) \quad (3)$$

where Y indicates the tag of POS or NEG .

3.3 Generation Probability of MST

A general MST structure is illustrated in Figure 2.

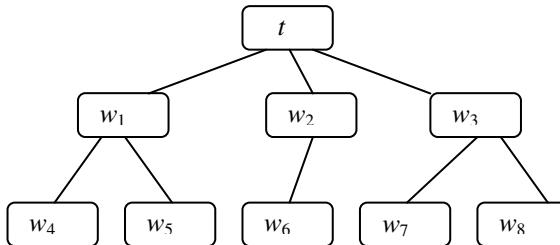


Fig. 2. The general structure of a sentence MST

In Figure 2, the topical term t is the root of the MST, w_1-w_8 are the context words. The reason for choosing t as the root is in accordance with the thought that we use the sentence context to evaluate the opinion (implicit or explicit in w_1-w_8) instead of the topical term t . Yet the term t is the trigger of this evaluation and therefore we start from it to calculate the generation probability of a MST. In the tree structure, each non-leaf word node can link to more than one word node, while the leaf nodes have only one node to link to. To calculate the MST, we split the whole MST into various individual sub-trees. A sub-tree is the one which only comprises a local parent and its directly-linked children.

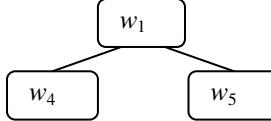
**Fig. 3.** A sub-tree instance from the MST

Figure 3 above illustrates a sub-tree instance from the MST. Taking it as an example, we explain how the sub-tree generation probability given t and Y can be calculated by (4).

$$\Pr(\text{subtree} | Y, t) = \Pr(w_1 | Y, t) \times \Pr(w_4 | w_1, Y, t) \times \Pr(w_5 | w_1, Y, t) \quad (4)$$

where $Y=NEG$ or $Y=POS$) (4) calculates the generation probability of the parent node w_1 in one sub-tree. If we assume the independence of sub-trees, we will have the generation probability of the whole MST in (5).

$$\begin{aligned} \Pr(MST | Y, t) &= \prod_{\langle w_i, w_j \rangle \in MST} \Pr(\langle w_i, w_j \rangle | Y, t) \\ &= \Pr(t | Y, t) \prod_{w \in W, w \neq t} \Pr(w | Y, t) \prod_{\langle w_i, w_j \rangle \in MST} \Pr(w_j | w_i, Y, t) \end{aligned} \quad (5)$$

where W indicates the bag-of-word of s , and the pair $\langle w_i, w_j \rangle$ indicates a link between w_i and w_j in the MST of s . Because the topical term t is the given condition, $\Pr(t | Y, t)$ equals to 1, and can be discarded.

The polarity of the sentence based on (5) can then be determined by

$$\begin{aligned} f(s) &= \log \left(\frac{1 \times \prod_{w \in W, w \neq t} \Pr(w | POS, t) \times \prod_{\langle w_i, w_j \rangle \in MST} \Pr(w_j | w_i, POS, t)}{1 \times \prod_{w \in W, w \neq t} \Pr(w | NEG, t) \times \prod_{\langle w_i, w_j \rangle \in MST} \Pr(w_j | w_i, NEG, t)} \right) \\ &= \log \left(\frac{\prod_{w \in W, w \neq t} \Pr(w | POS, t)}{\prod_{w \in W, w \neq t} \Pr(w | NEG, t)} \right) + \log \left(\frac{\prod_{\langle w_i, w_j \rangle \in MST} \Pr(w_j | w_i, POS, t)}{\prod_{\langle w_i, w_j \rangle \in MST} \Pr(w_j | w_i, NEG, t)} \right) = f_U(s) + f_L(s) \end{aligned} \quad (6)$$

Finally, (6) represents the summation of two terms, suggesting that the calculation of generation probability can be split into two parts. Obviously, the first part is a unigram model and the second part is a link model. We use f_U and f_L to refer to the decision functions based on unigrams and links, respectively. If a MST is more likely generated by positive than by negative polarity, it has a higher chance to provide the positive perspective rather than the negative one, and vice versa. Such a combined form is close to the one in [8]. But their combined model for topic detection is just pieced together artificially, while ours is deducted from a general probability framework.

3.4 Document Sentiment Classification

This study focuses on the sentiment classification of a document by comparing the generation probabilities of a collection of MSTs generated from the subjective sentences (indicated by C^*) in the document. The difference is derived from the positive

evaluation models (S^P) and the negative evaluation models (S^N)¹. The log-ratio decision function is defined in (7).

$$f(d) = \log \left(\frac{\Pr(C^{s^*} | S^P)}{\Pr(C^{s^*} | S^N)} \right) = \log \left(\frac{\prod_{MST \in C^{s^*}} \Pr(MST | S^P)}{\prod_{MST \in C^{s^*}} \Pr(MST | S^N)} \right) \quad (7)$$

(7) assumes the independence of the subjective sentences in the document d . Take a sentence s_q^* in the document d for example. The symbols t_q , MST_q and W_q indicate the term contained in s_q^* , the corresponding MST structure and the word bag of s_q^* . Using the logarithm rule, (7) can be rewritten into (8).

$$\begin{aligned} f(d) &= \sum_{s_q^* \in C^{s^*}} \sum_{w \in W_q} \log \left(\frac{\Pr(w | POS, t_q)}{\Pr(w | NEG, t_q)} \right) + \sum_{s_q^* \in C^{s^*}} \sum_{w_i, w_j \in MST_q} \log \left(\frac{\Pr(w_j | w_i, POS, t_q)}{\Pr(w_j | w_i, NEG, t_q)} \right) \\ &= f_U(d) + f_L(d) \end{aligned} \quad (8)$$

By taking the combined form, it is convenient for us to compute each individual part in (8) one by one. Also the combined form allows us to balance the contributions of the components. As a result, the decision function can be converted to (9) by linear interpolation. We investigate the contribution of each part and the coefficient θ in Section 5.

$$f(d) = \theta f_U(d) + (1 - \theta) f_L(d): \begin{cases} > 0 & POS \\ < 0 & NEG \end{cases} \quad (9)$$

Notice that the neutral cases are not considered in this study.

4 Model Parameter Estimation

When identifying the opinion polarity of a review, two kinds of parameters, $\Pr(w|Y, t)$ and $\Pr(w_j|w_i, Y, t)$, in the evaluation models need to be estimated. Let all the subjective sentences in training data consist of the training collection $T_Y^{s^*}$, i.e.

$$T_Y^{s^*} = \{s^* \mid \text{Polarity}(s^*) = Y\} \quad (10)$$

We assume the polarity of each training subjective sentence is directly inherited from the training document that contains it. This makes it easy to compile training sentences since we do not need to annotate a large number of subjective sentences manually and thus avoid a time-consuming effort. However, we are aware that a document with positive perspective may contain sentences that convey a negative point of view, and vice versa. This issue will be addressed in our future work.

We use the maximum likelihood estimate (MLE) to estimate the parameters $\Pr(w|Y, t)$ in the unigram models and the parameters $\Pr(w_j|w_i, Y, t)$ in the link models. In addition, we choose the Kneser-Ney algorithm to smooth $\Pr(w_j|w_i, Y, t)$.

¹ S^P is the set of positive evaluation models of all topical terms, and S^N has the corresponding meaning.

4.1 MLE for $\Pr(w|Y,t)$ and $\Pr(w_j|w_i,Y,t)$

It has been concluded in [9] that unigrams are credible in sentiment classification. Therefore, we employ the MLE to estimate the unigram models but put more efforts on the link models with additional consideration on the smoothing issue. That is,

$$\Pr_{MLE}(w|Y,t) = \frac{c(w|Y,t)}{c(*)|Y,t)} \quad (11)$$

$c(w|Y,t)$ is the number of times that the word w occurs in $T_Y^{s^*}$ ($Y=POS$ or NEG). $c(*)|Y,t)$ is the total number of the words (indicated by “*”) in $T_Y^{s^*}$. w and t appear in the same sentence, as well as * and t .

Likewise, the MLE is applied to estimate the parameters of the link models.

$$\Pr_{MLE}(w_j|w_i,Y,t) = \frac{c(w_j|w_i,Y,t)}{c(*)|w_i,Y,t)} \quad (12)$$

4.2 Kneser-Ney Smoothing for Link Model

Kneser and Ney [6] introduce an extension of absolute discounting smoothing by combining the lower-order distribution with the higher-order distribution. Chen and Goodman [3] then advance the Kneser-Ney’s algorithm by selecting the lower-order distribution. They demonstrate that Kneser-Ney smoothing performs best when compared with other commonly-used smoothing techniques, by testing on different conditions. We apply Chen’s revised Kneser-Ney algorithm here.

For the link models, we define the smoothed distribution P_{KN} to be the following modified form of Kneser-Ney smoothing.

$$\Pr_{KN}(w_j|w_i,Y,t) = \frac{\max\{c(w_i,w_j|Y,t)-D,0\}}{\sum_{w_j} c(w_i,w_j|Y,t)} + \frac{D}{\sum_{w_j} c(w_i,w_j|Y,t)} N_{1+}(w_i,\bullet|Y,t) \Pr_{KN}(w_j|Y,t) \quad (13)$$

where D is the fixed discount from observed links [3]. The notations N_{1+} and “•” are meant to evoke the number of words that have one or more counts, and a free variable that is summed over. Specifically,

$$\Pr_{KN}(w_j|Y,t) = \frac{N_{1+}(\bullet,w_j|Y,t)}{N_{1+}(w_i,\bullet|Y,t)} \quad (14)$$

5 Experiments and Discussion

The evaluation models are tested on the corpus of “movie review” provided by [10]. This corpus contains 1000 positive and 1000 negative reviews². In all the experiments, only the stemmed words that occur more than twice in the 2000 reviews are considered. Stop words are excluded³. The remaining stemmed words constitute the vocabulary of the models.

² <http://www.cs.cornell.edu/people/pabo/movie-review-data/>

³ There are 294 words in our stop-word list.

5.1 Experiments on Topical Term Extraction

In this study, the topical terms are semi-automatically extracted from the movie corpus (M) by comparing M with a non-movie review background corpus (B), and then selecting the top-ranked terms as the topical terms manually.

Intuitively, the topical terms ought to be domain relevant and rarely appear in other domains. We investigate the Inverse Word Frequencies (IWF), defined in [2] as our filter (i.e. the logarithm part in (15)). Given a candidate w , assume its frequency in the movie review corpus is $fre_M(w)$ and its frequency in the background corpus is $fre_B(w)$. The termhood of w is defined by (15).

$$Termhood(w) = fre_M(w) \log\left(\frac{N}{fre_B(w)}\right) \quad (15)$$

where N is the size of the background corpus. Human selection is intervened after termhood calculation. At last, we refer the top 30 terms ranked by termhood as the topical terms (see Appendix). Although a variety of methods have been proposed to select the most significant terms, the focus of this work is to examine the effect of the evaluations.

Table 1 below shows the examples of the links directly associating the terms “film”, “plot” and “movie” with the words “good”, “great” and “bad”. The values in the table denote the probabilities of these links in either the positive or the negative models, which are true in accordance to our general understanding.

Table 1. Probabilities of Example Links

	film			plot			movie		
	good	great	bad	good	great	bad	good	great	bad
Positive	0.056	0.020	0.004	0.029	0.016	0.003	0.061	0.018	0.012
Negative	0.048	0.012	0.019	0.011	0.007	0.019	0.057	0.012	0.044

5.2 Experiments on Sentiment Classification

Three sets of experiments are conducted to evaluate and discuss the proposed approach. The first set of experiments compares five different models by setting $\theta = 0.7$ and selecting the first 10 topical terms to guide the link extraction process. The second set of experiments changes the value of the interpolation parameter θ in the combined model to examine how the link models contribute to the combined models. Finally, the last set of experiments checks if the number of the topical terms is a key feature. The performance is evaluated according to the average accuracy by 10-fold cross validation.

The second column in Table 2 presents the results of the five models. They are the unigram model (i.e. f_{U_MLE}), the link model estimated by MLE only (i.e. f_{L_MLE}), the smoothed link model (i.e. f_{L_KN}), the non-smoothed combined model (i.e. $f_{U_MLE} + f_{L_MLE}$) and the smoothed combined model (i.e. $f_{U_MLE} + f_{L_KN}$). The next two columns are the percentages of the changes over the unigram model and the link model estimated by MLE only, respectively.

It shows that the smoothed combined model performs the best and achieves a significant improvement of 5.5% when compared with the unigram model. That is to say, the improvement by integrating the links is promising. In addition, the smoothed link models outperform the non-smoothed link models, either combined with the unigram model or not. The link models alone clearly perform the worst. This is understandable. The insufficient link data hurt their performance. That is why currently the unigram model is a more important component in the combined models. Overall, the results are encouraging to us. The accuracy of the smoothed combined model is comparable with the published results which are around 0.86 on this data set and by 10-fold cross validation too [10]. However, there is still much room for improvement.

Table 2. Average Accuracy of Five Models

Models	Avg. Acc.	Change (%)	Change (%)
f_{U_MLE}	0.820	--	+13.1
f_{L_MLE}	0.725	-11.6	--
f_{L_KN}	0.755	-7.3	+4.1
$f_{U_MLE} + f_{L_MLE}$	0.840	+2.5	+15.9
$f_{L_MLE} + f_{L_KN}$	0.865	+5.5	+19.3

Figure 4 plots the average accuracy of a series of comparative experiments by tuning the value of the parameter θ from 0.1 to 0.9 with the step of 0.1. These experiments are conducted for all the five models presented in Table 2. At most θ value points, the smoothed combined model stands above the non-smoothed combined model. Once again, it acknowledges the benefit from the use of Kneser-Ney smoothing. Besides, it is easy to conclude from Figure 4 that the links make an incontestable contribution to the overall performance of the combined models for most value points.

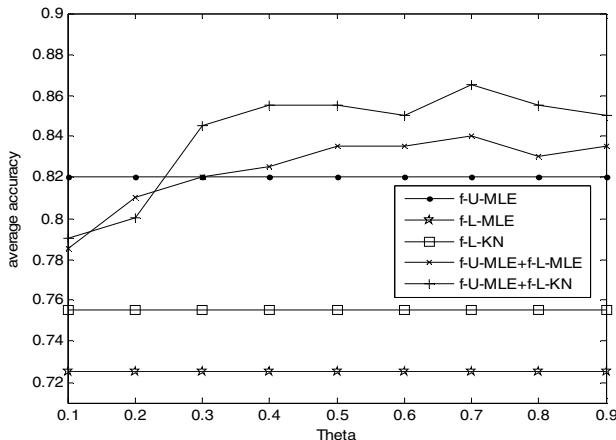


Fig. 4. Average Accuracy by Tuning x Value of the interpolation parameter θ

When θ equals to 0.7, it achieves the best result for $(f_{U_MLE}+f_{L_KN})$. We can also exploit the EM algorithm to estimate the value of theta.

To see whether more terms can provide more informative links, we conduct another set of experiments by choosing the number of topical terms from 2 to 30. These experiments are conducted with the smoothed combined model and the parameter θ is set to 0.7. The results in Table 3 suggest that 10 is a good choice. It is not necessary to use a larger term set on the current data set, because more terms would bring more “noise” into the model. We believe that the selection of the topical terms is essential to link models because of the sparse training data.

Table 3. Average Accuracies by Choosing Different Number of Topical Terms

# of Terms	2	4	6	8	10	12	15	20	25	30
Avg. Acc.	0.815	0.830	0.830	0.845	0.865	0.840	0.845	0.835	0.820	0.830

6 Conclusion

In this paper, we present a novel evaluation modeling approach to sentiment classification.

- (1) We assume that a topical term and its context can help to determine the sentence polarity;
- (2) We investigate the ability of capturing sentiment structures by constructing MSTs;
- (3) We create sentiment evaluation models by learning from the selected topical terms and their specified contexts;
- (4) We also discuss how to refine the evaluation models through the Kneser-Ney smoothing technique.

The experiments on a public available movie review corpus show that the proposed approach to sentiment classification is reasonably good. The results are comparable with the existing published results on the same corpus. In particular, the improvement from the evaluation models is encouraging. It shows that the proposed approach is able to learn the positive and negative contextual knowledge effectively in a supervised manner.

This study may suggest a direction to develop more effective evaluation models for sentiment classification. It drives us to further study how to define and make use of the link information appropriately and effectively.

It should be noted that our goal is not to show that our approach can perform much better than the classical machine learning method, but to investigate the role of evaluation models in document-level sentiment classification. In the future, we plan to improve the evaluation model based method by exploring the hierarchical link models, and integrating conceptual features into the models.

Acknowledgments. The research work presented in this paper was supported by a grant from the RGC of the HKSAR (Project No: PolyU 5211/05E).

References

1. Bai, X., Padman, R., Aioldi, E.: Sentiment extraction from unstructured text using tabu search-enhanced markov blanket. In: Proceedings of International Workshop on Mining for and from the Semantic Web (2004)
2. Basili, R., Moschitti, A., Pazienza, M.: A text classifier based on linguistic processing. In: Proceedings of IJCAI 1999. Machine Learning for Information Filtering (1999)
3. Chen, S.F., Goodman, J.: An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report: TR-10-98, Harvard University (1998)
4. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the 10th International Conference on KDD, pp. 168–177 (2004)
5. Kennedy, A., Inkpen, D.: Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters. Computational Intelligence 22(2), 110–125 (2006)
6. Kneser, R., Ney, H.: Improved backing-off for m-gram language modeling. In: Proceedings of the IEEE Internaltional Conference on Acoustics, Speech and Signal Processing, vol. 1, pp. 181–184 (1995)
7. McDonald, R., et al.: Structured Models for Fine-to-Coarse Sentiment Analysis. In: Proceedings of the 45th ACL, pp. 432–439 (2007)
8. Nallapati, R., Allan, J.: Capturing Term Dependencies using a Language Model based on Sentence Trees. In: Proceedings of the 11th CIKM, pp. 383–390 (2002)
9. Pang, B., et al.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Proceedings of EMNLP (2002)
10. Pang, B., Lee, L.: A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In: Proceedings of the 42nd ACL, pp. 271–278 (2004)
11. Popescu, A., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of EMNLP, pp. 339–346 (2005)
12. Rigsbergen, V.: Information Retrieval, Butterworths (1979)
13. Turney, P.D.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of ACL, pp. 417–424 (2002)
14. Turney, P.D., Littman, M.L.: Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical Report EGB-1094, National Research Council, Canada (2002)
15. Yi, J., et al.: Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In: Proceedings of the 3rd ICDM, pp. 427–434 (2003)

Appendix: 30 Selected Topical Terms

“film”, “movie”, “character”, “scene”, “time”, “story”, “play”, “plot”, “show”, “performance”, “star”, “actor”, “director”, “action”, “role”, “audience”, “comedy”, “fact”, “cast”, “script”, “act”, “part”, “screen”, “picture”, “hollywood”, “feature”, “series”, “writer”, “dialog”, and “box”

Learning to Identify Comparative Sentences in Chinese Text

Xiaojiang Huang, Xiaojun Wan, Jianwu Yang, and Jianguo Xiao

Institute of Computer Science & Technology of Peking University,

Beijing 100871, China

{huangxiaojiang,wanxiaojun,yjw,xiaojianguo}@icst.pku.edu.cn

Abstract. Identifying comparative sentences in natural language is an important step for extracting comparative relations. To our knowledge, there is no research on identifying Chinese comparative sentences automatically. This paper first defines the problem of Chinese comparative sentence identification, and then proposes to use several classifiers to classify a Chinese sentence into either “comparative” or not. Various linguistic and statistical features have been explored, such as keywords and sequential patterns. Experimental results demonstrate the good effectiveness of the sequential patterns, i.e. the classifiers with sequential patterns can significantly outperform the traditional term-based classifier. We also empirically investigate the important factors that affect classification performance.

Keywords: Chinese comparative sentence identification, comparative mining, text classification, sequential pattern.

1 Introduction

Extracting comparative relations between objects can lead to many applications. For example, we can tell consumers about the differences between similar products; give managers similar cases for reference; help people know each other better by finding their common interests, etc.

Comparative relation mining is a new task in the NLP and IR fields, and a few researches have been done on this task. Jindal and Liu proposed methods to identify comparative sentences in English corpus[1], and then extract elements of comparisons[2]. Zhai et al. researched on mining similarities and differences among multiple text sets with a cross-collection mixture model[3]. Sun et al.[4] and Luo et al.[5] compared two objects by using each object as a query to a search engine and then matching the results. Feldman et al. studied extracting comparison between products from forum discussions[6].

So far, the research on Chinese comparative sentences focuses only on linguistic issues, including the definition and boundary[7,8], the typical forms[9], the semantic[10], the evolution, and the comparative analysis between Chinese and foreign languages, or between Mandarin and dialects[11]. To the best of our knowledge, there is no research on automatically identifying Chinese comparative relations yet.

In this paper, we try to use data mining technologies to identify Chinese comparative sentences automatically. We first analyze Chinese comparative sentences, clarify some ambiguous sentence forms, and define the task of Chinese comparative sentence identification. We then use several classifiers to classify a Chinese sentence into either “Comparative” or not, by using linguistic and statistic features, including keywords and POS tags, as well as sequence patterns mined from corpus. We also investigate some factors that can affect the pattern mining and classification performances, including the sequence generating strategy and the maximal pattern length. Our experimental results show that the keyword-based classifiers can obtain high precision but low recall, while the pattern-based classifiers can improve the recall and F-measure, though losing a little precision. In addition, the pattern-based SVM classifier reaches its top performance when using short patterns mined from sequence dataset generated by sub-sentences. Overall, our approach can identify Chinese comparative sentence effectively.

The rest of this paper is organized as follows: The next section discusses the problem of Chinese comparative sentence identification. Section 3 presents the proposed methods. Section 4 gives the evaluations and Section 5 concludes this paper and talks about some future works.

2 Problem Definition

2.1 Chinese Comparative Sentence

Comparative sentences exist in almost all languages. Lerner defined comparative as “universal quantifiers over degrees” [12], while Stassen proposed that “a construction in a natural language counts as a comparative construction” “if that construction has the semantic function of assigning a graded position (i.e. non-identical) on a predicative scale to two (possibly complex) objects” [13]. Chinese linguists have been researching in comparison since the Chinese grammar system was founded in 1898[7]. Generally speaking, most scholars have achieved agreement on its semantic concepts, but argue on the coverage and boundary.

In concept, a Chinese comparative sentence (or CCS for short) is a sentence that assigns positions on a scale to two or more objects. The scale presents some property, state, or affair of these objects. The positions can be similar or different, while the difference can be either gradable (greater / less) or non-gradable. By these different relations, the category of CCSs can be further partitioned into several subcategories. The taxonomic hierarchies are shown in Table 1.

A typical CCS contains four basic elements, including the comparee (i.e. what is compared), the standard (i.e. to what the comparee is compared), the parameter (i.e. the scale on which the comparee and standard are measured), and the result (i.e. the predicate that describes the positions of comparee and standard). For example:

Example 1. 中国的人口数量比美国大. (China has a larger population than US.)

Table 1. Hierarchies of Chinese Comparative Sentences

Subcategory		Sample
Equative	Same	我和他一样高(I am as tall as him)
	Similar	我和他差不多高(My height is similar to his)
Differential	Gradable	我比他高(I am tall than him)
		我比他矮(I am shorter than him)
	Non-Gradable	我和他身高不同(My height is different from his)

In this sentence, “中国”(China) is the comparee; “美国” (US) is the standard; the parameter is “人口数量” (population); and the result is “大” (larger). Note that some CCSs may omit one or more elements, if they are clear in the context.

According to the definition, some sentences can be easily identified as comparative, such as “ X 比 Y R ” (X is more R than Y) , “ X 有/没有 Y R ” (X is / [is not] as R as Y), “ X 跟 Y 一样 R ” (X is same R as Y), where X and Y are two objects, and R is an adjective. Che also collected some common forms in [14]. However, there are still quite a few ambiguous sentences, which we try to clarify as follows:

- Sentences having form as “越…越…” (the more..., the more...) or “越来越...” (more and more...) are NOT comparative, because it is hard to ensure the comparees and standards.
- Sentences like “连...都/也...” (even...) are NOT comparative. Actually, they are minor premises of a kind of a syllogisms whose conclusions have meanings of similarity. For example:

Example 2. 连小孩都懂这个道理。(Even children understand this reason.)

Given the major premise “Adults understand reasons better than children”, the author of Example 2 means that all adults ought to understand this reason too. However, the comparison relies on other knowledge, but is not expressed directly by the sentence, thus we do not consider this sentence as comparative.

- Sentences like “ X 比较 R ” (X is a bit R) are NOT comparative. The semantic of word “比较” is ambiguous. It can mean “a bit”, which is absolute, and can also mean “middle of degree”, which is related to some standards. However, the standards are usually uncertain.
- A sentence is NOT comparative, if its chunk is not comparative. For example, the sentence “最终我们获得了成功” (We succeed at last) is not comparative, because its chunk is “我们获得了成功” (We succeed), and the comparative component “最终” (at last) is only used as a modifier.
- Sentences like “与其…, 不如...” (rather than) ARE comparative, for they do propose that something is better than the other thing.
- Contrast sentences ARE comparative. Contrast sentences usually have two sub-sentences, each of which describes an object. For example:

Example 3. 这张桌子新，那张桌子旧。(This desk is new, but that one is old.)

The predicates of the two sub-sentences can be synonyms, antonyms, or positive and negative forms. Although contrast sentences have particular forms, they describe similarities or differences between objects, which satisfies our definition.

2.2 The Identification Task

The task of CCS identification aims to determine

1. whether a specific Chinese sentence in the text is comparative or not, and
2. if so, which subcategory it belongs to.

Formally, for the set of sentences S and the set of categories C , the task is to find a function $f : S \rightarrow C$ that can assign a class label for each sentence. When performing task a, the label set C is {Non-comparative, Comparative}; and when performing task b, it can be {Non-comparative, equivalent, gradable differential, non-gradable differential}, or even {Non-comparative, exactly same, similar, greater, less, non-gradable differential}, etc. In this paper, we focus on the first sub-task, i.e. distinguishing comparative sentences from non-comparative ones, leaving task the second sub-task for further study.

3 Proposed Approach

The CCS identification problem defined in the previous section can be formalized as a classification problem. We apply three widely-used classifiers to this task, including SVM, Naïve Bayes and Decision Tree classifier. Given these classifiers, which features are used to represent the sentences is the key issue of identification. Linguistic researches show that many CCSs contain some particular words or satisfy some grammar forms, indicating that keywords and structural patterns can be used as features.

3.1 Identification through Term-Based Features

In Chinese, several kinds of words can indicate comparative sentences:

- **Relative adverbs**, such as “更”(more), “最” (most), “顶” (top), are widely used in comparisons. Their function is similar to comparative suffix “-er” or superlative suffix “-est” in English.
- **Pivots**, which introduce the standards of comparisons in grammaticalized CCSs, are important marks of comparative structures. For example, the word “比” (than) is a pivot in “我比他高10cm” (I am taller than him by 10cm), and so is “有” (as) in “地上的土有个铜钱厚” (The dust on the ground is as thick as a coin).
- **some verbs and adjectives**, such as “超过” (exceed), “一样” (same), “差不多” (more or less), etc, are also signs of comparisons.

An important fact is that many indicative words are not very strong evidences, because they have other meanings and are also used in non-comparative sentences. Fortunately, their meanings and functions are usually interrelated with POS tags in particular sentence. For example, when used as conjunction, the word “和” means “and” and is not a indication. However, when used as preposition, it means “as” and is a widely used pivot. So we combine the spelling with the POS tag of each word as a monolithic term feature. Note that there are no comparative or superlative form of adjectives and adverbs in Chinese, so we can not identify CCSs through POS tags alone.

In order obtain the list of indicative terms, we collected some keywords from the patterns given by Che[14] and a corpus we have labeled, as well as other words provided through our knowledge. We represent each sentence by a binary vector of these term features, then train the classifiers and finally classify a sentence into either “Comparative” or “Non-Comparative”. We also tried use all possible word/POS pairs in the vocabulary and tag set as features, expecting the classifiers could learn which terms are important indications.

3.2 Identification through Sequential-Based Features

Though some CCSs contain keywords which rarely appear in non-comparative sentences, however, some CCSs share similar vocabulary with non-comparative ones. Thus it is rather difficult to distinguish them through term-based features.

As mentioned in previous sections, there are some common forms or patterns in many CCSs. If we can extract these patterns, we can determine a sentence’s category according to which patterns it matches. Since we want to distinguish CCSs from non-comparative sentences, we must find the characteristic patterns of each class. By assigning a class label to each sequence, the *Class Sequential Rule* mining task[1] aims to find patterns having high correlation with each class.

Sequence and Class Sequential Rule

Formally, let $I = \{i_1, i_2, \dots, i_n\}$ be a set of *items*. An *itemset* X is a non-empty set of items. A *sequence* s is an ordered list of itemsets, denoted as $\langle a_1 a_2 \dots a_r \rangle$, where a_i is an itemset, also called an *element* of sequence. The *length* of a sequence is the number of items in the sequence. A sequence $s_1 = \langle a_1 a_2 \dots a_r \rangle$ contains another sequence $s_2 = \langle b_1 b_2 \dots b_m \rangle$, if there exist integers $1 \leq j_1 < j_2 < \dots < j_m \leq r$, such that $b_1 \subseteq a_{j_1}, b_2 \subseteq a_{j_2}, \dots, b_m \subseteq a_{j_m}$.

In class sequential rule mining problem, the input data D is a set of pairs, $D = \{(s_1, y_1), (s_2, y_2), \dots, (s_n, y_n)\}$, where s_i is a sequence, and y_i is the class label. A *Class Sequential Rule* (CSR) is an implication $s \rightarrow y$, where s is a sequential pattern, y is a class label.

A data instance (s_i, y_i) is said to *cover* the CSR if s_i contains s . A data instance (s_i, y_i) is said to *satisfy* the CSR if s_i contains s and $y_i = y$. The *support* of the rule is the fraction of total instances in D that satisfies the rule. The *confidence* of the rule is the proportion of instances in D that covers the rule also satisfies the rule. Note that the confidence is a measurement of correlation between a pattern and the class.

Given a labeled sequence data set D , a minimum support threshold and a minimum confidence threshold, the CSR mining algorithms find all CSRs that satisfy these constraints. Any traditional sequential pattern mining algorithms such as GSP and PrefixSpan can be easily modified to mining CSR, just counting support instances of each class separately and dropping any rules whose confidence is lower than the threshold. The first component of each minded rule, i.e. the sequential pattern, is collected for further identification procedures.

Mining Indicative Patterns from Text

In order to apply CSR mining algorithm to the sentences corpus, we first transform the corpus into a set of sequences. The simplest strategy is to change each word identity of a sentence into an element of the corresponding sequence. However, this approach will lead to a very sparse dataset in which the sequences share few elements in common. We use Jindal's strategy, in which the spelling and POS tag of a keyword are integrated as one item, while only the POS tags of other words are used[1]. We also tried another strategy that all words and their POS tags are used as two items of elements; however, it does not perform well according to our experimental study.

The length of the sequence is another important factor. When the unit is too short, few meaningful features will be involved; however, when it is too long, redundant elements will bring interference. Jindal used a window of 7 words to generate sequences[1]. However, the comparative sentences in Chinese are not as compact as those in English. For example:

Example 4. 核心/n 方面/n 与/p 目前/t 65/m nm/q 双核/n 炫龙/n 64/m 位/q 处理器/n 完全/d 相同/a o(The kernel is exactly same as current 65 nm Turion64 X2 processor.)

This sentence has a long standard (目前65nm双核炫龙64位处理器, current 65nm Turion64 X2 processor), which leads to a large gap between keywords “与”(as) and “相同”(same). As contrast, the similar phrase “same as” in English has no gap between the two words, and the phrase “as ... as” has only one word in the gap. This difference makes it rather difficult to find a proper window radius for Chinese comparative sentences.

According to our observations, most grammaticalized comparative structures, or their main compositions, occur in a sub-sentence. So we transform each sub-sentence into a sequence. Though this approach has negative effect on contrast sentences, it is effective in the mass, because most of common CCSs are grammaticalized.

After generating the sequence dataset, we apply a modified PrefixSpan algorithm to extract the rules, gather the pattern of each rule, and then obtain a pattern set P .

Classification on Patterns

After the pattern set P is extracted, each sequence is matched against every pattern in P . The matching result is a binary value that indicates whether the sequence contains the pattern or not. All these results form a vector as the

representation for the sequence. Formally, given pattern set $P = \{p_1, p_2, \dots, p_m\}$, a sequence s is represented as $\langle f_1, f_2, \dots, f_m \rangle$, where

$$f_i = \begin{cases} 1 & \text{if } s \text{ contains } p_i \\ 0 & \text{otherwise} \end{cases}, 1 \leq i \leq m. \quad (1)$$

The classifiers are trained on these feature vectors, and then applied to classify the sequences generated from test set of sentences.

When a sentence generates only one sequence, the class of sentence is exactly same as the predicated label of sequence. When a sentence generates more than one sequence, two integration strategies can be applied. The first one is to integrate matching results of all sequences into one feature vector to represents the sentence. However, it does not perform well in our experimental study. As we know, we judge a sentence as comparative if it contains a comparative structure or component. So, we use the second approach to classify the sentence as follows,

$$C(sent) = \begin{cases} \text{'Comparative'} & \text{if } \exists seq (seq \in S \wedge C(seq) = \text{'Comparative'}) \\ \text{'Non-Comparative'} & \text{otherwise} \end{cases}. \quad (2)$$

where S is the set of sequences generated by the sentence.

4 Experimental Results

4.1 Dataset

For the novelty of the CCS identification task, there is no public benchmark dataset yet. Thus, we collected some notebook reviews from ZOL product forum¹ and then manually labeled each sentence in the reviews. The overview of dataset is shown in Table 2.

Table 2. Dataset Overview

	Non-Comparative	Comparative
Number	1297	458

We use 5-fold cross validation since the dataset is not very large, and report the average precision, recall as well as F-measure for evaluation.

4.2 Identification Results

In this experiment, we compare the performances of three classifiers using different features. We use SVM^{light} with the linear kernel², the Naïve Bayes of Multi-variate Bernoulli Model, and the C4.5 decision tree toolkit implemented

¹ <http://group.zol.com.cn>

² <http://svmlight.joachims.org/>

Table 3. Results of Different Approaches

	Precision	Recall	F-measure
Baseline	96.7%/-	64.2%/-	0.772/-
SVM/WP	98.7%/+2.1%	64.7%/+0.7%	0.781/+1.2%
NB/WP	78.6%-18.7%	40.7%/-36.6%	0.535/-31.5%
C4.5/WP	92.5%/-4.3%	73.4%/+14.3%	0.817/+5.8%
SVM/KWP	95.7%/-1.0%	69.9%/+8.9%	0.806/+4.4%
NB/KWP	94.7%/-2.1%	72.8%/+13.4%	0.822/+6.5%
C4.5/KWP	95.8%/-0.1%	71.3%/+11.1%	0.815/+5.6%
SVM/CSR	91.4%/-5.5%	79.6%/+23.9%	0.850/+10.1%
NB/CSR	92.3%/-4.6%	71.5%/+11.3%	0.804/+4.1%
C4.5/CSR	90.5%/-6.4%	79.0%/+23.1%	0.843/+9.2%

by J. R. Quinlan³. The features include words plus their POS tags (denoted as WP), manual selected keywords plus their POS tags (denoted as KWP) and patterns obtained by Class Sequential Rule mining (denoted as CSR). The support threshold used in CSR mining is 10% of the minimum frequency of items occurred in the rule, and confidence threshold is 0.65. The baseline system is traditional word-based SVM classifier. The performances are shown in Table 3. Each cell reports corresponding performance value and the improvement against baseline.

Generally speaking, all the precision values are quite good, while the recall values are low. When using manually selected term-based features (i.e. KWP), the term-based classifiers improved the recall value. The results show the decision tree classifier is influenced little by this feature selection (C4.5/WP vs. C4.5/KWP). However, the Naïve Bayes classifier suffers great damage from redundant terms (NB/WP vs. NB/KWP), which is also mentioned in other text classification tasks[15]. The SVM and decision tree using sequential patterns as features obtain much higher recall and F-measure than the other systems, though losing a little precision (SVM/CSR and C4.5/CSR). The Naïve Bayes classifier seems not benefit from these pattern features as much as KWP.

4.3 Effects of Sequence Generating Strategies

As discussed in previous section, sequence generating strategies will affect pattern extraction and final classification. In this experiment, we compare 3 different strategies. The first one is to change the whole sentence into a sequence (denoted as WS); the second one is to change words around a keyword into a sequence (denoted as C_n , where n is the radius of window); and the last one is to transform each sub-sentence into a sequences (denoted as SS). For each strategy, patterns are extracted from the generated sequence dataset and a SVM classifier are training and applied using these pattern features. We choose SVM because

³ <http://www.rulequest.com/Personal/>

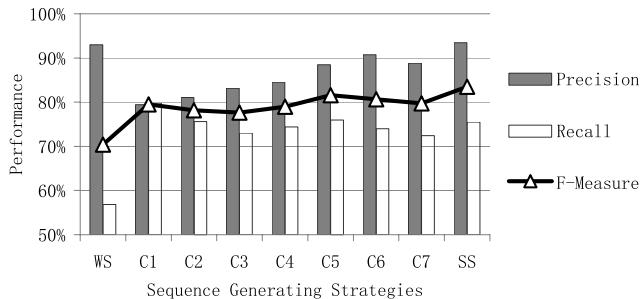


Fig. 1. Evaluation of Different Sequence Generating Strategies

its performance is better than other classifier as shown in previous section. The performances of SVM classifiers are shown in Fig. 1.

According to the results, the window strategy is better than WS strategy. The best radius is 5, which is bigger than that Jindal used in English (3 actually). Meanwhile, the sub-sentence strategy outperforms both WS and window strategies, though the disparity is not quite large.

4.4 Effects of Patterns' Length

We already know that sequences' length can affect classification performance. An interesting question is whether the length of patterns has any influence? Or how long is enough for a pattern to indicate a comparative structure?

In this experiment, we compare the performances of SVM classifier using patterns of different length. Given a maximal length, any patterns longer than it are excluded from the pattern set, and not considered by the classifier. The results are shown in Fig. 2.

When the maximal length is 1, the pattern mining procedure actually degrades to a kind of keywords selection, and the classifier gets highest precision but lowest recall. The classifier performs best at the maximal pattern length of 2 or 3. F-measure decreases while pattern length grows higher because the recall

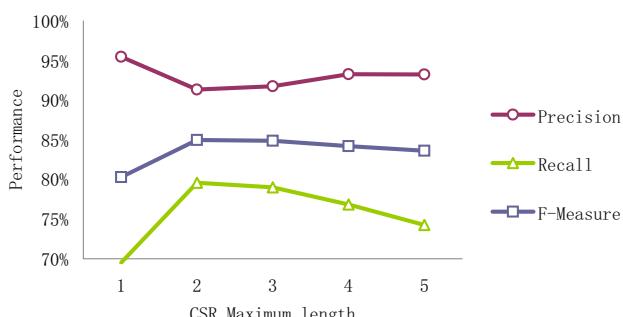


Fig. 2. Effective of Constraint on Patterns' Maximum Length

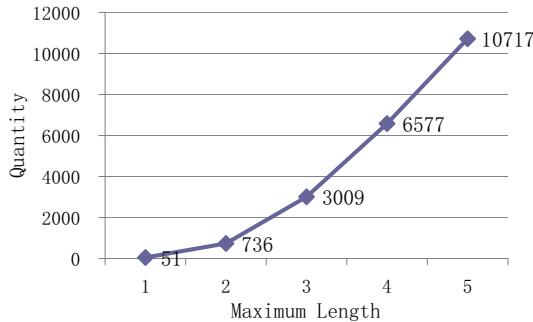


Fig. 3. Avg. Quantity of CSRs vs. Maximum Length

value drops fast. Though the difference of performance is not quite significant, the quantity of rules explodes when pattern length increases, which is shown in figure 3. So, limiting maximal pattern length to 2 or 3 is a good choice.

4.5 Case Study

Case 1. (Comparative)拿笔记本来玩游戏的人没有拿笔记本来办公的人多。(People who use notebook to play games are not as many as those who use them to work.)

Term-based classifiers predicate it as non-comparative, because the indicative word “没有” (are not as) is quite common in non-comparative sentences. Pattern-based classifier identified it correctly, according to these patterns:

- | | |
|-------------------------------|-------------------------------|
| 1. */u 没有/d (Non-Comparative) | 5. 没有/d */a (Comparative) |
| 2. */v 没有/d (Non-Comparative) | 6. 没有/d */v (Non-Comparative) |
| 3. */n 没有/d (Non-Comparative) | 7. 没有/d */n (Non-Comparative) |
| 4. 没有/d (Non-Comparative) | |

In fact, the decisive pattern is 5, which is a simplified form of linguistic pattern “没有 $Y R$ ”, where X is the comparee, Y is the standard, and R is the result.

Case 2. (Comparative) “属性”中的数值大于剪切下来的图片。(The value in “attribute” is bigger than that of the picture cut down.)

The linguistic pattern of this sentence is “ $X R$ 于 Y ”. Again, term-based SVMs give a wrong label because “于” (than) is also common in non-comparative corpus. Pattern-based SVM works correctly, according to following patterns:

- | | |
|------------------------------|------------------------------|
| 1. */a 于/p (Comparative) | 3. 于/p */a (Non-Comparative) |
| 2. */v 于/p (Non-Comparative) | 4. 于/p */n (Non-Comparative) |

We can see pattern 1 is similar to the linguistic pattern.

Case 3. (Non-comparative) 此双核解决方案有利于提高系统性能。(This duo-core solution is favorable for improving system performance.)

Pattern-based SVM makes a mistake on this sentence. Its evidences are

- | | |
|------------------------------|------------------------------|
| 1. */m 于/p (Non-Comparative) | 4. */v 于/p (Non-Comparative) |
| 2. */r 于/p (Non-Comparative) | 5. 于/p */v (Non-Comparative) |
| 3. */a 于/p (Comparative) | 6. 于/p */n (Non-Comparative) |

This sentence do satisfy pattern “ $X R$ 于 Y ”, however, the adjective “有利” (favorable) is not used to describe properties of objects, so it does not mean comparison. Other examples include “便于” (convenient to), “安于” (be resting in), etc. A possible solution is put these adjectives into the set of keywords, but it is rather difficult to collect all of them. This problem needs more study in the future.

5 Conclusions

This paper defines the task of Chinese comparative sentences identification, and proposed an approach to use classifiers to classify a Chinese sentence into either “comparative” or not. Various linguistic and statistical features have been explored, including keywords and sequential patterns. The experiment shows the effectiveness of our approach. More specific, term-based SVM and decision tree classifier as well as all pattern-based classifiers defeat the baseline system based on bag-of-words model; and pattern-based SVM outperform the others in recall and F-measure, though losing a little precision. Two important factors that affect pattern-based SVM, including sequence generating strategy and maximal length of patterns, are also been studied. When using short patterns mined from sequence dataset generated with sub-sentence strategy, the pattern-based SVM classifier perform best.

Meanwhile, all classifiers are far away from perfect. Chinese have much more features, including collocations, semantic constraints, etc. How to enhance the precision and recall by adding new features to the classifier needs more study. In addition, the pattern extraction can also be improved. For example, we may decrease interferes of complex attributes by extracting the trunks of sentences. A large dataset of Chinese comparative sentences is also needed to be constructed.

Acknowledgements. This work was supported by the National Science Foundation of China (No.60703064), the Research Fund for the Doctoral Program of Higher Education of China (No.20070001059) and the National High Technology Research and Development Program of China (No.2008AA01Z421).

References

1. Jindal, N., Liu, B.: Identifying comparative sentences in text documents. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 244–251. ACM Press, New York (2006)
2. Jindal, N., Liu, B.: Mining comparative sentences and relations. In: Proceedings of the 21st National Conference on Artificial Intelligence, AAAI 2006 (2006)

3. Zhai, C., Velivelli, A., Yu, B.: A cross-collection mixture model for comparative text mining. In: KDD 2004: Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 743–748. ACM Press, New York (2004)
4. Sun, J.T., Wang, X., Shen, D., Zeng, H.J., Chen, Z.: Cws: a comparative web search system. In: WWW 2006: Proceedings of the 15th international conference on World Wide Web, pp. 467–476. ACM Press, New York (2006)
5. Luo, G., Tang, C., Tian, Y.L.: Answering relationship queries on the web. In: WWW 2007: Proceedings of the 16th international conference on World Wide Web, pp. 561–570. ACM Press, New York (2007)
6. Feldman, R., Fresko, M., Goldenberg, J., Netzer, O., Ungar, L.: Extracting product comparisons from discussion boards. In: Proceedings of the 7th IEEE International Conference on Data Mining, pp. 469–474 (2007)
7. Ma, J.: Ma Shi Wen Tong. Commercial Press, Shanghai (1898)
8. Lü, S.: Zhongguo Wenfa Yaolüe. Commercial Press, Shanghai (1942)
9. Lü, S.: Xiandai Hanyu Babai Ci. Commercial Press, Shanghai (1980)
10. Liu, Y.: Xiandai Hanyu Bijiao Fanchou de Yuyi Renzhi Jichu. Academia Press, Shanghai (2004)
11. Liu, D.: Framework and thinking of research in differential comparative sentences. In: Dai, Q. (ed.) Modern Linguistic Theory and Research in Languages of Chinese Minorities. Ethnic Publishing House, Beijing (2003)
12. Lerner, J.-Y., Pinkal, M.: Comparatives and nested quantifications. In: Semantics: Critical Concepts in Linguistics, vol. V, Operators and Sentence Types, pp. 70–87 (2004)
13. Stassen, L.: Comparison and Universal Grammar. Basil Blackwell, Malden (1985)
14. Che, J.: A brief analysis of comparative sentences in modern chinese. Journal of Hubei Normal University (Philosophy and Social Sciences) (3) (2005)
15. Langley, P., Sage, S.: Induction of selective bayesian classifiers. In: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (1994)

Efficient Exhaustive Generation of Functional Programs Using Monte-Carlo Search with Iterative Deepening

Susumu Katayama

University of Miyazaki

1-1 W. Gakuenkibanadai, Miyazaki, Miyazaki 889-2155, Japan
skata@cs.miyazaki-u.ac.jp

Abstract. Genetic programming and inductive synthesis of functional programs are two major approaches to inductive functional programming. Recently, in addition to them, some researchers pursue efficient exhaustive program generation algorithms, partly for the purpose of providing a comparator and knowing how essential the ideas such as heuristics adopted by those major approaches are, partly expecting that approaches that exhaustively generate programs with the given type and pick up those which satisfy the given specification may do the task well. In exhaustive program generation, since the number of programs exponentially increases as the program size increases, the key to success is how to restrain the exponential bloat by suppressing semantically equivalent but syntactically different programs. In this paper we propose an algorithm applying random testing of program equivalences (or Monte-Carlo search for functional differences) to the search results of iterative deepening, by which we can totally remove redundancies caused by semantically equivalent programs. Our experimental results show that applying our algorithm to subexpressions during program generation remarkably reduces the computational costs when applied to rich primitive sets.

1 Introduction

Inductive functional programming is a machine learning field of generation of functional programs by generalization from ambiguous specifications such as input-output examples or constraints over programs. Due to the ambiguity in the way to generalize the specification, in inductive program synthesis it is often the case that the generated programs do not meet the user's intention.

Human programmers usually take the following steps:

invent the algorithm → check it by browsing → test it

among which inductive synthesis replaces the algorithm invention part and helps the testing part.

There exist two approaches to inductive functional programming: the generate-and-test approach such as genetic programming (GP) (e.g. [1],[2]) that first generates programs and then tests if they satisfy the specification, and the analytical

approach that is to some extent based on analysis of the I/O examples, such as the two step methods that first generate a non-recursive program implementing the computational traces from I/O examples and then fold it into a recursive program. (e.g. [3]) Recently, in the analytical approach an algorithm that extends the classical Summers' method is proposed, that searches the hypothesis space narrowed by the template that is obtained by generating the least general generalizations of the input set and the output set [4], replacing the two step methods for its efficiency.

Inductive functional programming by GP can be applied to various problem frameworks. On the other hand, because GP algorithms usually search rather a big hypothesis space, without human labor they tend to consume more computation time. The recent Summers-like method synthesizes programs quickly, though they are limited to synthesis from I/O examples that satisfies some conditions.

We have been working on efficient implementation of exhaustive program generation for given types.[5][6][7] Our main interest is to tell the baseline performance of non-heuristic search, and hopefully provide a new, usable method within the generate-and-test framework. Although our algorithm described in those papers successfully generates small programs without any prior knowledge except the type information, it have been having the following problems:

- it lacks in formalization, although it should efficiently generate infinite number of proofs based on Herbelin's LJT with regard to Curry-Howard isomorphism, i.e., correspondence between proofs and programs;
- it generates lots of mathematically equivalent functions implemented in different ways, most of which are actually identity or constant functions, and which cause inefficiency and human unreadability of the results.

This research continues our policy, and proposes an algorithm that completely removes the redundancy caused by semantically equivalent programs. Instead of removing functions that are theoretically known to be equivalent as suggested in [7], from the generated lazy infinite stream our proposed algorithm completely removes the redundancy caused by semantically equivalent programs, by combining Monte-Carlo search with iterative deepening. By extending the literature on random testing[8], our algorithm can be applied polymorphically — it can be applied to any function, provided that its parameter values can be generated randomly¹ and that an equivalence relation can explicitly be defined between its return values.

Experimental results show that our algorithm effectively restrain exponential bloat when applied to a rich primitive set. This means that our algorithm is useful in practical cases where we want to generate expressions consisted of standard library functions rather than reinventing well-known toy functions from scratch.

¹ Note that [8] shows even higher-order functions can be generated randomly.

2 Exhaustive Program Generation

In this section we review our systematic search algorithm with regard to automatic theorem proving.

Curry-Howard isomorphism is the observation that logic formulas and their proofs have the same structure as types and functions. For example, just in the same way as deriving a proof for B from proofs for $A \rightarrow B$ and A , we can obtain the value fx of type B from a function f of type $A \rightarrow B$ and a value x of type A , where $A \rightarrow B$ denotes the function type taking A as the argument and returning B . Also, just in the same way as deriving a proof for $A \rightarrow B$ from a proof of B under the assumption of A , we can obtain a function $\lambda x.E$ of type $A \rightarrow B$ by constructing a value E of type B using a variable x of type A as its argument. So far we explained the isomorphism between the propositional logic and simply typed lambda calculus, but there are also isomorphisms between richer logic and lambda calculus.

Under Curry-Howard isomorphism, an automated theorem prover corresponds to an algorithm generating a functional program of the given type. Taking advantage of this fact, some theorem provers such as Coq and Agda have the aspect of deductive programming systems that generate a function satisfying the specification given as a type, though it is hard to totally automate deductive programming under such an expressive type system and they depend on human guidance to some extent.

On the other hand, our systematic exhaustive search algorithm corresponds to generating infinite number of proofs as an infinite stream, under second-order intuitionistic propositional logic, and picking up those which satisfy the given specification. Extending automatic provers to generate infinite number of proofs instead of only one does not dramatically change the algorithm a lot, except that

- in order to consider combinations of infinite possibilities, we have to interleave them somehow or other, using e.g. Spivey’s monad for breadth-first search[9] or monad for depth-bound search[10];
- in order to generate all the proofs, even if $A \leftrightarrow B$ we may not replace A with B , because there can be many proofs for $B \rightarrow A$, which means G4ip[11], a.k.a. Dyckhoff’s LJT [12], cannot be applied, at least straightforwardly;
- the algorithm must be as efficient as possible, using e.g. memoization, though we do not exhaust here all of those that were used.

Our algorithm uses the inference rules of the cut-free LJT with the implication and universal quantification in the Curry style, which is behind the systematic exhaustive search algorithm. More exactly, we prohibit higher-rank polymorphism and use unification for efficiency reasons.

An inductive data type can be made available by providing its constructors and its induction function as assumptions at the left hand side of the turnstile. For example, generating programs of type $\forall a. [[a]] \rightarrow a$ corresponds to proving the following sequent:

$$\boxed{\quad} :: \forall a.[a], (\:) :: \forall a.a \rightarrow [a] \rightarrow [a], foldr :: \forall ab.b \rightarrow (a \rightarrow b \rightarrow b) \rightarrow [a] \rightarrow b ; \vdash \forall a. [[a]]$$

where $[X]$ denotes the type for lists of X 's.

One drawback of this approach is that the induction function $foldr$ introduces an existential type when instantiating the type variable a using the $\forall L$ rule, which makes the algorithm inefficient. For this reason or other, [7] prohibited functional types appearing within container types, as in $[a \rightarrow b]$ or $(a \rightarrow b, c)$, but still suffers from vast search space.

In this paper, we regard lists in the same light as their isomorphic types $\forall b.(a \rightarrow b \rightarrow b) \rightarrow b \rightarrow b$, and use the following rule:

$$\frac{\Gamma, xs :: [A]; \vdash op :: A \rightarrow B \rightarrow B \quad \Gamma, xs :: [A]; \vdash x :: B}{\Gamma, xs :: [A]; \vdash foldr\ op\ x\ xs :: B} \text{ foldr}$$

The point is that the type A can be obtained by pattern matching, and thus does not introduce a new existential type.

The same idea applies to other inductive types. We do not deal with coinductive types in this paper.

3 Thinning Up a Stream of Program Sets

3.1 Monte-Carlo Filtration of Program Sets

We use Monte-Carlo search to see if two functions are different, by searching for a point where their values are different. More specifically, we define an equivalence relation based on a random point set as in Definition 1.

Definition 1 (Equivalence by a random point set). *For a random point set $r = \{r_1..r_n\}$, we define*

$$f \sim_r g \Leftrightarrow f \sim_{\{r_1..r_n\}} g \stackrel{\text{def}}{\Leftrightarrow} \forall i \in \{1..n\}. f(r_i) = g(r_i) .$$

For any point set r , \sim_r becomes an equivalence relation. Moreover, $\sim_{\{r_1..r_n\}}$ is a refinement of $\sim_{\{r_1..r_{n-1}\}}$, i.e. $f \sim_{\{r_1..r_n\}} g \Rightarrow f \sim_{\{r_1..r_{n-1}\}} g$. If the random sequence r is exhaustive, e.g., if r is uniform and its domain set is countable, $\sim_{\{r_1..r_\infty\}}$ should equal to the intentional equality. (Note that any set of valid programs or Turing machines is always countable.)

Our algorithm uses a lazy infinite stream to correctly yield a complete set of representatives in the limit. The rough idea is:

- by random testing we can often prove the differences between functions, but can never prove the equivalences;
- therefore, we just abandon functions that may be equivalent to other functions, except one representative, i.e., we obtain (for each depth bound) a complete set of equivalence class representatives by the equivalence based on some random number/function set;
- because we use iterative deepening for generating programs, by using different set of random numbers at each depth limit, “innocent” functions that are unfortunately abandoned but are in fact different from others will eventually be recovered at some depth, provided that we use an exhaustive random number generator.

Table 1. Example of how filtration after program generation works

This is an example of obtaining the representatives for each depth bound $\{[f_1], \{f_1\}, \{f_1, f_2\}, \dots\}$ from the set of functions for each depth bound $\{[f_1], \{f_1, f'_1\}, \{f_1, f'_1, f''_1, f_2\}, \dots\}$ and random number sequences for each depth bound $\{[1, 4], [1, 4, 2], [1, 4, 2, 3], \dots\}$, where $f_1 = f'_1 = f''_1$ and f_1 and f_2 are defined as follows:

x	1	2	3	4	5	6	...
$f_1(x)$	1	2	3	4	5	6	...
$f_2(x)$	1	2	1	4	5	6	...

but we do not know these facts in advance.

depth bound	1	2
(multi)sets of functions	$\{f_1\}, (*1)$	$\{f_1, f'_1\},$ $[1, 4, 2],$
random numbers	$[1, 4],$	
map functions	$\{[f_1(1) = 1, f_1(4) = 4]\},$	$\{ [f_1(1) = 1, f_1(4) = 4, f_1(2) = 2],$ $[f'_1(1) = 1, f'_1(4) = 4, f'_1(2) = 2] \},$
equivalence classes	$\{\{f_1\}\}$	$\{\{f_1, f'_1\}\}$
representatives	$\{f_1\},$	$\{f_1\},$
differentiate if desired (*2)	$\{f_1\},$	$\{\},$
	3	...
	$\{f_1, f'_1, f''_1, f_2\},$ $[1, 4, 2, 3],$...
cont'd	$\{ [f_1(1) = 1, f_1(4) = 4, f_1(2) = 2, f_1(3) = 3],$ $[f'_1(1) = 1, f'_1(4) = 4, f'_1(2) = 2, f'_1(3) = 3],$ $[f''_1(1) = 1, f''_1(4) = 4, f''_1(2) = 2, f''_1(3) = 3],$ $[f_2(1) = 1, f_2(4) = 4, f_2(2) = 2, f_2(3) = 1] \},$...
	$\{\{f_1, f'_1, f''_1\}, \{f_2\}\}$...
	$\{f_1, f_2\},$...
	$\{f_2\},$...

(*1) Although we use the set brackets, i.e. $\{ \text{ and } \}$, we assume these can be multisets, because we do not have a universal method to prove two functions are equivalent.

(*2) *differentiate* $[S_1, S_2, S_3, \dots] = [S_1, S_2 \setminus S_1, S_3 \setminus S_2, \dots]$. Also see Theorem 1.

Actually, just following the above policy breaks the implicit assumption of iterative deepening that the search space of a deeper iteration is a superset of that of a shallower one. However, if we include all the random sample points used in earlier iterations, i.e., if we append new random sample points instead of totally replacing the point set in each iteration, the equivalence relation refines as the point set increases, and thus we can assure that representatives that once appear will always appear at each of the deeper levels, as stated in the following Theorem 1. Thanks to this theorem, we can differentiate the filtration results by using the syntactical difference at the end if necessary. (Table 1)

Theorem 1. Let $S(s)$ denote the set we obtain by removing the structure of list s . (For example, $S([3, 6, 2, 2, 5]) = \{2, 3, 5, 6\}$) There exists an $O(mn \log n)$ -time algorithm A that takes a list of length n and a random sequence of length m and returns a list, such that $S(A(xs, \{r_1 \dots r_m\}))$ is a complete set of representatives of the quotient set $S(xs)/_{\sim \{r_1 \dots r_m\}}$, and $S(A(xs, \{r_1 \dots r_{m-1}\})) \subset S(A(xs \# ys, \{r_1 \dots r_m\}))$ where $\#$ denotes list concatenation.²

Proof. (sketch) $A(xs, rs)$ is the algorithm that sorts xs by the preorder \leq_{rs} and the equivalence \sim_{rs} , and selects the first element from each of the resulting

² We follow the conventions in the functional programming literature and use plural forms for list variables, e.g. xs , ys , etc. rather than x , y , etc. respectively.

equivalence classes, where \leq_{rs} is the lexicographical order of the function values at the random points, defined recursively as

$$f \leq_{\phi} g , \\ f \leq_{\{r_1..r_m\}} g \Leftrightarrow f \leq_{\{r_1..r_{m-1}\}} g \wedge (f \sim_{\{r_1..r_{m-1}\}} g \rightarrow f(r_m) \leq g(r_m)) .$$

The sorting algorithm used here must be stable, i.e. it must not change the order between equivalent elements. (Many well-known sorting algorithms such as mergesort and quicksort satisfy this requirement.)

It is trivial that the above algorithm A requires $O(mn \log n)$ -time and the result of A is a complete set of representatives. We can prove the inclusion relation between the complete set of representatives by using the next Lemma 1 with the total order of “appearing earlier in xs ”. \square

Lemma 1. *Let \sim and \approx denote two equivalence relations on U , where \approx refines \sim . Let \leq denotes a total order on U . For all finite $S, T \subset U$ such that $\forall s \in S. \forall t \in T. s \leq t$, define complete sets of representatives of \sim and \approx as*

$$Q = \left\{ \min_{\leq} c \mid c \in S/\sim \right\} \\ R = \left\{ \min_{\leq} c \mid c \in (S \cup T)/\approx \right\}$$

then,

$$Q \subset R$$

where $\min_{\leq} P$ is the minimal element of P by \leq , i.e., $\min_{\leq} P = x$ s.t. $x \in P \wedge \forall y \in P. x \leq y$.

This lemma means that if we always pick up the elements that have some property most as the representatives of two equivalence relations where one refines the other and from them, the resulting complete set of representatives of the former includes that of the latter.

3.2 Filtration during Program Generation

By applying the above method to an infinite set of functional programs, we can provably obtain an infinite stream of complete set of equivalence class representatives. Our further interest now is to apply the filter to subprograms *during* program generation rather than *after* program generation, in the hope of some leverage in saving heap consumption.

This is achieved by applying this filter to each item on the memoization table that binds types to sets of expressions. However, there is an implementation issue with regard to existential types: in order to provide polymorphism, our old method memoizes the function that binds types which may include existential types to the set of all expressions whose type unifies with the given type, and generates subexpressions recursively; on the other hand, in order to apply our Monte-Carlo filter, the element functions in the infinite set to be filtered must have the same type. For example, the memo table binds query $[a]$ to the set of

expressions that may include expressions with type `[Char]` and those with type `[Int]` — this obviously causes problems when thinning up the set.

In order to cope with this situation, we use two different memoization tables: one is dedicated to enumerate possible substitutions for each of the existential types, and the other holds the (Monte-Carlo filtered) expressions that have the same type as the query type. In order to obtain a stream of expressions of a given type, our algorithm firstly looks up the first table to obtain possible substitutions, replaces the existential types of the query type, and then looks up the second table.³

Another problem is that the number of random samples used per one expression increases as we go deeper iteration, fueling the fire rather than restraining the exponential bloat. This is problematic especially when applying our filter to subexpressions generated during the whole program generation in order to leverage the efficiency. For this reason, we define two exhaustive filters:

- Filter 1**, which is efficient but permits some duplicates, that uses a different set of few random numbers for each data type at each iteration, and amends the fewness by accumulating the resulting stream (Table 2), and
- Filter 2**, which is inefficient but prohibits any duplicate, that add a random number as the search goes deeper (Table 1).⁴

By applying Filter 1 during program generation and then applying Filter 2 to the final result, we obtain exhaustive but not redundant results. Moreover, this process is more efficient than only applying Filter 2 during program generation, because the amount of data is already thinned up by the Filter 1 when Filter 2 is applied.

The idea behind Filter 1 instead of Filter 2 is using the union of the set of representatives under criterion (random point) r_1 , that of representatives under criterion r_2 , ... instead of using the set of representatives under their direct product (r_1, r_2, \dots) . After applying whichever filter, programs returning different values at random point r_n will survive for all n .

Remark 1. In order to apply Monte-Carlo search, there must be an algorithm for generating random number sequence for the domain type. Fortunately, this can be achieved easily in most types including functional ones, using the polymorphic random testing library QuickCheck[8].⁵

4 Experimental Results

We applied our algorithm to MagicHaskeller[13], our systematic exhaustive search library for Haskell.

³ These steps can be optimized by holding pointers to the entries in the second table, along with substitutions at each entry of the first table.

⁴ Note that the two filters only differ in the sets of random numbers.

⁵ In software engineering, Monte-Carlo search for programming errors is called random testing.

Table 2. Example of how filtration during generation works

The sets of functions to be filtered are $\{\{f_1, f_2\}, \{f_1, f_2, g_1\}, \{f_1, f_2, g_1, h_1, h_2\}, \dots\}$, where f_1 and f_2 are obtained from the first search, g_1 is obtained from the first deepening, and h_1 and h_2 are obtained from the second deepening.

Assume that $f_1 = g_1$ and $f_2 = h_1$, and that the values of these functions are defined as follows:

x	1	2	3	4	5	6	...
$f_1(x)$	1	2	3	4	5	6	...
$f_2(x)$	1	2	1	4	5	6	...
$g_1(x)$	1	2	3	4	5	6	...
$h_1(x)$	1	2	1	4	5	6	...
$h_2(x)$	1	1	1	1	1	1	...

but we do not know these values in advance.

depth bound	1	2	3	...
sets of functions	$\{f_1, f_2\}$,	$\{f_1, f_2, g_1\}$,	$\{f_1, f_2, g_1, h_1, h_2\}$,	...
random numbers	$[1, 2]$,	$[3]$,	$[2, 6]$,	...
map functions	$\{ [f_1(1) = 1, f_1(2) = 2], [f_2(1) = 1, f_2(2) = 2] \}$,	$\{ [f_1(3) = 3], [f_2(3) = 1], [g_1(3) = 3] \}$,	$\{ [f_1(2) = 2, f_1(6) = 6], [f_2(2) = 2, f_2(6) = 6], [g_1(2) = 2, g_1(6) = 6], [h_1(2) = 2, h_1(6) = 6], [h_2(2) = 1, h_2(6) = 1] \}$,	...
equivalence classes	$\{\{f_1, f_2\}\}$,	$\{\{f_1, g_1\}, \{f_2\}\}$	$\{\{f_1, g_1, f_2, h_1\}, \{h_2\}\}$,	...
representatives	$\{f_1\}$,	$\{f_1, f_2\}$,	$\{f_1, h_2\}$,	...
accumulate(*1)	$\{f_1\}$,	$\{f_1, f_2\}$,	$\{f_1, f_2, h_2\}$,	...

(*1) $\text{accumulate}[S_1, S_2, S_3, \dots] = [S_1, S_1 \cup S_2, S_1 \cup S_2 \cup S_3, \dots]$. The whole algorithm should work even when the union is that of multiset (because finally duplicates will be removed by Filter 2), but we can remove some duplicates at this step by syntactical equivalence.

4.1 Experiment Conditions

The current MagicHaskeller is released with several program generator algorithms and some options. The algorithms differ in the hypothesis space in the way described in Sect. 2, and options permit us to selectively enable each rule for theoretical equivalence check between expressions, which are based on some known optimization rules[7].

In this paper we stick to the newly introduced generator described in Sect. 2 and do not compare it with the old program generator with vast hypothesis space, due to the page limitation. Also, we disable the theoretical equivalence check.

We used Version 6.8.2 of Glasgow Haskell Compiler (GHC) on Linux 2.6.22-14, running Intel Pentium D 2.8GHz as a single processor. We modified MagicHaskeller to use depth-bound search with iterative deepening instead of breadth-first search, where the expressions are prioritized by the program size, that is measured by the number of function applications.

We experimentally discuss how many random sample points for each depth bound should be used for Filter 1 in Sect. 4.2, because there is a tight trade-off. The numbers used for each depth bound of Filter 2 are $[6, 7, 8, \dots]$. This decision is based on our intuition that two functions are often equivalent if their return values correspond at six points, our confidence in the algorithm's property that the functions will be recovered later if they are actually different, and our observation that the program generation is the bottleneck and the computation

cost of Filter 2 does not affect the total cost very much, if Filter 1 is applied during program generation.

Each execution timeouts 20 milliseconds after invocation. Programs that caused either timeout or an error during execution are removed for the iteration. (Stack space overflow is the most common one.) Since we are using a lazy language, such an error or timeout may occur during comparison between the return values of two programs; in such cases, both programs are removed, because it is difficult to tell which program is to be blamed.

We use different primitive sets named `mnat`, `mlist`, `mlistnat`, and `mrich`. `mnat` is a function set related to natural numbers, i.e. zero, successor, curried paramorphism, and addition, where `Int` is used to represent the type of natural numbers. `mlist` is a function set related to lists, i.e. nil, cons, and curried list paramorphism. `mlistnat` is the union of `mnat` and `mlist`. `mrich` is a rich function set mainly related to lists and booleans, i.e., `mlist` plus `not`, `map`, append, `filter`, `concat`, `concatMap`, `length`, `replicate`, `take`, `drop`, `takeWhile`, `dropWhile`, `lines`, `words`, `unlines`, `unwords`, `reverse`, `and`, `or`, `any`, `all`, and `zipWith` functions plus binary append (`(++)`), and (`&&`), or (`(||)`), extensional equality (`(==)`), and extensional inequality (`(/=)`) operators, where the equality and inequality operators are defined for some types.⁶

The experiments are made reproducible by using Version 0.8.4 of MagicHaskeller.

4.2 How Many Random Sample Points in Filter 1?

We use iterative deepening and the result of each iterations is filtered by Monte-Carlo method, in the way already discussed. So far so good, but there remains an issue of how many random sample points to be used at each iteration, and we do not have any conclusive theory on the strategy. In general, using more random points at each iteration means more difference to be proved and less expressions to be lost in the early iterations, but also means more execution time. One may think that more random points should be used for smaller depth bounds because small expressions are repeatedly reused everywhere. Others may think that less should be used for them because they do not directly affect the final result.

Figure 1 depicts the experimental results of the trade-off lines for some simple strategies. Since less time and more expressions are desirable provided that those expressions are proved to be different, strategies near the down-right corner should be good strategies. Table 3 defines the strategy families we tried. Strategies which appear in Table 3 but not in the legend of Fig. 1 are those which are known to perform very poorly and omitted to avoid clutter. For each strategy family, points and error bars for $n = 1 \dots 10$ are plotted, which represent the averages and standard deviations of 5 runs.

Graphs in Fig. 1 suggest the simple flat strategy is nearly the pareto optimal in both graphs; according to Fig. 1a the optimal parameter n is around 4 to 6, while according to Fig. 1b the optimal n is around 8 to 9, though in the latter case the graph has not converged yet.

⁶ Currently MagicHaskeller does not support type classes.

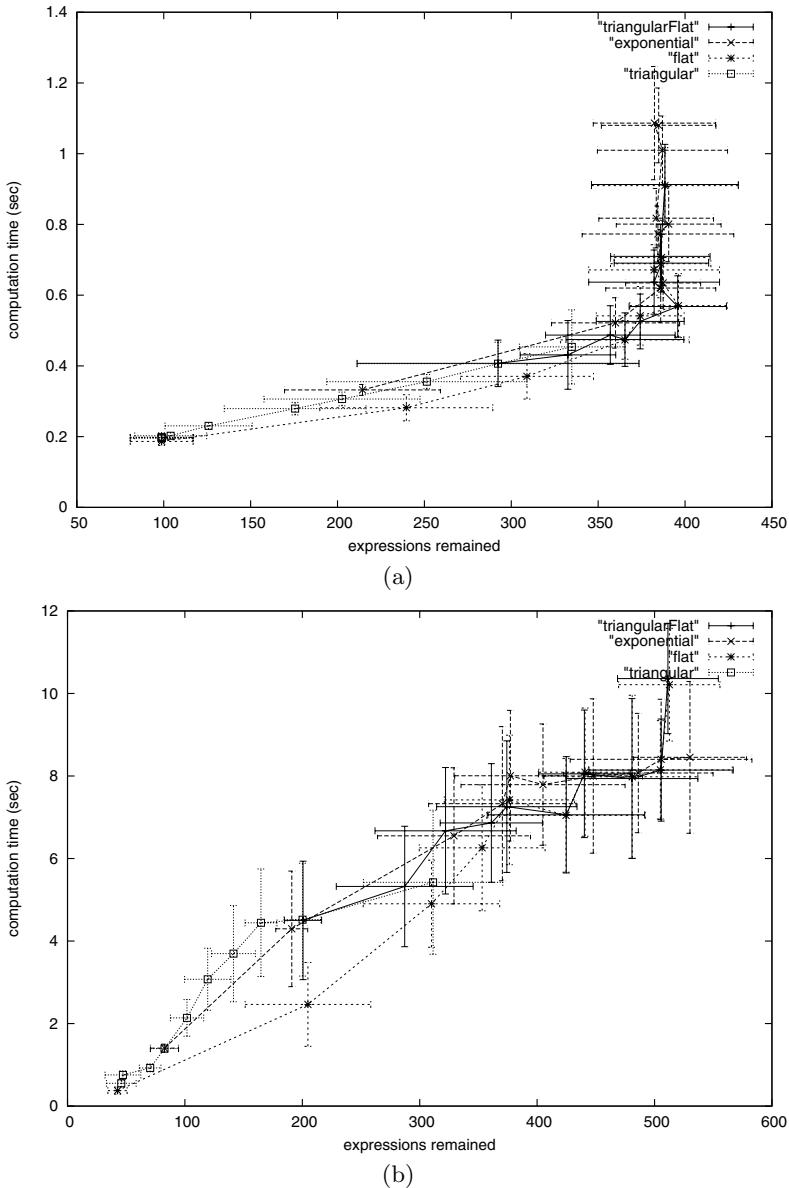


Fig. 1. Trade-off lines between the computation time and the number of remaining generated expressions within finite depth.

(a) number of functions with type $\text{String} \rightarrow \text{String}(*1)$ until the depth bound $t = 7$, generated from the `mrich` primitive set, (b) number of functions with type $\text{Int} \rightarrow \text{Int}$ until the depth bound $t = 10$, generated from the `mnat` primitive set.

(*1) In Haskell, `String` is an alias to `[Char]`.

Table 3. Strategy names

strategy family name	# of random points (d : depth bound)	example of $n = 2, t = 10$
delta	n if $d = 0$; 1 otherwise	[2,1,1,1,1,1,1,1,1,1]
exponential	$\lceil 2^{4-d/n} \rceil$	[16,12,9,6,5,3,3,2,2,1,1]
flat	n	[2,2,2,2,2,2,2,2,2,2]
trapezoidal	$[3.5 + n + (8 - 2n)d/t]$	[5,5,6,6,7,7,7,8,8,9,9]
triangular	$\max\{1, t - d - (n - 2)\}$	[10,9,8,7,6,5,4,3,2,1,1]
triangularFlat	$\max\{n, t - d\}$	[10,9,8,7,6,5,4,3,2,2,2]
steepTriangular	$\max\{1, n(t - d)\}$	[20,18,16,14,12,10,8,6,4,2,1]

Table 4. Time spent for generating all the possible expressions within 8 function applications. The “# of exprs” column shows the number of expressions generated at each depth. (NB: this does not mean “within each depth-bound”, i.e., this is the differentiated value.) “h/e” means out of memory.

primitive set	query type	not filtered		not filtered	
		time (sec)	# of exprs	time (sec)	# of exprs
mnat	$\text{Int} \rightarrow \text{Int}$	0.70	[2,2,6,22,78,324,1492,7726,42994]		
mllistnat	$\text{Int} \rightarrow \text{String} \rightarrow \text{String}$	0.58	[2,0,0,14,22,74,492,3030,14776]		
mrich	$\text{String} \rightarrow \text{String}$	h/e	[2,5,42,225,1755,12226,98008,771208,		
mrich	$(\text{Char} \rightarrow \text{Int}) \rightarrow \text{String} \rightarrow [\text{Int}]$	10.78	[1,2,12,63,415,2736,20393,155031,1240668]		
cont'ed	with Filter 2	with Filter 2	with Filters 1,2	with Filters 1,2	
	time (sec)	# of exprs	time (sec)	# of exprs	
	6.06	[2,2,3,4,9,20,44,98,286]	3.20	[2,2,3,4,12,14,53,119,251]	
	2.60	[2,0,0,3,0,3,13,10,107]	2.35	[2,0,0,0,0,1,7,11,60]	
	h/e	[2,1,3,13,19,101,304,1087,	55.11	[2,1,3,12,19,112,265,918,3793]	
	298.92	[1,0,0,3,1,3,21,22,120]	10.25	[1,0,0,3,0,0,22,13,128]	

4.3 Efficiency

Table 4 shows the time spent for computation and the number of generated programs without/with our Monte-Carlo filters. Based on the observation seen in the last section, we used the flat strategy with $n = 5$.

By comparing the number of expressions after applying only Filter 2 and that after applying Filters 1,2, one can tell that few different programs are lost by using Filter 1 except for the case of generating $\text{Int} \rightarrow \text{String} \rightarrow \text{String}$. Also, applying Filter 1 always reduces the computation time, especially when applied to results generated using the `mrich` primitive set.

5 Conclusions

We presented an algorithm for stripping mathematically equivalent functions from a prioritized infinite bag of functions, which is obtained as a search result. We implemented a function that takes such a prioritized bag as an argument and returns its complete set of representatives as a prioritized infinite set of functions. Our algorithm does not require that the equivalence between each function in the prioritized set is explicitly defined, but that its parameter values can be generated randomly, and that equivalence between return values is explicitly defined. Also, we applied the proposed algorithm to removing duplicates in sets of subexpressions during program generation by MagicHaskeller. Our experimental

results show that it is effective for restraining the exponential bloat to some extent when using a relatively large primitive set. This means that our algorithm is useful in practical cases where we want to generate expressions consisted of standard library functions rather than reinventing well-known toy functions from scratch.

References

1. Olsson, R.: Inductive functional programming using incremental program transformation. *Artificial Intelligence* 74(1), 55–81 (1995)
2. Yu, T.: Polymorphism and genetic programming. In: Miller, J., Tomassini, M., Lanzi, P.L., Ryan, C., Tetamanzi, A.G.B., Langdon, W.B. (eds.) EuroGP 2001. LNCS, vol. 2038, pp. 218–233. Springer, Heidelberg (2001)
3. Schmid, U.: Inductive Synthesis of Functional Programs – Learning Domain-Specific Control Rules and Abstract Schemes. Springer, Heidelberg (2001); Habilitation thesis
4. Kitzelmann, E.: Data-driven induction of recursive functions from input/output-examples. In: AAIP 2007: Proceedings of the Workshop on Approaches and Applications of Inductive Programming, pp. 15–26 (2007)
5. Katayama, S.: Power of brute-force search in strongly-typed inductive functional programming automation. In: Zhang, C., Guesgen, H.W., Yeap, W.-K. (eds.) PRICAI 2004. LNCS (LNAI), vol. 3157, pp. 75–84. Springer, Heidelberg (2004)
6. Katayama, S.: Library for systematic search for expressions and its efficiency evaluation. *WSEAS Transactions on Computers* 12(5), 3146–3153 (2006)
7. Katayama, S.: Systematic search for lambda expressions. In: Trends in Functional Programming, Intellect, vol. 6, pp. 111–126 (2007)
8. Claessen, K., Hughes, J.: QuickCheck: a lightweight tool for random testing of Haskell programs. In: ICFP 2000: Proceedings of the 5th ACM SIGPLAN International Conference on Functional Programming, pp. 268–279. ACM, New York (2000)
9. Spivey, M.: Combinators for breadth-first search. *Journal of Functional Programming* 10(4), 397–408 (2000)
10. Spivey, M.: Algebras for combinatorial search. In: Workshop on Mathematically Structured Functional Programming (2006)
11. Hudelmaier, J.: Bounds on cut-elimination in intuitionistic propositional logic. *Archive for Mathematical Logic* 31, 331–354 (1992)
12. Dyckhoff, R.: Contraction-free sequent calculi for intuitionistic logic. *Journal of Symbolic Logic*, 795–807 (1992)
13. Katayama, S.: MagicHaskeller (2005),
<http://nautilus.cs.miyazaki-u.ac.jp/~skata/MagicHaskeller.html>

Identification of Subject Shareness for Korean-English Machine Translation

Kye-Sung Kim, Seong-Bae Park, Hyun-Je Song, Se-Young Park, and Sang-Jo Lee

Department of Computer Engineering

Kyungpook National University

702-701 Daegu, Korea

{kskim, sbpark, hjsong, sypark}@sejong.knu.ac.kr, sjlee@knu.ac.kr

Abstract. One of the most critical issues in translating Korean into other languages is the common use of empty arguments. Since even mandatory elements in Korean are often dropped unlike English, the missing elements should be resolved during translation to obtain grammatical sentences. In this paper, we focus on missing subjects in intra-sentential level, which can be regarded as the identification of subject sharing between clauses. In order to reflect syntactic information in resolving missing subjects, we use a parse tree kernel, a specialized convolution kernel. In experimental evaluation, syntactic information turns out to be positively related to the identification of subject shareness. Our method achieves an accuracy of 81.39% and outperforms the baseline system assuming that two adjacent clauses share a subject.

1 Introduction

Research on machine translation has been focusing on increasing the quality of translation. Recent studies of machine translation try to obtain good translations using statistical machine translation based on bilingual corpora. Since such statistical MT systems have a number of advantages between languages with a similar syntactic structure, some of recent studies focus on a word reordering between two languages [12,16].

Although the studies on machine translation between Korean and English have already been going on for more than ten years, the MT systems show poor performance for long sentences because of several factors [10]. The fundamental cause of poor performance comes from the differences in the word order of Korean and English, but another significant problem is to resolve missing arguments in sentences. This is necessary to improve quality of the translations including grammatical correctness. Unlike English, Korean allows free omission of elements and even mandatory elements are often dropped. Therefore, they should be recovered in order to obtain grammatical English sentences. The omission of elements in Korean can occur in various syntactic positions, but most of them appear in subject positions. According to Hong [15], the rate of subject drop is 57% in spoken Korean, which is higher than other elements. Kim [20] also showed that the proportions of clauses with a verb that contain an overt subject in Korean adult-adult conversations and writing are just 31% and 51% respectively. The corpus used in our experiments shows that the rate of clauses with missing

subjects is 67%. Therefore, this paper will handle the omission of subjects, which is the most frequent case.

In previous works on Korean-to-English machine translation systems, some studies proposed the use of semantic features for the recovery of missing elements [2,21]. However it is difficult to use semantic features in a practical system because of the lack of reliable semantic resources for Korean. Kim [14], in a view of dependency parsing, proposed a subject-clause (S-clause) as a group of words containing several predicates and a common subject. To detect the boundary of a S-clause in Korean texts, they employed only shallow morphological features for predicates.

In our approach, we focus on finding clauses which share a same subject in the same sentence. Since clauses which share a subject are likely to have similar structures, we propose a method which identifies common subjects between clauses through the use of syntactic information. Each sentence is assumed to be segmented into a set of clauses in advance. Then, all possible pairs of clauses are made in order to compare their syntactic information. Support vector machines with a parse tree kernel are used to determine if two clauses share a subject in our experiments. The parse tree kernel is a specialized convolution kernel and efficiently reflects structural information [11,1].

When a subject is omitted in a clause, the subject of previously appearing clause tends to be its referent [22]. The baseline method for our task considers that two adjacent clauses share a same subject and its accuracy is nearly 63%. In experimental evaluation, we show that the proposed method outperforms the baseline method and syntactic information plays an important role in solving this problem. Another benefit of our work is that the proposed method can be applied to similar problems for missing elements.

The rest of this paper is organized as follow. Section 2 surveys the previous works on recovery of missing elements. In Section 3, we describe the significance of subject identification. Section 4 proposes a method for identifying subject sharness in machine learning approach. Section 5 presents the experimental results. Finally, Section 6 concludes this paper with future research directions.

2 Related Works

Recent works on machine translation between Korean and English have focused on pattern-based translation [7,19]. Therefore, there have been a few previous studies on missing arguments in Korean-English machine translation. Egedi [2] proposed the use of semantic features for the recovery of topicalized arguments in the translation of Korean to English. Lee [21] suggested a discourse module for identifying the referents of missing arguments in their system. However, it is difficult to use semantic features in a practical systems, since reliable semantic resources for Korean are unavailable.

The resolution of missing subjects is mainly investigated in recent research on zero pronouns. These studies are mainly classified into two types. One approach is based on heuristic rules and most of them use a centering theory [13,10]. The centering theory [5] focuses on the resolution of inter-sentential anaphora and has been used as the basis for pronoun resolution. Since several rules and constraints used in the centering framework are English-oriented in general, it must be extended in order to handle languages with

Korean sentence :
영희가 바빠서 그곳에 가지 못했다.
English sentence :
Younghée / was busy / there / could not go
Reordered English sentence :
[subj?] could not go there because Younghée was busy
English translation :
<u>Younghée</u> could not go there because Younghée was busy
Edited English translation :
<u>Younghée</u> could not go there because <u>she</u> was busy

Fig. 1. An example of a Korean sentence and its English translation

free word order. Roh [10] suggested an algorithm for resolving zero pronoun in a cost-based centering model which is the revised centering model for Korean. However, they did not resolve all zero pronouns in Korean. Because zero pronouns in Korean freely occur in various grammatical positions, complicated heuristic rules should have been maintained in order to grasp all zero pronouns.

The other approach is based on machine learning methods. Kawahara [4] employed case frames and distance tendency for reflecting the structure in Japanese texts. They proposed a distance measure to capture structural preference between zero pronouns and their possible antecedents. However the measure did not completely reflect syntactic information in the texts. Most of previous studies assumed that before resolving them zero pronoun are correctly detected. However, Zhao [17] attempted to identify candidates of zero pronouns automatically.

Some studies recently attempted to combine the centering model and a machine learning methods. Isozaki [6] proposed a method that combines ranking rules and machine learning. The ranking rules are used in sorting the candidates of a zero pronoun while each candidate is classified using support vector machines. In a view of dependency parsing, Kim [14] proposed a subject-clause (S-clause) as a group of words containing several predicates and a common subject. They showed that the S-clause information is effective in analyzing long sentences. However, they employed only shallow features in sentences and did not consider syntactic information.

3 Significance of Subject Identification

Korean is a head-final language and it has a relatively free word order. Unlike English, the arguments of verbs are often omitted in Korean texts and it is one of the significant features of Korean.

The identification of missing subjects is an important problem in Korean-English machine translation, since English requires overt subjects. In addition, the repeated

(1) 항공기가 착륙을 시도하였으나 시계가 나빠 되돌아갔다.
A1 : 항공기가 착륙을 시도하였으나 the airplane tried to land
A2 : 시계가 나빠 the visibility was bad
A3 : 되돌아갔다 _____ returned
(2) 처음에는 서툴렀으나 시간이 가면서 큰 어려움이 없었다.
B1 : 처음에는 서툴렀으나 at first _____ [be] unhandy
B2 : 시간이 가면서 as time passes
B3 : 큰 어려움이 없었다. _____ [have] little difficulty

Fig. 2. An example of Korean sentences with missing subjects

subjects can be replaced with appropriate pronouns during translation. Therefore, the referents of missing subjects must be identified in order to ensure grammatical correctness of sentences and lexical appropriateness. Figure 1 shows the example of a Korean sentence with a missing subject and its English translation. The Korean sentence in Figure 1 consists of two clauses, which is one main clause and one subordinate clause. The subject ‘영희 (Younghhee)’ in the main clause is omitted and the missing subject needs to be resolved before translation. As mentioned above, the identification of subjects is related to the use of pronouns in English. In Figure 1, the redundant subject ‘영희 (Younghhee)’ in the subordinate clause can be replaced with ‘she’ since ‘영희 (Younghhee)’ is a human being and female. To solve this problem, we first must identify the referents of missing subjects in clauses. However, it is important to determine the boundary for identifying referents since possible candidates of a missing subject may not exist in the same sentence. In English, most cases prefer the previous subject when a subject is omitted in a clause [22], but it is not always true in Korean because of several factors such as frequent omission of subjects and the word order between both languages. Figure 2 shows another example of Korean sentences.

In Figure 2, two sentences (1) and (2) both consist of three clauses and they have two subordinate clauses and one main clause. For the missing subject in clause (A3), its possible candidates within the sentence are ‘airplane’ and ‘visibility’ which are in the preceding clauses. In the sentence (1), the actual missing subject is ‘airplane’ in clause (A1). However, the referent of missing subjects in sentence (2) does not exist in the same sentence. This is a more semantic problem and it needs to be handled in discourse level. When translating, it has to be translated into a generic or deictic pronoun.

In this paper, we focus on missing subjects in intra-sentential level. In Korean, most clauses without a subject (nearly 82%) share overt subjects in the same sentence (see

Table 4). Therefore, in this work, we first focus on resolving missing subjects of which referents can be found within the sentence.

4 Identification of Subject Shareness

4.1 Overview

Frequent omissions of subjects in Korean sentences imply that several predicates can share one subject. This is related to the subject-sharing problem of clauses. Therefore, we will resolve this problem by identifying whether clauses share same subjects. We assume that missing subjects are already detected and marked in each clause like most studies on zero pronouns. Thus, all the sentences considered in this work are complex compound ones which contains one or more missing subjects.

A given sentence is first segmented into a set of clauses. The clauses are made as pairs within the sentence. Each pair is made up of two clauses: one with a missing subject, the other with an overt subject. In Figure 3, the sentence consists of four clauses where clause (c3) and (c4) have null subjects. In our method, the number of selected pairs is four: (c1,c3), (c1,c4), (c2,c3) and (c2,c4). The pair (c1,c2) and (c3,c4) are excluded in our experiments. The first one is excluded because both clauses in the pair have overt subjects. The second one is done because all clauses have missing subjects.

(3) 공군기가 들어왔으나 시계가 나빠 착륙을 못하고 돌아갔다.

- C1 : 공군기가 들어왔으나
a military airplane came back
- C2 : 시계가 나빠
the visibility was bad
- C3 : 착륙을 못하고
_____ could not land
- C4 : 돌아갔다
_____ returned

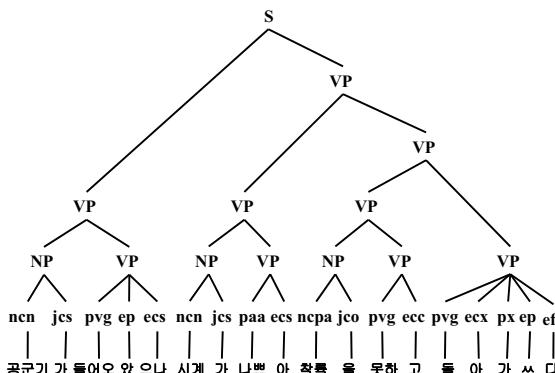


Fig.3. An example Korean sentence and its corresponding parse tree

4.2 Support Vector Machines

The subject-sharing between clauses can be considered as a binary classification task. Let $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ be a set of training examples where $y_i \in \{-1, +1\}$ and $\mathbf{x}_i = \langle c_{i1}, c_{i2} \rangle$. Here, each c_{ij} is a clause and y_j is the class label associated with this training sample. The value $+1$ of y_i implies that c_{i1} and c_{i2} share a subject.

The identification of subject shareness is to estimate a function $f : \mathbf{X} \rightarrow Y$. After the function f parameterized by θ is trained with D , the subject-sharing y^* of an unlabeled example \mathbf{x} can be determined by

$$y^* = \arg \max_{y \in \{-1, +1\}} (f(\mathbf{x}, \theta) = y).$$

Since our task is a binary classification, support vector machines (SVM) are adopted as an implementation of the function f . The decision function of SVMs is defined by

$$y^* = \operatorname{sgn}\left(\sum_{j \in SV} y_j \alpha_j \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}) + b\right), \quad (1)$$

where ϕ is a non-linear mapping function from \Re^N to \Re^H ($N \ll H$), SV is a set of support vectors, and $\alpha_j, b \in \Re$, $\alpha_j \geq 0$. The mapping function ϕ should be designed such that all training examples are linearly separable in \Re^H space. Since it is crucial to design an explicit form of ϕ , the inner product of $\phi(\mathbf{x}_j)$ and $\phi(\mathbf{x})$ is computed using a simple kernel such that

$$K(\mathbf{x}_j, \mathbf{x}) = \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}).$$

As a result, when a kernel K_P is designed to compute the inner product between two clauses, Equation (1) is rewritten as

$$y^* = \operatorname{sgn}\left(\sum_{j \in SV} y_j \alpha_j K_P(\mathbf{x}_j, \mathbf{x}) + b\right). \quad (2)$$

In order to apply SVM to our task, a number of positive and negative examples used as D are generated. The training examples are generated automatically from a parsed corpus. For instance, in Figure 3, a pair of (c1) and (c3) is used as a positive example since they share a subject ‘military airplane’. On the other hand, the pair of (c2) and (c3) is used as a negative example because these two clauses do not share a subject.

4.3 Parse Tree Kernel

A parse tree kernel is used in our method for modeling syntactic information of clause-pairs. The parse tree kernel is a specialized convolution kernel introduced by Haussler [3] and efficiently reflects structural information [11,1]. In the vector representation of a parse tree, the features correspond to the subtrees that can possibly appear in the parse tree. The value of a feature is the frequency of the corresponding subtree in the parse tree. Collins and Duffy [11] proposed a method to compute the inner product of two vectors without accessing the large vectors directly.

Let st_1, st_2, \dots be all subtrees possibly appearing in parse trees. A parse tree T is represented as a vector $V_T = (\#st_1(T), \#st_2(T), \dots, \#st_n(T))$, where $\#st_i(T)$ is the frequency of st_i in T . Then the inner product of the vector representations of two trees, T_1 and T_2 , becomes

$$\begin{aligned} & < V_{T_1}, V_{T_2} > \\ &= \sum_i \#st_i(T_1) \cdot \#st_i(T_2) \\ &= \sum_i \left(\sum_{n_1 \in N_{T_1}} I_{st_i}(n_1) \right) \cdot \left(\sum_{n_2 \in N_{T_2}} I_{st_i}(n_2) \right) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} C(n_1, n_2) \end{aligned} \quad (3)$$

where N_{T_1} and N_{T_2} are the sets of nodes in T_1 and T_2 respectively, $I_{st_i}(n_1)$ is a function that returns the frequency of st_i rooted at n_1 in T_1 , and $C(n_1, n_2)$ is the sum of the product of the numbers of times each subtree appears at n_1 and n_2 , i.e.

$$C(n_1, n_2) = \sum_i I_{st_i}(n_1) \cdot I_{st_i}(n_2)$$

$C(n_1, n_2)$ can be recursively calculated by using the following recursive rules:

1. If the productions at n_1 and n_2 are different, $C(n_1, n_2)=0$,
2. Else if both n_1 and n_2 are pre-terminals, $C(n_1, n_2)=1$,
3. Else,

$$C(n_1, n_2) = \prod_i^{nc(n_1)} (1 + C(ch(n_1, i), ch(n_2, i)))$$

where $nc(n_1)$ is the number of children of node n_1 and $ch(n_1, i)$ is the i -th child node n_1 .

A training example \mathbf{x}_i in D is a pair of clauses. Therefore, when $\mathbf{x}_i = < c_{i1}, c_{i2} >$ and $\mathbf{x}_j = < c_{j1}, c_{j2} >$, the parse tree kernel K_P for comparing them is defined as

$$K_P(\mathbf{x}_i, \mathbf{x}_j) = K_P(c_{i1}, c_{j1}) + K_P(c_{i2}, c_{j2}), \quad (4)$$

where K_P is an inner product of two clauses represented as parse trees. That is,

$$K_P(c_i, c_j) = < V_{c_i}, V_{c_j} >,$$

and the inner product is computed using Equation (3).

In applying the parse tree kernel to the identification of subject shareness, the normalization of the kernel is required since the kernel depends on the size of the trees. A normalized parse tree kernel is defined as follows.

$$K'_P(T_1, T_2) = \frac{K_P(T_1, T_2)}{\sqrt{K_P(T_1, T_1) \cdot K_P(T_2, T_2)}},$$

and $K'_P(T_1, T_2)$ is bounded between 0 and 1.

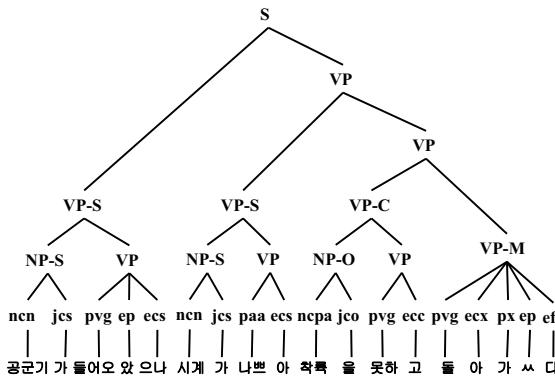


Fig. 4. An example of a parse tree with renamed node labels

4.4 Use of Renaming Nodes on a Parse Tree

The syntactic role of pre-terminal nodes will affect the results. For instance, the role of noun phrases with case markers is an important clue in the identification of missing subjects. In order to incorporate the syntactic role of pre-terminal nodes the types of phrases are first determined with respect to their syntactic role. The noun phrases and verb phrases are classified into three and five types respectively (see Table 1). The noun phrases are divided according to their case marker. The verb phrases are divided into five types: main, subordinate, coordinate, embedded, and others. The coordinate and subordinate type are determined by the kind of conjunction between clauses, and the embedded type is all dependent clauses excluding subordinate clauses. The main clauses are the ones which can stand alone as a complete simple sentence. Figure 4 shows the example of a parse tree with renaming nodes.

Table 1. The renamed nodes reflecting the syntactic roles

(1) Noun phrases		(2) Verb phrases	
NP-S	subject (case markers: ‘○’(-i), ‘가’(-ga))	VP-S	subordinate clause
NP-O	object (case markers : ‘을’(-ul), ‘를’(-lul))	VP-C	coordinate clause
NP	others	VP-E	embedded clause
		VP-M	main clause
		VP	others

5 Experiments

5.1 Dataset

We evaluate the proposed method using the parsed corpus which is a product of STEP 2000 project supported by Korean government. We manually analyze missing subjects in the parsed corpus. Then, the complex compound sentences with one or more missing subjects are only extracted. The number of selected sentences is 5,217 and the

Table 2. A simple statistics on the dataset used in our experiments

Dataset	Number
Sentences	5,217
Clauses	20,705
All possible pairs	36,061
Pairs actually extracted	17,169

Table 3. The distribution of overt subjects and missing subjects

Type of clause	Overt subj	Null subj
Coordinate Clause	474	1,334
Subordinate Clause	1,035	3,206
Main Clause	3,669	1,544
Embedded Clause	1,599	7,844
Total	33%	67%

sentences are segmented into 20,705 clauses (on average, 3.99 clauses/sentence and 6.78 words/clause). A simple statistics on the dataset is given in Table 2. Among all possible pairs of clauses, there are 17,169 clause-pairs with missing subject in one of two clauses.

Table 3 shows the distribution of both overt subjects and missing subjects in our dataset. The use of overt subjects is relatively frequent in main clauses and the other three clauses often omit a subject in a similar proportion. As a result, the proportions of clauses with an overt subject is only 33% in all clauses. In addition, most missing subjects (nearly 82%) are referring to overt subjects within the same sentence (inter-sentential) as shown in Table 4. There are a few cases which refer to something reached beyond the same sentence. Accordingly, most of missing subjects in complex compound sentences can be resolved in inter-sentential level.

Table 4. The boundary for detecting referents of missing subjects

Intra-sentential level	Inter- or extra-sentential level
11,383	2,545

5.2 Experimental Results and Analysis

Our experiments are performed in five-fold cross validation and *SVMlight* [18] is used as a classifier. The precision and recall in our method is calculated as follows and the results are shown in Table 5.

$$\text{Precision} = \frac{\text{number of correctly identified pairs}}{\text{number of identified pairs}}$$

$$\text{Recall} = \frac{\text{number of correctly identified pairs}}{\text{number of true pairs}}$$

Table 5. Experimental results

Method	Accuracy	Prec	Rec	F-score
Baseline	0.6233	0.4800	0.5700	0.5275
Svm1	0.8143	0.7681	0.6683	0.7147
Svm2	0.8139	0.7640	0.6919	0.7262

svm1 : using an original parse tree

svm2 : using a parse tree with renaming nodes

The baseline method for our task assume that two adjacent clauses share a same subject and its accuracy is nearly 62.3%. In Figure 5, ‘Svm1’ is the result obtained from the parse tree kernel and ‘Svm2’ is the result using the tree with renaming nodes under the same method. The F-score of the proposed method is 72.62 and the proposed method outperforms the baseline method.

5.3 Use of Composite Kernel

It is generally believed that the meaning of words could affect the omission of elements. In order to see the effect of this semantic information, a composite kernel of a parse tree kernel and a lexical semantic kernel is adopted. The parse tree kernel models syntactic information, while the lexical semantic kernel measures semantic similarity of the words in a clause pair.

Like the parse tree kernel in Equation (4), a lexical semantic kernel K_L of two data point $\mathbf{x}_i = \langle c_{i1}, c_{i2} \rangle$ and $\mathbf{x}_j = \langle c_{j1}, c_{j2} \rangle$ is defined as

$$K_L(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^2 \sum_{\alpha \in W_{ik}, \beta \in W_{jk}} K_l(\alpha, \beta),$$

where W_{ik} is a set of words appearing in a clause c_{ik} . $K_l(\alpha, \beta)$ returns the lexical similarity between two words α and β . It is measured by the semantic similarity of Jiang and Conrath [9] based on English WordNet, since no reliable thesaurus for Korean is publicly available. All Korean words are translated into English words through a bilingual dictionary.

The composite kernel K for resolving missing subjects is then

$$K = r \cdot K_P + (1 - r) \cdot K_L,$$

where r ($0 \leq r \leq 1$) is a mixing parameter. Figure 5 shows the result obtained with the composite kernel with various values of r . As r increases, the performance monotonically increases but gets flatten when r is larger than 0.6. That is, from $r = 0.6$ the performance is not sensitive to the change of r . The fact that the performance with larger r is superior to that with small r implies that syntactic information is more positively related to the resolution of missing subjects.

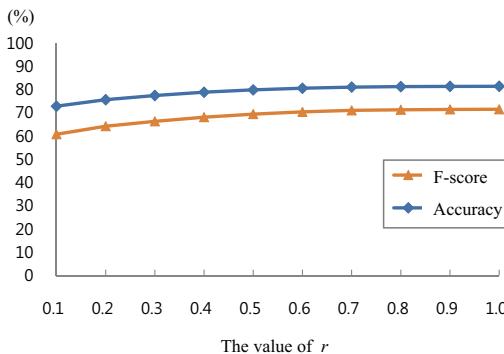


Fig. 5. The variation of the parameter r and the performance

6 Conclusion

We proposed a method for resolving missing subjects in Korean. Since most missing subjects can be resolved in intra-sentential level, the omission of elements is regarded as the identification of subject sharing between clauses. The identification is made with support vector machines which use the parse tree kernel to compare syntactic information of clause-pairs.

Our method outperforms the baseline system assuming that two adjacent clauses share a subject. In experimental evaluation, a composite kernel is also compared with a parse tree kernel in order to see whether the meaning of words affects the recovery of the element omission. According to the experimental results, the syntactic information plays an important role in solving our task.

In the future, we will apply the proposed method to a practical Korean-English machine translation system. Our work can be applied to similar problems of missing elements such as zero pronouns, and coreference resolution.

Acknowledgements

This work was supported in part by MIC & IITA through IT Leading R&D Support Project and by the Korean Ministry of Education under the BK21-IT Program.

References

1. Moschitti, A.: Making Tree Kernels Practical for Natural Language Learning. In: proceedings of the 11th International Conference on European Association for Computational Linguistics, pp. 113–120 (2006)
2. Egedi, D., Palmer, M., Park, H.S., Joshi, A.K.: Korean to English Translation Using Synchronous TAGs. In: Proceedings of the First Conference of the Association for Machine Translation in the Americas, pp. 48–55 (1994)
3. Haussler, D.: Convolution Kernels on Discrete Structures. UCS-CRL-99-10, UC Santa Cruz (1999)

4. Kawahara, D., Kurohashi, S.: Zero Pronoun Resolution based on Automatically Constructed Case Frames and Structural Preference of Antecedents. *Journal of Natural Language Processing* 11(3), 3–19 (2004)
5. Grosz, B.J., Joshi, A.K., Weinstein, S.: Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics* 21(2), 203–225 (1995)
6. Isozaki, H., Hirao, T.: Japanese zero pronoun resolution based on ranking rules and machine learning. In: *Proceedings of Empirical Methods in Natural Language Processing*, pp. 184–191 (2003)
7. Kim, J.-J., Choi, K.-S., Chae, Y.-S.: Phrase-Pattern-based Korean to English Machine Translation using Two Level Translation Pattern Selection. In: *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 31–36 (2002)
8. Peral, J., Ferrandez, A.: Pronominal Anaphora Generation in an English-Spanish MT Approach. In: *Computational Linguistics and Intelligent Text Processing*, pp. 187–196 (2002)
9. Jiang, J.J., Conrath, D.W.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: *Proceedings of the 10th International Conference on Research in Computational Linguistics* (1997)
10. Roh, J.-E., Lee, J.-H.: An Empirical Study for Generating Zero Pronoun in Korean based on Cost-based Centering Model. In: *Proceedings of Australasian Language Technology Association*, pp. 90–97 (2003)
11. Collins, M., Duffy, N.: Convolution Kernels for Natural Language. In: *Proceedings of NIPS 2001*, pp. 625–632 (2001)
12. Collins, M., Koehn, P., Kucerova, I.: Clause Restructuring for Statistical Machine Translation. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 531–540 (2005)
13. Kim, M.-K.: A Centering Dynamics Approach to Zero Pronouns in Korean. *The Discourse and Cognitive* 10(3), 57–73 (2003)
14. Kim, M.-Y., Lee, J.-H.: Two-Phase S-Clause Segmentation. *IEICE Transaction on Information and System* E88-D(7), 1724–1736 (2005)
15. Hong, M.: Centering theory and Argument Deletion in Spoken Korean. *The Korean Journal Cognitive Science* (11-1), 9–24 (2000)
16. Chang, P.-C., Toutanova, K.: A Discriminative Syntactic Word Order Model for Machine Translation. In: *Proceedings of 45th Annual Meeting of the Association for Computational Linguistics*, pp. 9–16 (2007)
17. Zhao, S., Ng, H.T.: Identification and Resolution of Chinese Zero Pronouns: A Machine Learning Approach. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 541–550 (2007)
18. Joachims, T.: Making large-Scale SVM Learning Practical. In: Scholkopf, B., Burges, C., Smola, A. (eds.) *Advances in Kernel Methods - Support Vector Learning*. MIT-Press, Cambridge (1999)
19. Roh, Y.-H., Hong, M., Choi, S.-K., Lee, K.-Y., Park, S.-K.: For the Proper Treatment of Long Sentences in a Sentence Pattern based English-Korean MT System. In: *Proceedings of Machine Translation Summit IX*, pp. 23–27 (2003)
20. Kim, Y.-J.: Subject/Object Drop in the Acquisition of Korean: A Cross-Linguistic Comparison. *East Asian Linguistics* 9(4), 325–351 (2000)
21. Lee, Y.-S., Yi, W.S., Seneff, S., Weinstein, C.J.: Interlingua-Based Broad-Coverage Korean-to-English Translation in CCLINC. In: *Proceedings of the first International Conference on Human language Technology Research*, pp. 1–6 (2001)
22. Leffa, V.J.: Clause Processing in Complex Sentences. In: *Proceedings of 1st International Conference on Language Resources and Evaluation*, pp. 937–943 (1998)

Agent for Predicting Online Auction Closing Price in a Simulated Auction Environment

Deborah Lim, Patricia Anthony, and Chong Mun Ho

School of Engineering and Information Technology, Universiti Malaysia Sabah

Locked Bag 2073, 88999 Kota Kinabalu, Sabah Malaysia

Tel.: +6088-320347; Fax: +6088-320348

deborahlim.05@gmail.com, panthony@ums.edu.my, cmho@ums.edu.my

Abstract. Auction markets provide centralized procedures for the exposure of purchase and sale orders to all market participants simultaneously. Online auctions have effectively created a large marketplace for participants to bid and sell products and services over the Internet. eBay pioneered the online auction in 1995. As the number of demand for online auction increases, the process of monitoring multiple auction houses, picking which auction to participate in, and making the right bid become a challenging task for the consumers. Hence, knowing the closing price of a given auction would be an advantage since this information will be useful and can be used to ensure a win in a given auction. However, predicting a closing price for an auction is not easy since it is dependent on many factors. This paper reports on a predictor agent that utilises the Grey System Theory to predict the closing price for a given auction. The performance of this predictor agent is compared with another well known technique which is the Artificial Neural Network. The effectiveness of these models is evaluated in a simulated auction environment.

Keywords: Online Auction, Grey System Theory, Artificial Neural Network.

1 Introduction

Auctions have been widely used for centuries. An auction is defined as a bidding mechanism, described by a set of auction rules that specify how the winner is determined and how much he has to pay [17]. Online auctions have expanded rapidly over the decade and had turned to a fascinating new type of business or commercial transaction in the digital era. Unlike traditional auction houses, online auction websites offer a better place for people to purchase and publicize their merchandise through a bidding process. Over the last few years, the number of online auction houses has increased tremendously. Some examples of popular online auction houses include e-Bay, Amazon, Yahoo!Auction, Priceline, UBid, lelong.com and FirstAuction. Online auction has also given consumers a “virtual” flea market with all the new and used merchandises from around the world. The auctioneers also have the ability to market their valuable items globally.

Online auctions provide many benefits compared to the traditional auctions. One disadvantage of having traditional auction is that it requires simultaneous participation

of all bidders or agents at the same location. In online auction, this does not exist as online auction allows clients to make their purchases anywhere anytime. Online auctions also provide bidders more flexibility on when to submit their bids since online auctions usually last for days or even weeks. Compared to the traditional business type, internet auctions can be a co-effective way on testing on the product markets and are able to liquidate dated or overstocked merchandise especially for small business owners. In other words, internet auction is able to be used to test the market response for a new product and is a good way to sell the left over products quickly. Besides that, online auction can be more effective as the target audiences will be in a mass amount where there is no geographical limitation since both sellers and buyers do their trading in a “virtual” environment and any payment transaction can be made through the online banking. Having a relative low price and wider market in products and services, it had made the online auction a success where it attracts many bidders and also sellers as well. Online auctions also allow sellers to sell their goods efficiently and with little exertion required.

There are four main types of single-sided auctions that are commonly used in traditional auctions [9] which are ascending-bid auction (also called the open, oral, or English auction), descending-bid auction (also called Dutch auction), first-price sealed bid auction and second-price sealed bid auction (also called Vickrey auction). The English auction begins with the lowest price and bidders are free to raise their bid successively until there are no more offers to raise the bid and the bidder with the highest bid will be the winner. The Dutch auction is the opposite of an English auction, where the auctioneer will start with an initial high price and is progressively lowered until there is an offer from a bidder to claim the item. In the first price sealed bid, each bidder submits their offer for an item privately. The highest bidder gets the item and pays for the item based on his bid value. The Vickrey auction is similar to the first-price sealed bid auction, as the item goes to the highest bidder but he only pays a price equal to the second highest bid. Online auctions are similar to the traditional auctions but most auctions are constrained by time. Online auctions usually last for days and week depending on the seller’s requirement.

Due to the proliferation of these online auctions, consumers are faced with the problem of monitoring multiple auction houses, picking which auction to participate in, and making the right bid to ensure that they get the item under conditions that are consistent with their preferences [2]. These processes of monitoring, selecting and making bids are time consuming. The task becomes even more challenging when the individual auctions have different start and end times. Moreover, auctions can last for days or even weeks. Besides that, every bidder has his own reservation price or maximum amount that he is willing to bid for each item. If bidders are able to predict the closing price for each auction then they are able to make a better decision on the time, place and the amount they can bid for an item. In a situation where a bidder has to decide among the many auctions that are currently ongoing, this knowledge on closing price for an auction would be useful for the bidder to decide on which auction to participate, when to participate and at what price. There are other considerations that need to be taken into account to ensure that the bidder wins in a given auction. For example, in eBay, any bidder who wishes to participate in an online auction does not have any

information on how many bidders will bid in the auction, the number of bids and how the auction will progress over time.

Bidder who can accurately predict the closing price of a given auction would definitely has the edge against the other bidders since the bidder can decide when to bid and how much to bid. Besides that it would be a lot easier for bidders if they can guess the price of the auction, so that they only need to bid at the last minutes of the auction at a bid value slightly higher than the predicted price. Unfortunately, predicting a closing price for an auction is not easy since it is dependent on many factors such as the behaviour of the bidders and the number of the bidders who are participating in that auction.

Having many obstacles in bidding, many investors have been trying to find a better way to predict auction closing price accurately. Neural Network, Fuzzy Logic, Evolutionary Computation, Probability Function and Genetic Algorithm, have been integrated to become a more commendable practical model for prediction purposes. A large body of analysis techniques has been developed, particularly from methods in statistics and signal processing.

As being acknowledged, forecasting or prediction needs a lot of historical data. The most popular method which requires a large data set is Artificial Neural Network (ANN). It is based on computer algorithms that attempt to simulate the parallel, highly interactive distributed processing in brain tissue. ANN has been successfully applied to a wide variety of problems, such as storing and recalling data or patterns, classifying patterns, performing general mapping from input patterns to output patterns, grouping similar patterns, or finding solutions to constrained optimization problems [10]. In 2006, Li *et al.* [11] have concentrated on predicting the final prices of online auction items. Taylor and Buizza [12] worked on developing a more efficient load forecast by using ANN with weather ensemble predictions instead of single weather point prediction.

Another method used for prediction is the Grey System Theory. This is a new theory and method which applies to the study of unascertained problems with few or poor incoming information [13]. This new theory makes the process of prediction much easier since as little as four observations are required to predict the next value. It has been successfully applied to economical, management, social systems, industrial systems, ecological systems, education, traffic, environmental sciences, and geography [14].

There are many researches that have been engaged in the prediction and forecasting using Grey System Theory in the real world phenomenon. Lin & Valencia [12] proposed Grey Systems Science to the investigation of the relative degrees of importance of 22 variables affecting the individual or family in the decision-making of migration. In 2007, Wang *et al.* [16] applied Grey System GM (1, N) model to predict the spring flow in China. In Taiwan, Chiou *et al.* [4] used Grey prediction model for forecasting the planning material of equipment spare parts in the Navy of Taiwan. In our previous work, we developed a predictor agent to predict the closing price of online auction using the Grey System Theory [1]. We evaluated the effectiveness of the Grey System Agent, against the performance the Simple Exponential Agent [5] and the Time Series Agent [6]. We found that the Grey System Agent achieved a higher accuracy in predicting the online auction closing price.

In this paper we will investigate and compare the effectiveness of the Grey System Agent against a more established computer science technique namely the Artificial Neural Network in predicting the closing price of online auction since we have successfully proven its superiority against the mathematical techniques. The remainder

of the paper is structured as follows. In Section 2, the Grey System Theory and its Prediction Algorithm are elaborated. Section 3 describes the design of the Artificial Neural Network and the Backpropagation Prediction Algorithm. In Section 4, the electronic simulated marketplace used in this experiment will be described in detail. The experimental results are elaborated in Section 5, and finally the conclusion and future work are discussed in Section 6.

2 The Grey System Theory Design

The Grey System Theory was first proposed by Deng Julong (1982) [8], where this theory works on unascertained systems with partially known and partially unknown information by drawing out valuable information and also by generating and developing the partially known information where it helps in describing correctly and monitor effectively on the systemic operational behaviour [13]. Basically, the Grey System Theory was chosen based on colour [14]. For instance, “black” is used to represent unknown information while “white” is used to represent complete information. Those partially known and partially unknown information is called the “Grey System Theory”.

The Grey System Theory has been successfully applied to various fields and had made a success in analyzing uncertain systems that have multi-data inputs, discrete data, and insufficient data. Traditional prediction methods, such as time series, usually require a large amount of historical data and process a known statistical distribution in order to make an accurate assessment and prediction of the required parameters [4]. In contrast to the traditional prediction method, the main attributes of the Grey System theory, which is the core of the Grey forecasting theory, are it does not need to make strict assumptions about the data set and is used successfully to analyse uncertain systems that have multi-data inputs, discrete data, and insufficient data. These simplify data collection and allow for timely predictions to be made. Grey Systems Theory explores the law of subject’s motivation using functions of sequence operators according to information coverage. It is different from fuzzy logic since it emphasizes on objects with definite external extensions and vague internal meanings. Table 1 shows the Grey System Theory compared to other traditional forecasting models [3]. It can be seen that this model only requires short-term, current and limited data in order to predict a given value.

Grey prediction is a quantitative prediction based on Grey generating function, GM (1,1) model, which uses the variation within the system to find the relations between sequential data and then establish the prediction model. The Grey Prediction Model is

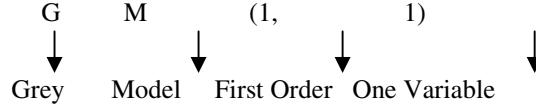
Table 1. Traditional Forecasting Model Attributes

Mathematical model	Minimum Observation	Type of sample	Sample interval	Mathematical requirements
Simple exponential function	5-10	Interval	Short	Basic
Regression analysis	10-20	Trend	Short	Middle
Casual regression	10	Any type	Long	Advanced
Box-Jenkins (Time Series ARIMA)	50	Interval	Long	Advanced
Neural network	Large number	Interval or not	Short	Advanced
Grey prediction model	4	Interval	Long	Basic

derived from the Grey System, in which one examines changes within a system to discover a relation between sequence and data. After that, a valid prediction is made to the system.

The equation $x^{(0)}(k) + az^{(1)}(k) = b$ is called a GM (1, 1) Model [14].

The meaning of the symbol GM (1, 1) is given as follows [14]:



The Grey Prediction Model has the following advantages [4]: (a) It can be used in situations with relatively limited data down to as little as four observations, as stated in Table 1. (b) A few discrete data are sufficient to characterize an unknown system. (c) It is suitable for forecasting in competitive environments where decision-makers have only accessed to limited historical data.

2.1 The Grey System Theory Prediction Algorithm

In this section, we describe our predictor agent algorithm which focuses on the Grey generating function, GM used in grey prediction [7, 14]. The algorithm of GM (1,1) can be summarized as follows.

Step 1. Establish the initial sequence from observation data. In this case, the data used is the previous values of the online auction closing price observed over time.

$$f^0 = \{f_1^0, f_2^0, f_3^0, \dots, f_n^0\}, \quad \text{where } n \geq 2. \quad (1)$$

Step 2. Generate the first-order accumulated generating operation (AGO) sequence

$$f^1 = \{f_1^1, f_2^1, f_3^1, \dots, f_n^1\}, \quad \text{where } f_t^1 = \sum_{k=1}^t f_k^0 \quad \text{and} \quad t = 1, 2, \dots, n. \quad (2)$$

Step 3. The grey model GM (1,1)

$$f_{t+1}^0 = a \left[-\frac{1}{2} (f_t^1 + f_{t+1}^1) \right] + b, \quad \forall t \geq 1. \quad (3)$$

Step 4. Rewrite into matrix form

$$\begin{bmatrix} f_2^0 \\ f_3^0 \\ \vdots \\ f_n^0 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}(f_2^1 + f_1^1) & 1 \\ -\frac{1}{2}(f_3^1 + f_2^1) & 1 \\ \vdots & \vdots \\ -\frac{1}{2}(f_n^1 + f_{n-1}^1) & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}, \quad \text{where } a \text{ and } b = \text{constant values.} \quad (4)$$

Step 5. Solve the parameter a and b

$$\begin{bmatrix} a \\ b \end{bmatrix} = (B^T B)^{-1} B^T F^0, \quad \text{where } F^0 = \begin{bmatrix} f_2^0 \\ f_3^0 \\ \vdots \\ f_n^0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} -\frac{1}{2}(f_2^1 + f_1^1) & 1 \\ -\frac{1}{2}(f_3^1 + f_2^1) & 1 \\ \vdots & \vdots \\ -\frac{1}{2}(f_n^1 + f_{n-1}^1) & 1 \end{bmatrix}. \quad (5)$$

Step 6. Estimate AGO (Accumulating Generation Operators) value

$$\hat{f}_{t+1}^1 = \left[f_t^0 - \left(\frac{b}{a} \right) \right] e^{-at} + \left(\frac{b}{a} \right), \quad \forall t \geq 1. \quad (6)$$

Step 7. Get the estimate IAGO (Inverse Accumulating Generation Operators) value or the estimated closing price for a given auction.

$$\hat{f}_t^0 = \hat{f}_t^1 - \hat{f}_{t-1}^1, \quad \forall t \geq 2. \quad (7)$$

Step 8. We use the average residual error for each set of data to calculate the accuracy of the predicted data. The formula for the average residual error (ARE) is given as

$$\left(\frac{1}{n} \sum_{t=1}^n \frac{|f_t^0 - \hat{f}_t^0|}{f_t^0} \right) \times 100\%. \quad (8)$$

where f_t^0 = real value of exchange rate at time t

\hat{f}_t^0 = estimated value of exchange rate at time t

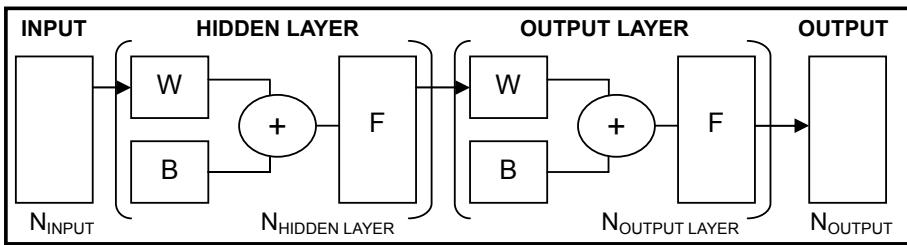
n = total observation .

3 Artificial Neural Network Design

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by biological nervous systems, such as the brain, process information [10]. A Neural Network is one of the most powerful data modeling tool that is able to capture and represent complex input and output relationships [10]. As in nature, a Neural Network consists of a large number of simple processing element called neurons, units, cells, or notes. First of all, ANN processes the information with the simple elements called neurons. Each neuron is connected to other neurons by means of directed communication links, each with an associated weight. ANN continues to pass signals between neurons over connection links. Lastly, each connection link has an associated weight, which in a typical neural network where the multiple the signal transmitted and each neuron applies an activation function to its net input to determine its output signals. The weight represents information being used by the net to solve a problem. By its learning algorithm to adjust weighted connections between Artificial Neurons, ANN is able to be "trained" to conduct specific tasks. The network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Here we apply an ANN training method, Feedforward Backpropagation, which is simply a gradient descent method to minimize the total squared error of the output computed by the net.

3.1 Feedforward Backpropagation Prediction Algorithm

To build a Neural Network, the number of input neurons, the number of output neurons, the number of hidden layers, the number of output layers and the number of neuron in both layers need to be determined. The ANN Agent predicts the closing price based on 60 sets of historical data. Besides the historical data, the starting price, the number of bidders and the number of bids are taken into consideration in the ANN prediction model. All the data are normalized before they are fed to the network and the weights of



Where N_{INPUT} = number of element in input vector
 $N_{HIDDEN\ LAYER}$ = number of neuron in hidden layers
 $N_{OUTPUT\ LAYER}$ = number of element in output layers
 N_{OUTPUT} = number of element in output vector
 W = weight value
 B = bias value
 F = activation function

Fig. 1. Feedforward Backpropagation Algorithm

the data set are found by Feedforward Backpropagation Algorithm. The Feedforward Backpropagation algorithm is shown in Figure 1.

The purpose of ANN Agent is to learn the trend of bidding from the historical data sets which we provided. In our experiments, the historical data are divided into the ratio of 2:1. The first two portion (40 sets of historical data) are set apart for learning purpose while the last portion (20 sets of historical data) are reserved for testing. In our case, the results using one hidden layer with 50 neurons and one output layer with 1 neuron produced the highest accuracy. For both hidden layer and output layer, three functions are available, namely Tangent Sigmoid Transfer Function (Tansig), Logarithm Sigmoid Transfer Function (Logsig) and Linear Transfer Function (Purelin). There are all together nine possibilities for choosing different functions at the hidden layers and output layer. All possibilities have been tested and the four possibilities which give the most accurate predicted results are used in the experiment.

4 The Electronic Simulated Marketplace

To compare the performance of the Grey Theory Agent against the ANN Agent, we developed an electronic marketplace to simulate the real online auctions environment [2]. This simulated electronic marketplace consists of a number of auctions that run concurrently. There are three types of auctions running in the environment: English, Dutch and Vickrey. The English and Vickrey auctions have a finite start time and duration generated randomly from a standard probability distribution, the Dutch auction has a start time but no pre-determined end time. At the start of each auction (irrespective of the type), a group of random bidders are generated to simulate other auction participants. These participants operate in a single auction and have the intention of buying the target item and possessing certain behaviours. They maintain the information about the item they wish to purchase, their private valuation of the item (reservation price), the starting bid value and their bid increment. These values are generated randomly from a standard probability distribution. Their bidding behaviour

is determined based on the type of auction that they are participating in. The auction starts with a predefined starting value; a small value for an English auction and a high value for a Dutch auction. There is obviously no start value for a Vickrey auction. The marketplace is flexible and can be configured to take up any number of auctions and any value of discrete time. All the auctions are assumed to be auctioning the item that the consumers are interested in and all auctions are selling the same item.

5 Experimental Evaluation

The purpose of this experiment is to compare and evaluate the performance of our predictor agent, the Grey System Agent against the ANN Agent that uses the Feedforward Backpropagation technique. Here, the Grey System Agent requires 4 – 10 historical data to predict the closing price of the auction whereas the ANN Agent requires 50 or more historical data (since ANN requires a large data set in order to make a prediction).

Using the simulated marketplace, we ran the auction from $t = 1$ until $t = 150$. We have also set most of auctions to close after $t = 30$. In one particular run, the closing price history for all auctions running in a marketplace are shown from $t = 41$ until $t = 150$. From $t = 41$ onward, consistency of obtaining the closing price at each t is preserved. We reset the t to start from $t = 1$ ($t = 41$) until $t = 110$ ($t = 150$) as the historical data. The accuracy of predicted values by the two agents is measured using the ARE.

This experimental evaluation is divided into two parts. In the first part of the experiment, we will calculate the predicted closing price by using fixed historical data generated by the simulated marketplace using the two agents. In the second part, we use moving data rather than fixed historical data for the two agents.

5.1 Fixed Historical Data

In the first experiment, five future closing price data are predicted by the two agents. The results of the predictions by the two agents are shown in Table 2, and 3.

It can be seen in Table 2 that by using 4 until 10 of the latest historical data, the ARE for Grey System Agent falls between 0.77% to 9.50% and the highest accuracy is recorded using 6 historical data (ARE = 0.77%). The ARE increases when more historical data are used and the highest ARE is recorded by using 10 historical data with 9.50%.

In Table 3, it can be seen that the average residual error (ARE) of the predicted results fall between 2.50% to 3.04%. It is also observed that the most accurate results (ARE is 2.50%) is obtained by using Tansig functions for both the hidden layer and the output layer.

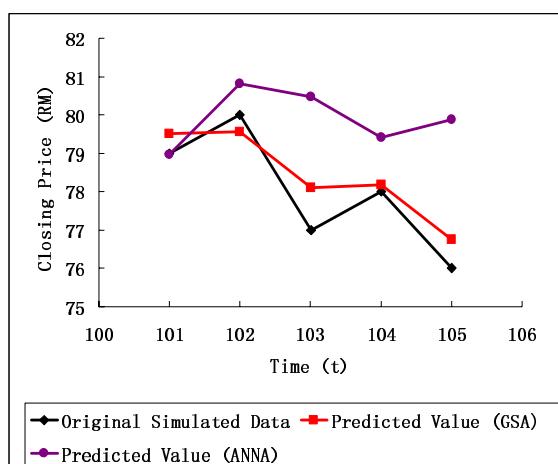
Based on these results, we can conclude that Grey System Agent, that used 6 historical data produced better result (ARE = 0.77%) than the Artificial Neural Network Agent (ARE = 2.50% for 60 set of historical data by using Tansig function for both hidden and output layers). Even with 6 historical data, the Grey System Agent is able to predict more accurately the closing price of the auctions in our simulated auction environment. Figure 2 shows how far off the two predictions against the actual closing price in this particular run. The predicted values by the Grey System Agent follow the trend of original data very closely whereas the predicted values for the ANN Agent also follows the shape of the actual data but it is further from the actual points.

Table 2. Result Performed by using Grey System Agent

No of Historical Data	Time t = 101, Original Data = 79	Time t = 102, Original Data = 80	Time t = 103, Original Data = 77	Time t = 104, Original Data = 78	Time t = 105, Original Data = 76	ARE (%)
4	79.34	79.21	77.59	76.48	74.90	1.12
5	78.53	77.09	73.71	71.38	67.13	5.15
6	79.52	79.55	78.10	78.17	76.76	0.77
7	75.36	74.29	71.62	71.37	69.53	6.56
8	80.72	82.40	83.04	86.65	89.23	8.30
9	80.47	81.82	82.05	85.17	87.17	7.64
10	80.81	82.60	83.37	87.13	89.87	9.50

Table 3. Result Performed by using Artificial Neural Network Agent

Function Type (Hidden Layer & Output Layer)	Time t = 101, Original Data = 79	Time t = 102, Original Data = 80	Time t = 103, Original Data = 77	Time t = 104, Original Data = 78	Time t = 105, Original Data = 76	ARE (%)
Tansig & Tansig	78.96	80.81	80.47	79.41	79.88	2.50
Logsig & Logsig	81.97	80.38	79.43	79.01	79.81	2.74
Tansig & Logsig	80.71	80.39	80.13	79.00	80.25	2.72
Logsig & Tansig	76.01	80.31	79.65	78.87	80.90	3.04

**Fig. 2.** Results Obtained By the Two Agents Over Time

5.2 Moving Historical Data

As mentioned earlier, at every time steps, there are auctions that will be closing. This simply means that, if one wants to predict the future data, one has to take into account the auctions that will be closing between the current time and the next time steps. Hence, we performed the following experiments in which we compared the result based on the moving historical data. Table 4 shows the result obtained by using 6 moving historical data for the Grey System Agent and 60 sets of moving historical data the Artificial Neural Network Agent compared with the original data generated by the

simulated auction. These values are used because of their performance in the previous experiment. In Table 4, it can be seen that The Grey System Agent outperformed the ANN Agent when $t = 104, 105, 107$ and 108 . Here, we can conclude that, even by using moving historical data, the predicted closing price of the Grey System Agent is more accurate when compared to the predicted closing price of the ANN agent even with a minimal number of observation data.

Figure 3 shows the predicted values over time for the Grey System Agent using fixed historical data and moving data. It can be observed that the values predicted by using both fixed and moving historical data are still very close to the real value at time step ($t = 102$ until $t = 106$). At $t = 107$, the accuracy using fixed historical data slightly decreases while the moving historical data result always follow the trend of the original data until the end. Based on these experiments it can be concluded that the Grey System Agent is able to predict a value that is very close to the real value over time by making use of the moving historical data. It has consistently outperformed the ANN agent in terms of ARE when predicting the values either using fixed historical data (Figure 2) or moving historical data (Table 4).

The ANN Agent also performed better when using the moving historical data as shown in Figure 4. From $t = 102$ until $t = 107$, the predicted closing price values using the moving data are more accurate than the predicted closing price values using the fixed historical data.

Table 4. Result Obtained by the Two Agents for Moving Historical Data

Time (t)	Original Data	(GSA) Forecast Using 6 Moving Historical Data (ARE %)	(ANNA) Forecast Using 60 Sets of Moving Historical Data by Using Tansig Function for both layer (ARE %)
102	80	78.86 (1.43)	80.48 (0.60)
103	77	78.12 (1.45)	76.71 (0.38)
104	78	78.60 (0.77)	79.25 (1.60)
105	76	77.41 (1.86)	79.51 (4.62)
106	79	78.45 (0.70)	79.28 (0.36)
107	80	80.02 (0.03)	81.28 (1.60)
108	82	82.24 (0.29)	80.05 (2.38)
Total Average ARE		0.93	1.65

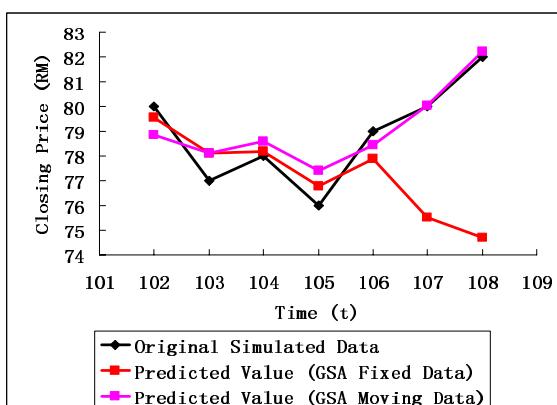


Fig. 3. Results for the Grey System Agents over Time

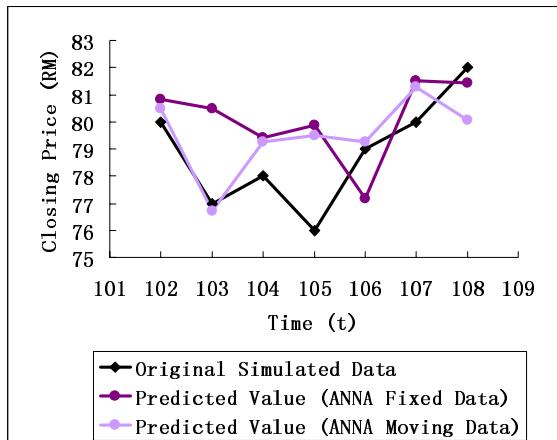


Fig. 4. Results for the Artificial Neural Network Agents over Time

The average ARE of Grey System Agent for moving historical data is 0.93% while ANN Agent given 1.65%. Besides that, the ANN Agent require 60 sets of historical data and also four input parameters (which are the closing price, the starting price, the number of bidder and also the number of bid) whereas the Grey System agents only requires 6 historical data. The Grey System Agent also achieved higher accuracy in predicting the closing price either for fixed historical data or moving historical data. In the context of online auction, user may only be able to access or view several values from the past history, so in this case the Grey System Agent can be used to make a quick prediction on the closing price of a given auction.

6 Conclusion and Future Work

This paper shows that using both methods, the accuracy rate always exceeds more than 95%. The Grey System Agent gives better result when less input data are used while the Artificial Neural Network Agent can only be used with the availability of a lot of information as well as many input parameters. The experimental results also showed that using moving historical data produces higher accuracy rate than using fixed historical data for both agents. This is important since, bidders in an online auction need to take into accounts all the auctions that are going to close within the prediction period. This closing price knowledge can then be used by the bidder to decide which auction to participate, when and how much to bid. This information will also allow the bidder to maximize his chances of winning in an online auction. Besides that, in the context of online auction, user may only be able to access or view several values from the past history, so in this case the Grey System Agent can be used to make a quick prediction on the closing price of a given auction.

For future work, we would also like to apply our prediction method to predict on the auction closing price in eBay and other online auctions. We would also like to investigate the applicability of the Grey System Theory to predict the bidder's arrival and their bids in a given auction.

References

1. Anthony, P., Deborah, L., Ho, C.M.: Predicting online Auction Closing Price Using Grey System Theory. In: Proceeding of Managing Worldwide Operations & Communications with Information Technology, pp. 709–713. IGI Publishing, Vancouver (2007)
2. Anthony, P., Jennings, N.R.: Agent for Participating in Multiple Online Auctions. ACM Transaction on Internet Technology 3(3), 1–32 (2003)
3. Chiang, J.S., Wu, P.L., Chiang, S.D., Chang, T.I., Chang, S.T., Wen, K.L.: Introduction to Grey System Theory. Gao-Li Publication, Taiwan (1998)
4. Chiou, H.K., Tzeng, G.H., Cheng, C.K., Liu, G.S.: Grey Prediction Model for Forecasting the Planning Material of Equipment Spare Parts in Navy of Taiwan. In: Proceedings of the World Automation Congress IEEE, vol. 17, pp. 315–320. IEEE, Los Alamitos (2004)
5. Deborah, L., Anthony, P., Ho, C.M.: Evaluating the Accuracy of Grey System Theory Against Time Series in Predicting Online Auction Closing Price. In: Proceedings of 2007 IEEE International Conference on Grey Systems and Intelligent Services, pp. 463–470. IEEE, Nanjing (2007)
6. Deborah, L., Anthony, P., Ho, C.M.: Predictor Agent for Online Auction. In: Proceeding of the The 2nd KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, pp. 27–28. Springer, Incheon (2008)
7. Deng, J., David, K.W.N.: Contrasting Grey System Theory to Probability and Fuzzy. ACM SIGICE Bulletin 20(3), 3–9 (1995)
8. Deng, J.: Control Problem of Grey System. System Control Letter 1(1), 288–294 (1982)
9. Klemperer, P.: Auction Theory: A Guide to the Literature. Journal of Economic Surveys 13(3), 227–286 (1999)
10. Laurene, F.: Fundamentals of Neural Networks (Architectures, Algorithms & Applications). Florida Latitude of Technology (1994)
11. Li, X.F., Liu, L., Wu, L.H., Zhang, Z.: Predicting The Final Price of Online Auction Items. In: Proceeding of Expert Systems with Application, pp. 542–550. Elsevier, Amsterdam (2006)
12. Lin, Y., Valencia, J.: Grey Analysis of Colombian Migration. In: Proceedings of 2007 IEEE International Conference on Grey Systems and Intelligent Services, pp. 7–12. IEEE, Nanjing (2007)
13. Lin, Y., Liu, S.: Historical Introduction to Grey Systems Theory. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, The Hague, The Netherlands, vol. 3, pp. 2403–2408 (2004)
14. Liu, S., Lin, Y.: Grey Information: Theory and Practical Application with 60 Figures. Springer, London (2006)
15. Taylor, J.W., Buizza, R.: Neural Network Load Forecasting With Weather Ensemble Predictions. Proceeding of IEEE Transaction On Power Systems 17(3), 626–632 (2002)
16. Wang, W., Hao, Y.H., Du, X.: Application of Grey System GM (1, N) Model to Predicting Spring Flow. In: Proceedings of 2007 IEEE International Conference on Grey Systems and Intelligent Services, pp. 448–452. IEEE, Nanjing (2007)
17. Wolfstetter, E.: Auctions: An Introduction. Journal of Economic Surveys 10, 367–420 (2002)

Feature Selection Using Mutual Information: An Experimental Study

Huawen Liu¹, Lei Liu¹, and Huijie Zhang^{1,2}

¹ College of Computer Science, Jilin University, Changchun 130012, China
Huaw.Liu@gmail.com, Liulei@jlu.edu.cn

² College of Computer, Northeast Normal University, Changchun 130021, China
Zhanghj167@nenu.edu.cn

Abstract. In real-world application, data is often represented by hundreds or thousands of features. Most of them, however, are redundant or irrelevant, and their existence may straightly lead to poor performance of learning algorithms. Hence, it is a compelling requisition for their practical applications to choose most salient features. Currently, a large number of feature selection methods using various strategies have been proposed. Among these methods, the mutual information ones have recently gained much more popularity. In this paper, a general criterion function for feature selector using mutual information is firstly introduced. This function can bring up-to-date selectors based on mutual information together under an unifying scheme. Then an experimental comparative study of eight typical filter mutual information based feature selection algorithms on thirty-three datasets is presented. We evaluate them from four essential aspects, and the experimental results show that none of these methods outperforms others significantly. Even so, the conditional mutual information feature selection algorithm dominates other methods on the whole, if training time is not a matter.

Keywords: Feature selection; mutual information; filter model.

1 Introduction

Along with the rapid accumulation of data, both the size and dimensionality of database are getting larger and larger. The straight misfortune to learning algorithms is that they can not effectively identify useful information from those overwhelming number of data. To alleviate this problem, two solutions are accessible. The first one is sampling technique, e.g., selective sampling [16], which picks less representative data to substitute the whole for learning. To some extent, this technique is very effective for databases with mass data. However, data is often characterized by a multitude of irrelevant or redundant features, which may also result in low efficiency of learning algorithms. Thus, feature selection (shortly, FS) is another direction to further improve learning efficiency by cutting the dimensionality of data down.

Feature selection algorithm (FSA) refers to the process of removing useless or insignificant features from the original space and retaining as more salient

features as possible. The preserved features must be competent for characterizing the main properties of data in the original feature space. FSAs may benefits learning algorithms at many aspects. For example, the computational cost of learning algorithm will be reduced and the prediction performance may be strengthen. In addition, the induced knowledge is more general to tolerate noises. Meanwhile, the “curse of dimensionality” problem can also be avoided [3]. During past years, many outstanding FSAs have been witnessed. Good surveys of FSA are available in literatures (e.g., [5,10,14,17]).

Generally, FSAs can be divided into three categories, i.e., supervised [10], unsupervised [7] and semi-supervised ones [25], according to wether the class labels are required. The former evaluates feature relevance by the correlation between features and class labels, while unsupervised one evaluates feature relevance by the capability of keeping certain properties of the data. Semi-supervised feature selection, however, utilizes both labeled and unlabeled data to validate the significance of features. From the view of feature evaluation manner, FSAs are roughly classified as *embedded*, *wrapper* and *filter* methods [5,17]. Embedded method means the feature selection stage is integrated into the process of training for a given learning algorithm. While wrappers choose those features with high priorities estimated using a specified learning algorithm itself as part of the evaluation function. As an example, Huang et al. [11] integrated a hybrid genetic algorithm into feature selection to achieve high global predictive accuracy as well as local search efficiency. Usually, wrapper methods can achieve optimal feature subsets and show better performance [17]. However, they require much more training time and are less general because they are highly coupled with specified learning algorithms.

Filter model, however, is independent learning algorithms. It evaluates feature individually according to a pre-specified criterion, and picks the best features out. For instance, RELIEF [13] is one of typical filter selection algorithms. Due to its computational efficiency, this model is very popular to high-dimension data. For filter model, evaluation criterion and search strategy are two important aspects [21]. Roughly speaking, there are four metrics to evaluate the goodness of feature subsets, i.e., distance, information, dependency and consistency measurements [17]. Among these metrics, the information one has received much concern recently. The reason is that it can exactly quantify the uncertainty of variable and express non-linear correlation between features [6]. Moreover, other three metrics are sensitive to the concrete values of the training data, which results in they are less robust and easily affected by noise or outlier data. Nowadays, a modest number of mutual information based feature selection algorithms (MIFSAs) have been developed, and more efficient and sophisticated approaches are still emerging. Unfortunately, no systematic study has been conducted on the reliability and effectiveness of these MIFSAs and the multitudinous algorithms will bewilder practitioners.

In this paper, we firstly introduce a general information measurement which brings other proposed criteria together, and then present the empirical comparison of MIFSAs in the classification issue. To achieve impartial results, three

classifiers have been used to validate the effectiveness and performance of eight MIFSA on thirty-three UCI benchmark datasets. To the best of our knowledge, we haven't noticed any similar work on this topic before. Another purpose of this paper is to give miners a guideline that there has no MIFSA which outperforms others in all situations and the good choice of feature selectors is determined by specific problems at hand.

The structure of the rest is organized as follows. Section 2 presents a general scheme of MIFSA. In section 3, we firstly introduce a general information criterion function for MIFSA, and then briefly discuss the relationship between it and other eight information criteria investigated in the state of the art. Experimental results conducted to compare them on four aspects are outlined in Section 4. Finally, conclusions and future works are given in the end.

2 Unified Scheme of MIFSA

Given a dataset $T=(\mathcal{D}, \mathcal{F}, C)$ with n instances represented by m features, where \mathcal{D} , \mathcal{F} and C are instances, features and class labels respectively. The task of classification is to tag instances with a label $c \in C$ in low probability of error $\varepsilon_{\mathcal{F}}(T)$. Theoretically, having more features implies more discriminative power in classification. However, many features are relevant to each other and they have no contribution to classification, except to degrade performance of classifiers. Thus, it is necessary to remove redundant and irrelevant features as more as possible, without losing information greatly. Generally, the task of feature selection in classification issue is formally defined as the process of selecting a minimum optimal feature subset $S \subseteq \mathcal{F}$ to characterize data \mathcal{D} and its classification error $\varepsilon_S(T)$ with labels C is approximately equal to the initial one $\varepsilon_{\mathcal{F}}(T)$.

In order to scale how much information is embodied in the feature subset F , a generalized criterion function $J(F)$ of mutual information (MI) is adopted. MI derived from information entropy is a nonparametric, nonlinear measurement of relevance of variable [6]. Let X and Y be two features with discrete values, and $p(x)$ denote the marginal density function of X . The uncertainty of X can be measured by *entropy* $H(X)$, where $H(X) = -\sum p(x) \log p(x)$. Additionally, the *mutual information* (MI) $I(X; Y)$ of X and Y , which quantifies how much information is shared between them, is defined as $I(X; Y) = \sum \sum p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$. If X and Y are highly related, $I(X; Y)$ will be very high. Otherwise, $I(X; Y) = 0$ implies these two features are totally unrelated or independent with each other.

MI is capable of quantifying the amount of information contained in a feature or a group of features. From the view of MI, classification means to minimize the uncertainty of predictions for the known observations represented by F . Thus, the purpose of MIFSA for classification is to achieve the highest possible value of $J(F)$ with the smallest possible size of F . For a specific dataset $T=(\mathcal{D}, \mathcal{F}, C)$, however, there has 2^m feature subsets $F \subseteq \mathcal{F}$. Hence, it is impossible to calculate $J(F)$ for all feature subsets F in a brute-force way, and this problem is NP-hard. To obtain an approximate optimal solution, many search strategies have been adopted in FSAs, such as Branch & Bound (BB), sequential forward selection

(SFS), sequential backward elimination (SBE) and oscillating search (OS). More details can be consulted to literature [21].

Due to its simple mechanism and high efficiency, sequential forward selection (SFS) is frequently taken as the subset search technique in most MIFSAs. The common assumption behind this search method is the monotonic property in picking up important features [17], that is, increasing the number of features will improve the performance of learning algorithms. Formally, if $F' \subseteq F''$, then $J(F') \leq J(F'')$. Hence, the problem of calculating $J(F)$ is now migrated into evaluating individual candidate features $J(f)$. More specifically, it begins with an empty set and adds at a time the candidate feature with the most positive influence on the criterion. It will be terminated when pre-specified stopping criteria is satisfied. For example, the number of selected features is larger than a threshold or $J(S)$ has not been improved as adding one more feature. To be more explicitly, the unified scheme of MIFSA is shown as follows.

Algorithm 1. The unified scheme of MIFSA.

Input: Training dataset $T = (\mathcal{D}, \mathcal{F}, C)$;

Output: An optimal feature subset S .

- 1). Initialize candidate and selected feature subsets $F=\mathcal{F}$, $S = \emptyset$;
 - 2). **Repeat**
 - 3). Calculate $J(f)$ using MI metric for each candidate feature $f \in F$;
 - 4). $S=S \cup \{f\}$ and $F=F \setminus \{f\}$, where $J(f)$ is the largest one;
 - 5). **Until** pre-specified stopping criteria is satisfied;
 - 6). Return S as the selected feature subset;
-

3 Information Measurements

During past years, many MIFSAs have been proposed to select features. Although they adopt different criterion functions, the distinctness between them lies in the specific form of information criterion function $J(f)$. Even so, their common goal is to minimize redundancy and maximize relevance between candidate and selected features. Without loss of generality, for the candidate feature $f \in F$, its criterion function is

$$J(f) = \alpha \cdot g(C, f, S) - \delta, \quad (1)$$

where $g(C, f, S)$ is information function about the candidate feature f , the labels C and the selected subset S . This function is mainly used to measure the relevance between f and C under the context of S . Usually, $g(C, f, S)$ takes the form of mutual information $I(C; f)$ or its conditional one $I(C; f|S)$. α is a coefficient to regulate relative importance of $g(C, f, S)$. While δ is a deviation operator to penalize the redundancy brought by f . Similarly, δ takes as a function formalization of MI among f , C and individual feature $s \in S$.

Here we choose eight typical information measurements proposed in literatures to make a comparison. In the following, we will briefly review these information measurements and discuss the relationship between the general criterion function

and them one by one. For the sake of convenience, $S \subseteq \mathcal{F}$ and $F \subseteq \mathcal{F}$ represent selected and candidate features respectively, while $f \in F$ and $s \in S$ are candidate and selected features.

MI. Mutual information, which is also known as information gain or best individual feature [12], perhaps is the most naïve measurement. For the candidate feature f , its criterion function of MI is $J(f) = I(C; f) = H(f) - H(f|C)$, where $H(f)$ is information entropy of f and $H(f|C)$ is its conditional one with respect to C . That is to say, $\alpha = 1$, $g(C, f, S) = I(C; f)$ and $\delta = 0$ in Eq. (1). This method chooses the best individual features out at feature selection procedure [12]. It firstly evaluates all candidate features individually according to the criterion function $J(f)$, and then sorts them in descending order in terms of $J(f)$. After that, the best k features are picked out to take the place of the whole features.

MIFS. For the MI criterion, one may observe that its metric $J(f)$ does not take into consideration redundancy and relevance between f and s . This, however, will bring redundancy between f and S . Moreover, the individually selected features may not be an optimal solution when they are grouped into the whole [17]. To cope with this issue, Battiti [1] harnessed the relevance (i.e., $I(f; s)$) between f and s to penalize $I(C; f)$, and his criterion function is $J(f) = I(C; f) - \beta \sum_{s \in S} I(f; s)$, where $0.5 \leq \beta \leq 1$. Since the value of the parameter β will perform great effect on the results of MIFSA and its appropriate value is hard to be obtained, Peng et al. [19] directly assigned $1/|S|$ to β and then took this function as the evaluation criterion in their wrapper model, where $|S|$ is the number of selected features.

MIFS-U. In MIFS, the penalized factor (i.e., the second component) can not exactly represent the incremental information of f when S is known. Additionally, it does not involve the relevance between s and C . Thus, Kwak and Choi [15] utilized coefficient of uncertainty $CU(f, s)$ to illustrate the relevance between f and s , where $CU(f, s) = I(f; s)/H(s)$. In their method, the penalty factor is revised as $\beta \sum_{s \in S} CU(f, s) \cdot I(C; s)$. That is to say, their criterion function is $J(f) = I(C; f) - \beta \sum_{s \in S} CU(f, s) \cdot I(C; s)$. Similarly, Huang et al. [11] set the parameter $\beta = 1/|S|$, and then trained the selector with a genetic algorithm to achieve optimal results.

mMIFS-U. As mentioned above, it is a troublesome issue to regulate an appropriate value to β in MIFS and MIFS-U. To immigrate this harassment, Novovičová et al. [18] recently developed a modified version of MIFS-U, which is called mMIFS-U. The difference between them is that maximum operation has been adopted in mMIFS-U, rather than the sum operation in MIFS-U. That is, the criterion function of mMIFS-U is $J(f) = I(C; f) - \max_{s \in S} (CU(f, s) \cdot I(C; s))$. The advantage of this metric is that it is free from the parameter β .

SU. Rather than regulating the deviation δ to lessen the redundancy between f and S , Yu and Liu [24] resorted to the coefficient α of $I(C; f)$ to dominate the impact of relevance. In their algorithm, a metric called *symmetrical uncertainty*

has been employed to portray the correlation between features. For any feature f , its symmetrical uncertainty with C is defined as $SU(C, f) = 2I(C; f)/(H(f) + H(C))$, where $H(f)$ is the information entropy of f . Indeed, this correlation is their criterion function $J(f)$ to choose significant features. Additionally, they removed redundant features by virtue of approximate Markov Blanket technique.

DDC. In SU, the redundancy between f and S has not been involved explicitly. Qu et al. [20] argued that this may provide false or incomplete information to the relevance. Hence, they introduced the notation of *decision dependent correlation* $Q_C(f, s)$ to measure the relevant degree between f and s , where $Q_C(f, s) = [I(C; f) + I(C; s) - I(C; f, s)]/H(C)$. This correlation, together with $I(C; f)$, consists of their selection criterion. In their method, a candidate feature f is good if $I(C; f)$ is maximal, while $Q_C(f, s)$ is minimal for any $s \in S$. This criterion, however, is similar to $J(f) = I(C, f)/\sum Q_C(f, s)$.

CMIM. *Conditional mutual information*, denoted as $I(C; f|S) = H(C|S) - H(C|f, S)$, refers to the amount of common information between C and f when S is known. The larger value of $I(C; f|S)$ is, the more information can be brought by f about C which has not yet been contained in S . Unlike aforementioned criteria, the conditional MI maximization criterion exploits $I(C; f|S)$ to select those features correlated to ones already picked, for they do not carry any additional information for the classification. Unfortunately, the computational cost of $I(C; f|S)$ is high. To circumvent this problem, Fleuret [9] and Wang et al. [22] substituted S with a single feature $s \in S$. They pointed out that the candidate feature f is good only if it carries information about C , which has not been caught by any feature s already picked, that is, $I(C; f|s)$ is largest for any $s \in S$. More specifically, the criterion function is $J(f) = \min_{s \in S} I(C; f|s)$.

CR. In an analogical vein, Bell and Wang [2] defined a concept of *conditional variable relevance* $r(X; Y|Z)$ to describe the relative reduction of uncertainty of Y when X and Z are known, where X, Y and Z are variables. Contrastively, the unconditional one, denoted as $r(X; Y)$, does not take into consideration conditional information, and this is their metric in selecting feature, i.e., $J(f) = r(C; S, f) = I(C; S, f)/H(S, f)$. This means that the feature with the most incremental information will be firstly culled. As a matter of fact, $r(C; S, f)$ is another expression of $I(C; f|S)$ in terms of its definition.

4 Experimental Evaluation

4.1 Benchmark Datasets

To compare these MIFSA roundly, thirty-three benchmark datasets with different types and sizes were adopted in our experiments. These datasets are all available from the UCI Machine Learning Repository [4], and widely used to evaluate the performance of learning and selection algorithms. Table 1 summarizes general information. These datasets comprise a diverse mixture of feature

dimensionality, where the maximal one is up to 1558, and their sizes range from 32 to 10108. Hence, they can provide a comprehensive test to suit for MIFSA under different conditions.

Since some features are too trivial to suitable for classification, we omitted them before our experiments. For example, in the *Cylinder-bands* dataset, *timestamp*, *cylinder_number* and *customer* were excluded. Analogically, *name* in *Sponge* and *Flags*, *Instance_name* in *Splice*, *instance* in *Promoters*, *who* in *Kdd-internet-usage*, and *LRS_name* and *LRS_class* in *Spectrometer*, were all out of consideration. Moreover, the last feature in each dataset was taken as the classification label, except *Spectrometer* and *Kdd-internet-usage*, where the classification labels were *ID_Type* and *years-on-internet*, respectively. Additionally, missing values in datasets were replaced with the most frequent values (or means) for nominal (or numeric) features, and continuous features were discretized into nominal ones by the minimum description length method [8].

Table 1. The descriptions of datasets in our experiments

No.	Datasets	inst.	feat.	clas.	No.	Datasets	inst.	feat.	clas.	No.	Datasets	inst.	feat.	clas.
1	internet-ad	3279	1558	2	12	synth. contr.	600	60	6	23	musk clean2	6598	166	2
2	anneal	898	38	6	13	Kr-vs-kp	3196	36	2	24	optdigits	5620	64	10
3	arrhythmia	452	279	16	14	lung cancer	32	56	3	25	promoters	106	57	2
4	audiology	226	69	24	15	lymph	148	18	4	26	sonar	208	60	2
5	cylinder-bands	540	36	2	16	mfeat-factors	2000	216	10	27	soybean	683	35	19
6	dermatology	366	34	6	17	mfeat-fourier	2000	76	10	28	spambase	4601	57	2
7	flags	194	28	8	18	mfeat-karhunen	2000	64	10	29	spectf-total	349	44	2
8	hypothyroid	3772	29	4	19	mfeat-pixel	2000	240	10	30	spectrometer	531	100	4
9	ionosphere	355	34	2	20	mfeat-zernike	2000	47	10	31	splice	3190	60	3
10	internet usage	10108	70	5	21	mushroom	8124	22	2	32	sponge	76	44	3
11	ipums97-small	7019	60	9	22	musk clean1	476	166	2	33	waveform	5000	40	3

4.2 Experimental Settings

For the purpose of fair results, we employed the same stopping condition for MIFSA in our experiments, that is, the selection process will be terminated if $I(C; S)/I(C; \mathcal{F}) \geq 0.99$. This signals that information embodied in S about C is approximately equal to those of original feature space \mathcal{F} . In addition, the same quantity of features, which equals to the least number, was chosen by each selector for each dataset and these selected features were arranged in a descending order in terms of their priorities.

After insignificant features have been removed, datasets will be fed into classifiers to obtain classification errors. Here, we choose three classical and popular classifiers, i.e., Naive Bayes, 1-NN and C4.5. These classifiers stand for quite different learning approaches and are relatively fast in learning.

During the verification procedure, three ten-fold cross validations had been adopted for each algorithm-dataset combination, and the average values are the desirable results. To determine whether the difference between two classifiers is significant or not, paired *t*-test between selected features and original ones had been performed to each classifier. Throughout this paper, difference is considered significantly different if its *p*-value is less than 0.05 (i.e., confidence level greater than 95%). All experiments were carried out on a Pentium IV 2.8 GHz and 512 MB main memory, and the experimental platform is the Weka toolkit [23].

4.3 Experimental Results

Number of Selected Features and Consumed Time. The number of selected features is one of major aspects to measure whether a MIFSA is effective or not. The fewer number of features induced by selector, the more effective it is. Table 2 shows the number of features picked out by each selector. Correspondingly, a comparison of Win/Tie/Loss on the amount of selected features is presented in the top-right triangle of Table 3. For example, the entry “3/7/23” of the first row and the fifth column in Table 3 denotes that the MI selector wins over the CMIM one on three datasets, while loses twenty three cases.

Table 2. The number of selected features by eight MIFSA

No	MI	M1	M2	M3	CM	SU	DDC	CR	Ave	No	MI	M1	M2	M3	CM	SU	DDC	CR	Ave
1	572	531	594	301	244	604	1236	1250	667	18	6	6	6	6	6	7	6	6	
2	10	9	10	9	10	9	37	15	14	19	30	16	23	10	10	33	14	34	
3	60	63	63	42	33	64	247	110	85	20	11	9	11	8	10	11	11	10	
4	15	15	15	14	15	15	16	47	19	21	4	4	4	4	5	3	4	3	
5	20	32	19	32	20	20	34	20	25	22	63	49	60	36	31	91	75	102	
6	18	18	18	11	10	19	8	24	16	23	34	34	34	30	20	47	18	52	
7	8	7	9	8	9	9	9	11	9	24	9	7	7	7	7	9	8	20	
8	14	14	14	22	7	12	28	10	15	25	5	5	5	5	5	5	4	5	
9	11	10	10	11	10	14	28	20	14	26	17	15	17	15	16	18	54	17	
10	9	14	9	8	7	14	5	25	11	27	22	22	22	20	20	25	10	30	
11	6	16	6	3	3	15	3	26	10	28	29	36	31	29	26	37	51	43	
12	12	10	13	12	6	16	11	20	13	29	30	31	30	31	28	31	42	31	
13	34	30	34	33	32	34	28	34	32	30	64	46	63	65	49	70	60	83	
14	14	11	12	10	3	14	4	22	11	31	9	9	9	9	9	8	9	9	
15	7	9	7	7	8	11	11	14	9	32	5	3	4	4	2	4	2	6	
16	6	5	6	6	6	8	9	11	7	33	11	9	11	11	9	11	30	11	
17	10	9	9	10	9	10	11	11	10	Ave	36	33	36	25	21	39	64	65	

¹M1, M2, M3, CM denote MIFS, MIFS-U, mMIFS-U and CMIM, respectively; ‘Ave’ means Average.

On the ground of the results in Table 2 and 3, we can observe that there has no selector that outweighs others on all datasets. Even so, the CMIM selector outperforms other selectors in most cases, and the remainders are mMIFS-U, MIFS, MIFS-U, MI, DDC, SU and CR, which are arranged in descending order from the view of the number of selected features.

Table 3. A comparison of W/T/L on the selected features and consumed time

	MI	MIFS	MIFS-U	mMIFS-U	CMIM	SU	DDC	CR
MI	-	7/9/17	5/19/9	4/12/17	3/7/23	18/11/4	17/2/14	23/8/2
MIFS	3/0/30	-	14/13/6	10/9/14	8/8/17	22/7/4	21/1/11	26/4/3
MIFS-U	1/0/32	10/0/23	-	6/9/18	3/10/20	21/9/3	17/3/13	24/7/2
mMIFS-U	3/0/30	17/1/15	22/1/10	-	7/8/18	22/8/3	20/2/11	25/5/3
CMIM	0/0/33	5/0/28	5/0/28	3/0/30	-	25/6/2	22/3/8	28/4/1
SU	15/2/16	30/0/3	31/0/2	28/0/5	33/0/0	-	15/3/15	22/9/2
DDC	0/0/33	3/0/30	4/0/29	3/0/30	10/0/23	1/0/32	-	20/2/11
CR	16/0/17	30/0/3	30/1/2	27/0/6	33/0/0	14/2/17	33/0/0	-

Since the feature selection procedure currently works under off-line manner in many situations, computational cost, however, is less important. Due to space limitations, here we only list the W/T/L comparison on consumed time as the bottom-left triangle of Table 3. In our experiments, we found that the DDC took much more time than others, while the time cost of MI, SU and CR is far less than others and slightly different with each other. For example, the DDC selector costed several hours to complete the selection operation on the

Internet-ad dataset, whereas the MI selector took thirty seconds. Similarly, one can summarize that the priority order of these selectors is MI, SU, CR, mMIFS-U, MIFS, MIFS-U, CMIM and DDC.

Mean classification performance. To make a comparison on classification performance from the whole, we averaged classification accuracies of three classifiers on each dataset. The results are given in Table 4, where the ‘Total’ column denotes the average values of classification accuracies of these three classifiers on the original datasets. Notation ‘◦’ (or ‘•’) is used to illustrate that the classification capability with current feature selector is significantly worse (or better) at 0.05 level than those without using any selector.

Table 4 tells us that the performances of CMIM and MIFS selectors seem better than others from the point of having largest values. They have fourteen and eleven largest values over thirty three datasets, respectively. However, it is not a good idea to choose DDC and CR to select features, for they degrade significantly classification performance in most cases. One may notice that the accuracies of MIFSAs are lower than those without using selectors in many cases. However, this does not indicate that MIFSAs would deteriorate the performance of classifier or inferior to other filter methods, because the selectors chose the least important features in our experiments.

For the purpose of intuitively seizing which one performs the best on mean performance, we also compare them with each other using the W/T/L strategy and results are provided in Table 5. In the light of the results, CMIM slightly outperforms MIFS on the whole, and both of them are superior to other methods in

Table 4. A comparison of mean classification performance of MIFSAs

	Total	MI	MIFS	MIFS-U	mMIFS-U	CMIM	SU	DDC	CR
1	97.02	98.60	96.56	96.60	97.06	96.90	96.27	92.75 ◦	95.80
2	92.72	92.92	92.86	92.92	92.92	92.92	93.35	85.78 ◦	92.99
3	71.18	72.00	72.65	72.23	72.66	73.06	72.10	68.48	59.95 ◦
4	74.77	72.01	72.01	72.01	71.51	74.16	72.01	67.36	52.84 ◦
5	73.83	74.13	72.66	74.39	73.51	74.07	74.13	71.24 ◦	74.39
6	95.63	75.28 ◦	79.25	79.25	85.57	93.74	75.28 ◦	70.30 ◦	66.23 ◦
7	59.88	59.77	57.38	59.77	59.77	57.03	57.79	55.76 ◦	59.36
8	98.57	98.73	98.72	98.72	98.73	98.63	98.72	98.18 ◦	98.81
9	90.98	91.21	91.87	91.71	91.62	92.10	91.65	89.68	89.49
10	41.44	41.77	42.44	41.77	41.78	40.56	41.94	38.68 ◦	42.11
11	71.09	71.42	72.04	72.26	72.26	71.88	68.18	71.88	68.18
12	96.22	78.97 ◦	83.29	78.22 ◦	77.98 ◦	90.13	78.21 ◦	77.46 ◦	68.19 ◦
13	92.43	92.45	94.24 •	92.85	93.02	93.03	92.45	94.56 •	92.85
14	48.05	54.63	59.82 •	54.63	54.63	56.67 •	60.37 •	50.65	52.50
15	80.54	75.34	76.61	77.32	77.32 ◦	77.32 ◦	77.32 ◦	77.49 ◦	77.03 ◦
16	90.52	70.74 ◦	81.34	75.89	79.25	78.78	75.89 ◦	65.45 ◦	68.42 ◦
17	74.88	75.56	75.55	75.56	75.56	75.56	75.56	65.07 ◦	74.67
18	86.17	76.00 ◦	77.01	76.00	77.01 ◦	77.01 ◦	77.01 ◦	70.93 ◦	77.01 ◦
19	89.42	53.65 ◦	76.81	60.74	79.67	81.48	53.65 ◦	47.44 ◦	52.51 ◦
20	69.78	54.37 ◦	60.98	58.58	59.98	60.75	54.37 ◦	52.08 ◦	62.51
21	98.52	99.20	98.46	98.87	98.87	99.57 •	98.87	98.43	98.99
22	87.75	86.55	88.56	87.08	87.72	86.95	84.52	83.84	78.06 ◦
23	94.24	94.02	93.32	94.09	93.80	92.76 ◦	93.73	92.88 ◦	92.57 ◦
24	88.01	74.53	76.34	74.53	76.34	77.31	70.79	61.52 ◦	40.30 ◦
25	85.06	87.89	87.89	87.89	87.89	87.89	87.89	76.09 ◦	87.89
26	81.76	79.07	80.75	79.07	80.75	82.38	79.19	51.81 ◦	75.39
27	92.04	82.60	85.74	85.74	87.81	84.96	81.49	62.97 ◦	53.98 ◦
28	91.65	90.98	92.09	91.29	91.57	91.94	91.22	86.94 ◦	90.56
29	84.15	82.47 ◦	83.33	82.47 ◦	83.30	84.04	82.47 ◦	82.02 ◦	82.56 ◦
30	62.84	60.32	64.28	60.13	61.76	62.70	59.92 ◦	62.39	60.59
31	88.55	91.72	91.72	91.72	91.72	91.72	91.72	69.12 ◦	91.72
32	92.38	91.49	93.71	91.49	91.49	92.56	92.18	92.58	92.04
33	76.6	69.85 ◦	76.76	74.18	74.18	76.64	75.35	66.70 ◦	75.35

²◦ (or ‘•’) indicates the performance with selector is significant worse (or better) than those without selectors.

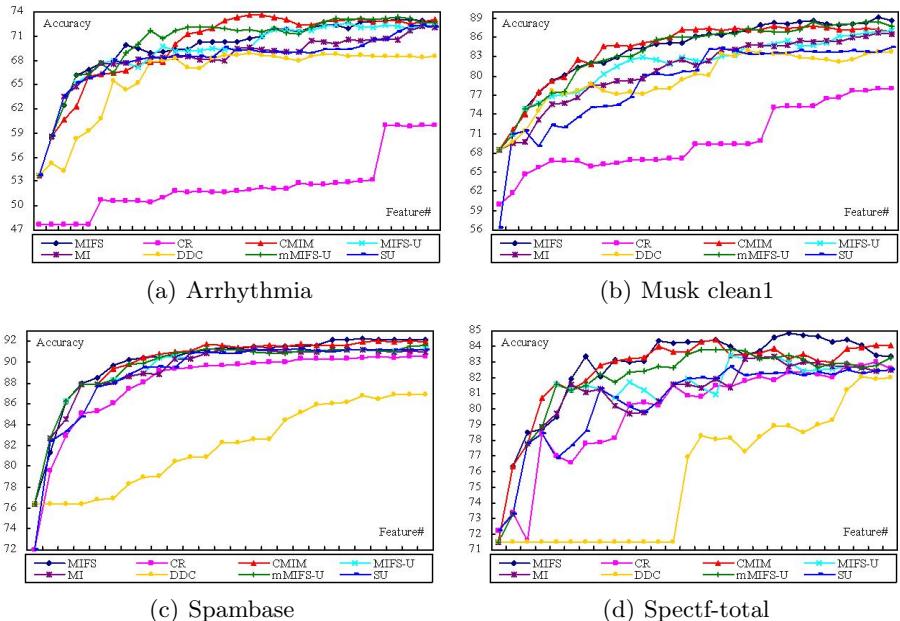
³The bold value is the highest one among eight feature selectors for the same database.

Table 5. A comparison of W/T/L on average performance in classification

	MI	MIFS	MIFS-U	mMIFS-U	CMIM	SU	DDC	CR
MI	-	7/4/22	4/14/15	5/8/20	5/4/24	11/10/12	28/0/5	20/2/11
MIFS	22/4/7	-	18/7/8	14/6/13	15/4/14	20/6/7	31/0/2	24/3/6
MIFS-U	15/14/4	8/7/18	-	5/11/17	8/5/20	17/8/8	30/0/3	19/4/10
mMIFS-U	20/8/5	13/6/14	17/11/5	-	9/6/18	18/5/10	30/0/3	22/3/8
CMIM	24/4/5	14/4/15	20/5/8	18/6/9	-	21/4/8	29/1/3	24/3/6
SU	12/10/11	7/6/20	8/8/17	10/5/18	8/4/21	-	29/0/4	19/5/9
DDC	5/0/28	2/0/31	3/0/30	3/0/30	3/1/29	4/0/29	-	14/0/19
CR	11/2/20	6/3/24	10/4/19	8/3/22	6/3/24	9/5/19	19/0/14	-

most cases. Contrastively, the priority order, from the viewpoint of classification performance, is CMIM, MIFS, mMIFS-U, MIFS-U, MI, SU, CR and DDC.

Performance of selected feature. To characterize the selected features induced by different selectors, we conducted experiments on four datasets (i.e., *Arrhythmia*, *Musk clean1*, *Spectf-total* and *Spambase*) whose least amount of selected features is larger than twenty. The experimental mode is the same with above, and the results are shown in Figure 1, where the accurate rate is the average value of three classifiers over three times.

**Fig. 1.** Accuracy vs. different numbers of selected features on four UCI datasets

As illustrated in Figure 1, the CR selector is worse than others in the *Arrhythmia* and *Musk clean1* datasets, while DDC loses its superiority and becomes the last one in the later two datasets. This also means that these two selectors have relatively poor performance. For the rest selectors, CMIM, mMIFS-U and MIFS

work well and outperform others in these four datasets. If these selectors are grouped into three levels, then CMIM, mMIFS-U and MIFS belong to the first level, while CR and DDC are the last one.

5 Conclusions

This paper firstly provides an unified framework of MIFSA, and then introduce a general information criterion for MIFSA. After that, the relationship between this criterion and other information metrics in the state of the art of filter MIFSA has been discussed. Moreover, an experimental comparison of eight popular filter MIFSA has been conducted on thirty-three benchmark datasets. The experimental results signal that there is no single approach outperforming others for every scenario. The rationale is that each MIFSA has its characteristics and its working way is determined by many aspects. Even so, the experimental results give us several good indications. For example, CMIM has the most powerful and stable performance over other methods we considered on the whole. Meanwhile, some methods such as DDC and CR are relatively poor. If training time is not a problem, however, CMIM, mMIFS-U and MIFS are perhaps better choices. Otherwise, SU and MI seem to be good candidates.

From evaluation criterion, one may notice that information entropy in up-to-date methods is estimated on the whole sampling space, which is determined once it has been given. This means that their values are invariable throughout the selection procedure. However, this can not exactly represent the relevance between features when the selection procedure continues to work, because for instances which can be identified by selected features, any candidate feature is redundant or irrelevant. Moreover, if a feature is important, it would be selected by these MIFSA several times simultaneously. Hence, our future work will be conducted on solving these issues by using other strategies, such as ensemble or boosting, to choose features in virtue of their own characteristics.

Acknowledgements

This work is supported by the Doctor Point Founds of Educational Department (20060183044) and Science Foundation for Young Teachers of Northeast Normal University(20081003).

References

1. Battiti, R.: Using Mutual Information for Selecting Features in Supervised Neural Net Learning. *IEEE Transactions on Neural Networks* 5(4), 537–550 (1994)
2. Bell, D.A., Wang, H.: A Formalism for Relevance and Its Application in Feature Subset Selection. *Machine Learning* 41, 175–195 (2000)
3. Bellman, R.: Adaptive Control Processes: A Guided Tour. Princeton University Press, Princeton (1961)
4. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases (1998), <http://www.ics.uci.edu/~mlearn/MLRepository.html>

5. Blum, A.L., Langley, P.: Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence* 97, 245–271 (1997)
6. Cover, T.M., Thomas, J.A.: *Elements of Information Theory*. Wiley, NY (1991)
7. Dy, J.G., Brodley, C.E.: Feature selection for unsupervised learning. *Journal of Machine Learning Research* 5, 845–889 (2004)
8. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous valued attributes for classification learning. In: *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 1022–1027 (1993)
9. Fleuret, F.: Fast Binary Feature Selection with Conditional Mutual Information. *Journal of Machine Learning Research* 5, 1531–1555 (2004)
10. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
11. Huang, J., Cai, Y., Xu, X.: A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters* 28, 1825–1844 (2007)
12. Jain, A.K., Duin, R., Mao, J.: Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(1), 4–37 (2000)
13. Kira, K., Rendell, L.: A practical approach to feature selection. In: *Proceedings of the 9th International Conference on Machine Learning*, pp. 249–256 (1992)
14. Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artificial Intelligence* 97(1-2), 273–324 (1997)
15. Kwak, N., Choi, C.-H.: Input feature selection by mutual information based on Parzen window. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(12), 1667–1671 (2002)
16. Lindenbaum, M., Markovitch, S., Rusakov, D.: Selective Sampling for Nearest Neighbor Classifiers. *Machine Learning* 54, 125–152 (2004)
17. Liu, H., Yu, L.: Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Transactions on Knowledge and Data Engineering* 17(4), 491–502 (2005)
18. Novovičová, J., Somol, P., Haindl, M., Pudil, P.: Conditional Mutual Information Based Feature Selection for Classification Task. In: *Proc. of the 12th Iberoamerican Congress on Pattern Recognition*, Valparaiso, Chile, pp. 417–426 (2007)
19. Peng, H., Long, F., Ding, C.: Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(8), 1226–1238 (2005)
20. Qu, G., Hariri, S., Yousif, M.: A New Dependency and Correlation Analysis for Features. *IEEE Transactions on Knowledge and Data Engineering* 17(9), 1199–1207 (2005)
21. Somol, P., Novovičová, J., Pudil, P.: Notes on The Evolution of Feature Selection Methodology. *Kybernetika* 43(5), 713–730 (2007)
22. Wang, G., Lochovsky, F.H., Yang, Q.: Feature Selection with Conditional Mutual Information MaxiMin in Text Categorization. In: *Proceedings of the 13th ACM CIKM 2004*, Washington, USA, pp. 342–349 (2004)
23. Witten, I.H., Frank, E.: *Data Mining - Practical Machine Learning Tools and Techniques with JAVA Implementations*, 2nd edn. Morgan Kaufmann, San Francisco (2005)
24. Yu, L., Liu, H.: Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning Research* 5, 1205–1224 (2004)
25. Zhao, Z., Liu, H.: Semi-supervised feature selection via spectral analysis. In: *Proceedings of the 7th SIAM International Conference on Data Mining*, Minneapolis, MN, pp. 1151–1158 (2007)

Finding Orthogonal Arrays Using Satisfiability Checkers and Symmetry Breaking Constraints*

Feifei Ma^{1,2} and Jian Zhang¹

¹ State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences
² Graduate University, Chinese Academy of Sciences
`{maff,zj}@ios.ac.cn`

Abstract. Orthogonal arrays are very important combinatorial objects which can be used in software testing and other areas. Mathematical methods for constructing such arrays have been studied extensively in the past decades. In contrast, computer search techniques, in particular exhaustive search methods, are rarely used to solve the problem. In this paper, we present an algorithm which finds orthogonal arrays of given sizes or shows their non-existence. The algorithm is essentially a back-track search procedure, but enhanced with some novel symmetry breaking (isomorphism elimination) techniques. The orthogonal array is generated column by column, and the constraints are checked by an efficient SAT solver or pseudo-Boolean constraint solver. We have implemented a tool called BOAS (Backtrack-style OA Searcher) using MiniSat and PBS. Experimental results show that our tool can find many orthogonal arrays quickly, especially those with strength higher than 2.

1 Introduction

The concept of **orthogonal arrays**, since introduced by C. R. Rao (1947), plays an important role in factorial designs in which each treatment is a combination of factors at different levels. Roughly speaking, an orthogonal array (OA) of strength t is an array with the property that all the ordered combinations of t symbols from different columns occur equally often in rows. As a combinatorial design with beautiful balancing property, OA has long been the interest of mathematicians. In addition, OAs are also frequently used in industrial experiments for quality and productivity improvement.

In recent years, OAs have also attracted the attention of researchers and engineers in software testing [3]. The Orthogonal Array Testing Strategy (OATS), a technique employing OA to represent uniformly distributed variable combinations, is very useful for integration testing of software components. It has been adopted by big companies like AT&T [12] and Motorola [9].

There are many mathematical results about OAs, either dealing with their construction or proving their non-existence given some parameters. However,

* This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. 60673044 and 60633010.

these mathematical results do not cover all cases. Sometimes we can only resort to computer search. So far, computer search methods for finding OAs have not received much attention. In this paper, we propose an exhaustive search method for finding OAs of given sizes. It integrates an efficient solver for the Boolean SATisfiability problem (SAT solver) and a solver for pseudo-Boolean constraints. In addition, it employs several novel symmetry breaking (isomorphism elimination) techniques. Experimental results show that the algorithm performs quite well in many cases, especially for finding OAs with high strengths.

The paper is organized as follows. In Section 2, we give a brief introduction to OAs. Then the search algorithm is presented in Section 3. In Section 4, we propose several heuristics for eliminating isomorphism. In Section 5, we give some experimental results. Finally, we discuss related works and possible improvements to our approach. Due to space limitation, we omit the detailed proofs for theorems and propositions. They can be found in our technical report [11].

2 Orthogonal Arrays

An **orthogonal array** (OA) of **run size** N , **factor number** k , **strength** t can be denoted by $\text{OA}(N, s_1, s_2, \dots, s_k, t)$. It is an $N \times k$ matrix satisfying:

1. There are exactly s_i symbols appearing in each column i , $1 \leq i \leq k$.
2. In every $N \times t$ sub-array, each ordered combination of symbols from the t columns appears equally often in rows.

We refer to s_i as the **level** of factor i . When the levels of an OA are not all equal, we also call it a **mixed orthogonal array (MOA)**. By combining equal entries in $\{s_i\}$, we may represent an $\text{OA}(N, s_1, s_2, \dots, s_k, t)$ in the shortened form $\text{OA}(N, s_{i_1}^{a_1} \cdot s_{i_2}^{a_2} \dots, t)$, where the exponents a_1, a_2, \dots indicate the number of factors at level s_{i_1}, s_{i_2}, \dots , respectively. For instance, an $\text{OA}(16, 4, 2, 2, 2, 3)$ can also be written as $\text{MOA}(16, 4, 2^3, 3)$. In the following sections, we use the terms “run” and “row”, “factor” and “column” interchangeably.

The symbols in an OA can be chosen arbitrarily, for example, a, b, c, \dots But conventionally, we often use the natural numbers $0, 1, 2, \dots$ A simple example is given in Figure 1. It is an instance of $\text{OA}(8, 2^4, 3)$. In each 8×3 sub-array, each ordered combination of symbols (e.g., $\langle 1, 0, 1 \rangle$) occurs exactly once.

OAs could be constructed by various mathematical means, e.g., Hadamard construction, juxtaposition, splitting, zero-sum, etc. There are also some bounds of parameters concerning the existence, the most famous among those being Rao’s generalized bound. For more details, one could refer to [5] or [7]. OAs of strength 2 have been studied extensively mainly through such methods. Brouwer et al. [4] discussed OAs of strength 3 and small run sizes in a similar way. As for OAs of higher strengths, there seem to be much fewer results.

3 The Search Procedure

We apply exhaustive search techniques to find (M)OAs of given sizes. The framework of our approach is described in this section.

3.1 The Basic Procedure

The approach is based on backtracking search, finding the orthogonal array column by column. It can be described as a recursive procedure like the following.

```
bool Colsch(j){
    cons = CONS_GEN(j, oaSol);
    while(true){
        column = SOLVE(cons);
        if(column == null) return FALSE;
        APPEND(column, oaSol);
        if(j == k) return TRUE;
        if(Colsch(j+1)) return TRUE;
        add NEGATE(column) to cons;
    }
}
```

Suppose we are processing column j and represent the obtained partial solution by `oaSol`. The function `CONS_GEN` obtains some constraints (denoted by `cons` in the pseudo-code) which are both necessary and sufficient for the current column to be consistent with the OA definition. Then the function `SOLVE` resorts to a pseudo-Boolean constraint solver or a SAT solver to find a solution (denoted by `column`). If there is no solution for the current column, the algorithm has to backtrack to the previous column and try another solution; otherwise the recently found column is added to the partial array `oaSol` by the function `APPEND`. When all the k columns are generated, a solution is obtained and the function returns `TRUE`. If the array is not completed, the procedure is executed recursively.

Our algorithm integrates a satisfiability checker as a search engine for solving part of the problem. The checker might be called a number of times, to find different solutions to a column. In the pseudo-code, the negation of the current solution, obtained by the function `NEGATE`, is added to the input file `cons` to guide the checker to find new solutions to the column.

One might wonder why we do not ask the satisfiability checker to do the whole search. The reason is that if an OA specification is directly encoded into SAT, there would be too many variables and clauses. It is difficult to describe the unique distribution of different symbol combinations in every t columns. Of course, if each symbol combination appears exactly once, there is an easy way. For example, k mutually orthogonal Latin squares of order N are equivalent to $\text{OA}(N^2, N^{k+2}, 2)$ and SAT solvers have been applied directly to the problem in the literature [2]. Efficiency is another issue. In Section 3 of reference [17], a direct SAT encoding of covering arrays is discussed, and it is reported that much time is needed to solve some small cases even after symmetry breaking.

In the next subsection, we shall discuss a trick which allows us to determine the first t columns of `oaSol`. Thus the search procedure begins at column $t + 1$, and the first call of the recursive function is `Colsch(t+1)`.

3.2 Preprocessing

Without loss of generality, we can assume in an OA problem $(N, s_1, s_2, \dots, s_k, t)$, the levels s_1, s_2, \dots, s_k are sorted in non-increasing order. Noticing the fact that in the first t columns all combinations of s_1, \dots, s_t symbols should appear $\lambda = \frac{N}{s_1 \times \dots \times s_t}$ times, we can generate the first t columns directly by enumerating all possibilities, each of which λ times. We call the generated partial array **init-block**. For example, the 8×3 sub-array formed by the first three columns of Figure 1 is an init-block.

3.3 Constraint Generation

Assume that we have constructed m columns ($m \geq t$). How can we generate the constraints for column $m + 1$? Our method is based on an observation in [7]:

Remark 1. An OA($N, s_1, s_2, \dots, s_k, t$) is also an OA($N, s_1, s_2, \dots, s_k, t - 1$).

If we extract $t - 1$ columns (denoted by $C_{i_1}, C_{i_2}, \dots, C_{i_{t-1}}$) from the matrix and partition the row numbers by the row vectors, i.e. putting the row numbers into the same set if the row vectors are identical, we can get $s_{i_1} \times \dots \times s_{i_{t-1}}$ mutually exclusive sets of equal size. We call each set of the partition a **p-set** induced by the sub-array.

Example 1. Consider the OA in Figure 1. The state of p-set stack after the init-block is constructed is shown in Figure 2. For each 8×2 sub-array in the init-block, there are four p-sets induced. More specifically, the p-set $\{4, 8\}$ is induced by the sub-array of column 2 and column 3 because row 4 and row 8 in the sub-array share the same row vector $\langle 1, 1 \rangle$, and so on. Similarly, for column 1 and column 2, if we examine the rows of the 8×2 sub-matrix, we can get the partition $\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\}$.

0 0 0 0
0 0 1 1
0 1 0 1
0 1 1 0
1 0 0 1
1 0 1 0
1 1 0 0
1 1 1 1

Fig. 1. OA($8, 2^4, 3$)

Column	P-set
2 3	$\{4, 8\} \{3, 7\}$
	$\{2, 6\} \{1, 5\}$
1 3	$\{6, 8\} \{5, 7\}$
	$\{2, 4\} \{1, 3\}$
1 2	$\{7, 8\} \{5, 6\}$
	$\{3, 4\} \{1, 2\}$

Fig. 2. Stack of p-sets

Theorem 1. An $N \times (m + 1)$ matrix is an OA($N, s_1, \dots, s_{m+1}, t$) iff the matrix formed by its first m columns is an OA(N, s_1, \dots, s_m, t), and for each $N \times (t - 1)$ sub-array of the first m columns, the s_{m+1} symbols in column $m + 1$ are equally distributed within the rows in each p-set induced by the sub-array.

According to Theorem 1, firstly we should calculate all p-sets induced by all $N \times (t - 1)$ sub-arrays from the first m columns, then the constraints for column

$m + 1$ can be obtained directly. At each search step we may add the p-sets incrementally to a stack so as to avoid recomputation.

3.4 Translation to Boolean and Pseudo-boolean Constraints

Once all the p-sets are computed, it's easy to translate the constraints to CNFs as an input file for the SAT solver. Suppose we are processing column $m + 1$. For an arbitrary p-set T , each of the s_{m+1} symbols from column $m + 1$ should appear $\frac{|T|}{s_{m+1}}$ times in the rows of T , where $|T|$ is the cardinality of T . In other words, each symbol must not appear $\frac{|T|}{s_{m+1}} + 1$ times or more. This constraint can be directly translated to logic formulas. For each $\mathcal{X} \subseteq T$, $|\mathcal{X}| = \frac{|T|}{s_{m+1}} + 1$, we add:

$$\bigwedge_{0 \leq i < s_{m+1}} (\bigvee_{j \in \mathcal{X}} v_j \neq i)$$

where v_j denotes the symbol in row j , column $m + 1$. The constraints are then expressed by the conjunction of all the CNFs.

For a p-set T , the above translation would generate $(\binom{|T|}{|\mathcal{X}|}) \times s_{m+1}$ CNFs. When the factor level s_{m+1} is small and T is large, the input file for SAT solver can be huge. So we present another encoding of the constraints which is more compact: translating to pseudo-Boolean constraints.

A pseudo-Boolean (PB) constraint is an equation or inequality between polynomials in 0-1 variables. A linear PB clause has the form: $\sum c_i \cdot L_i \sim d$, where $c_i, d \in \mathcal{Z}$, $\sim \in \{=, <, \leq, >, \geq\}$, and L_i s are literals. The constraint of a p-set is naturally represented by the following PB clauses:

$$\bigwedge_{0 \leq i < s_{m+1}} \sum_{j \in T} P_{j-i} = \frac{|T|}{s_{m+1}}$$

where P_{j-i} denotes the proposition $v_j = i$. In this way a p-set would generate only s_{m+1} PB clauses.

4 Exploiting Symmetries

For a constraint solving problem, a **symmetry** is a one to one mapping (bijection) on variables that preserves solutions and non-solutions. We say two solutions are **isomorphic** if there is a symmetry mapping one of them to the other. Since isomorphism is caused by symmetry, in the remainder of the paper, we use symmetry breaking and isomorphism elimination without distinction.

Obviously two orthogonal arrays with the same parameters are isomorphic if one can be obtained from the other by a finite sequence of row permutations, column permutations and permutations of symbols in each column. For example, Figure 3 illustrates two isomorphic OAs. It is better to eliminate isomorphism while searching for solutions as the whole search space is too redundant.

To eliminate isomorphism caused by permutations of rows and columns, a direct way is to introduce some order to fix the rows and columns. In a 2D-matrix,

0 0 0 0		0 0 1 0
0 0 0 1	1. Swap column 3 with column 4;	0 0 0 0
0 1 1 0	2. Permute symbol '0' and '1' in column 3	0 1 1 1
0 1 1 1		0 1 0 1
1 0 1 0		1 0 1 1
1 0 1 1		1 0 0 1
1 1 0 0		1 1 1 0
1 1 0 1		1 1 0 0

Fig. 3. Two isomorphic instances of OA(8, 2⁴, 2)

each row (column) can be viewed as a vector. The rows (columns) are *lexicographically ordered* if each row (column) is lexicographically smaller (denoted by \leq_{lex}) than the next (if any). Flener et al.[8] found that lexicographically ordering both the rows and the columns can break symmetries efficiently, while bringing in only a linear number of constraints. In our approach, we take this conclusion, adding the constraints that the matrix should be lexicographically ordered along both rows and columns. For MOAs, column lexicographic order is only imposed on the columns with the same factor levels.

Example 2. Figure 4 demonstrates three solutions of MOA(12, 3·2⁴, 2). A and B are lexicographically ordered along both the rows and the columns. Matrix C does not satisfy the lexicographic order since the third column $\not\leq_{lex}$ the fourth column, thus C would not be encountered in the search process.

In the following we will introduce two other symmetry breaking techniques. It can be easily proved that all these techniques could be combined to guide searching without losing any non-isomorphic solution [11].

4.1 Symbol Symmetries

There are symbol symmetries in the OA problem too. Assume that we are processing the column $X = \langle x_1, x_2, \dots, x_N \rangle$ and the column has s distinct symbols: $\{0, 1, \dots, s-1\}$. Permutations of these symbols would generate $s! - 1$ symmetries. In constraint programming symbol symmetries are also called symmetries of indistinguishable values, which can be tackled by imposing value precedence [14,10]. Our idea to break symbol symmetry is essentially the same. For each column vector in the isomorphic class, we can transform it to be lexicographically smallest by relabeling all the symbols, forcing their first appearances to be in an increasing order. In fact, the scope of their first appearances can be narrowed from $\{1, 2, \dots, N\}$ to $\{1, \dots, l\}$, where $l = \frac{N}{s_1 \times \dots \times s_{t-1}}$. Because after the preprocessing, in the first $t-1$ columns, the zero vector appears from row 1 to row l and the p-set $\{1, \dots, l\}$ is induced. All symbols in X would occur in the rows of a p-set. Formally, for all $1 \leq j < l$, $0 \leq p < q \leq s-1$, we add

$$\left(\bigwedge_{1 \leq i < j} (x_i \neq p \wedge x_i \neq q) \right) \rightarrow x_j \neq q \quad (1)$$

The formulas imply that for any two symbols p, q ($p < q$), the first appearance of q must not precede that of p in a column.

This trick is similar to the Least Number Heuristic (LNH) [18] essentially, except that, while LNH just needs to keep a variable to represent the largest value used in the assignments, here we add some propositional formulas.

Proposition 1. *A column of OA can be transformed to satisfy Formula (1).*

4.2 Filter

In this subsection we introduce a technique called Filter that can eliminate a class of symmetries, which arise from the automorphisms of init-block. If we permute the symbols in the init-block of an OA, reconstruct the init-block by swapping rows, we can always get another solution by performing some other isomorphic operations outside the init-block. We call this transformation procedure **init-block reconstruction**. The newly obtained array has the same init-block, satisfies all constraints of lexicographic order and symbol symmetry breaking, hence would be a solution to be discovered by the program.

We say two OAs are **symmetric with respect to (w.r.t.) an init-block reconstruction** if one is obtained from the other through this reconstruction.

Example 3. In Figure 4, we can obtain Matrix B from A by performing the following init-block reconstruction:

1. Permute symbol ‘0’ and ‘1’ in the first column.
2. Swap rows 1, 2, 3, 4 with rows 5, 6, 7, 8 respectively.
3. Permute symbol ‘0’ and ‘1’ in the last column.

After step 2, the init-block is unchanged. However, the last column of Matrix A is converted to $\langle 100101101010 \rangle$, contradicting Formula (1) which specifies that symbol ‘0’ must precede ‘1’. Step 3 is then performed to adjust the matrix and

0 0 0 0 0
0 0 1 1 1
0 1 0 1 1
0 1 1 0 0
1 0 0 0 1
1 0 0 1 0
1 1 1 0 0
1 1 1 1 1
2 0 1 0 1
2 0 1 1 0
2 1 0 0 1
2 1 0 1 0

(A)

0 0 0 0 0
0 0 0 1 1
0 1 1 0 1
0 1 1 1 0
1 0 0 0 1
1 0 1 1 0
1 1 0 1 0
1 1 1 0 1
2 0 1 0 0
2 0 1 1 1
2 1 0 0 0
2 1 0 1 1

(B)

0 0 0 0 0
0 0 1 0 1
0 1 0 1 0
0 1 1 1 1
1 0 0 1 1
1 0 1 1 0
1 1 0 0 0
1 1 1 0 1
2 0 0 1 1
2 0 1 0 0
2 1 0 0 1
2 1 1 1 0

(C)

Fig. 4. Three Instances of MOA($12, 3 \cdot 2^4, 2$)

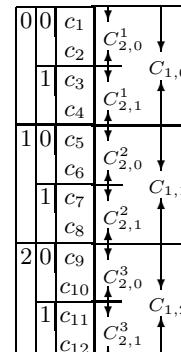


Fig. 5. Filter

we get Matrix B, which satisfies all symmetry breaking constraints. Therefore, A and B are symmetric w.r.t. the init-block reconstruction.

For an OA($N, s_1, s_2, \dots, s_k, t$), there are $s_1! \times s_2! \cdots s_t! - 1$ symmetries caused by init-block reconstructions except for the identity mapping. To break all these symmetries, it can be costly, because there are too many swappings and permutations to perform. We will introduce an approximate way to solve this problem. It is based on the following observation:

Suppose two matrices M_a and M_b are both OAs of $(N, s_1, s_2, \dots, s_k, t)$. We only focus on column $t + 1$. If $s_{t+1} > s_{t+2}$, column $t + 1$ cannot be exchanged with the following columns; For cases $s_{t+1} = s_{t+2}$, we can still fix column $t + 1$ by supposing the symbols in column $t + 1$ are special. Hence if column $t + 1$ of M_a and column $t + 1$ of M_b can't be obtained from each other by any init-block reconstruction, then the whole matrices M_a and M_b are not symmetric w.r.t. init-block reconstruction.

Once a symmetry caused by init-block reconstruction has been broken in column $t + 1$, it is prevented from spreading to the following columns. Breaking up symmetries in the column beyond the init-block is like setting a filter. To break such symmetries, we just add to column $t + 1$ a few constraints which are obtained according to the regularity of the init-block. We name this set of constraints **Filter**. Filter can not break all symmetries w.r.t. init-block reconstruction, but it can eliminate enough isomorphisms, yet the extra cost is negligible.

Now we describe how to set a Filter. For $1 \leq i \leq t$, denote the sub-vector of column $t + 1$ corresponding to symbol j in column i by $C_{i,j}$. To prevent symbol permutations in column i in the init-block, we can force the sub-vectors of column $t + 1$ corresponding to the symbols lexicographically ordered, i.e.,

$$\bigwedge_{0 \leq j < s_i - 1} C_{i,j} \leq_{lex} C_{i,j+1} \quad (2)$$

Take the case MOA(12, 3.2⁴, 2) for example. Suppose after preprocessing, the init-block is constructed and column $t + 1$ is represented by $\langle c_1, c_2, \dots, c_{12} \rangle$, as illustrated in Figure 5 ($C_{i,j}^k$ represents the k th segment of $C_{i,j}$). To prevent symbol permutations in the first column, we add $\langle c_1, c_2, c_3, c_4 \rangle \leq_{lex} \langle c_5, c_6, c_7, c_8 \rangle \leq_{lex} \langle c_9, c_{10}, c_{11}, c_{12} \rangle$. Similarly, for the 2nd column, we have $\langle c_1, c_2, c_5, c_6, c_9, c_{10} \rangle \leq_{lex} \langle c_3, c_4, c_7, c_8, c_{11}, c_{12} \rangle$. The third column of matrix A does not satisfy the first constraint since $\langle 0101 \rangle \not\leq_{lex} \langle 0011 \rangle$, thus would not be encountered during the search. The third column of B satisfies the two constraints.

Proposition 2. *An OA can be transformed so that column $t + 1$ satisfies the constraints of the Filter.*

Proof. For each i, j ($1 \leq i \leq t, 0 \leq j < s_i - 1$) contradicting Formula (2), adjust the matrix in this way: swap all those rows intersecting with the vector $C_{i,j}$ with those intersecting with $C_{i,j+1}$ accordingly, i.e., for each r that appears as a subscript in the vector $C_{i,j}$, exchange row r with row $r + \frac{N}{l_i \times s_i}$, then exchange symbol j with symbol $j + 1$ in column i to reconstruct the init-block. Each adjusting step makes column $t + 1$ lexicographically smaller while preserving the init-block. Since the

column vector has a lexicographic lower bound, the procedure will come to an end. Finally all the constraints of the Filter are satisfied. \square

5 Experimental Results

The OA searching tool BOAS (Backtrack-style OA Searcher) was implemented in the C programming language and integrated with the SAT solver MiniSat 2.0 beta [6] and the pseudo-Boolean constraint Solver PBS v2.1 [1]. We ran the program on an Intel 1.86GHZ Core Duo 2 PC with Fedora 7 OS.

To study the benefits of the symmetry breaking strategies, we carried out some experiments and asked BOAS to search the whole space to find all solutions. The numbers of loops and solutions are then compared. One loop means a successful assignment of values to the current column or concluding that there's no solution to the column. The efficiency of symbol symmetry breaking constraints is shown in Table 1. We can see that these constraints perform well for all level factors. For most of these cases, the program runs faster when using MiniSat as the search engine than using PBS, as can be revealed in Table 1. Therefore in Table 2 only the running times for MiniSat are listed, so as to make the table clear. Since there are always too many symmetries, when one technique is being tested, the others are enabled by default.

Table 1. Efficiency of Symbol Symmetry Breaking

OA	Breaking Symbol Symmetry?						
	No			Yes			
	Loop	Solution	Time (s)	Loop	Solution	Time (s)	MiniSat PBS
(18, 3 ⁶ , 2)	71237	41239	81.88 MiniSat	87.35 PBS	4116	922	14.90 18.06
(32, 4.2 ⁷ , 3)	17503	2958	9.98	13.85	2841	199	2.14 3.13
(64, 4 ⁴ .2 ⁶ , 3)	5312	192	1.57	3.25	707	3	1.08 1.07
(64, 4 ⁶ , 3)	800	576	1.38	3.15	133	1	0.76 1.11

Table 2. Efficiency of Filter

OA	Use Filter?						
	No			Yes			
	Loop	Solution	Time (s)	Loop	Solution	Time (s)	MiniSat PBS
(16, 2 ¹⁵ , 2)	327396	74	111.76	319776	74	90.73	
(32, 4.2 ⁷ , 3)	5517	597	7.40	2841	199	2.14	
(40, 5.2 ⁶ , 3)	6522	504	73.59	2976	144	27.88	
(56, 14.2 ⁴ , 3)	3432	0	252.78	2	0	0.12	
(60, 5.3.2 ² , 3)	720	719	19.14	11	10	0.00	
(64, 4 ⁴ .2 ⁶ , 3)	25452	108	201.33	707	3	1.08	
(128, 8.4.2 ³ , 4)	-	> 8000	-	11	10	0.00	

Table 2 shows the influence of Filter. From the table we can see that this strategy is more effective for OAs with large factor levels in the init-block, for example, the case $(56, 14.2^4, 3)$. When factor levels are large, there are more symbol permutations, resulting in more symmetries w.r.t. init-block reconstructions. The Filter strategy can eliminate enough of them.

Finding a Single OA

Now we see how our tool BOAS performs when we just want to find one solution. We concentrate on the OA cases with strengths no smaller than 3. As a result of space limitation, we just list some of them in Table 3. Many smaller cases which can be constructed in less than 0.01 second are omitted. The strength 3 cases with run sizes no more than 64 are from [4], and most of the non-existence results discussed in the paper can also be verified. The cases listed in Table 3 (A) all have solutions. For cases with strengths higher than 3, we have not seen any survey so far, so we also provide some conclusions for non-existence cases. ‘Para’ stands for ‘Parameters’, and ‘-’ means no result was obtained within the time limit of 2 hours.

From the table we can see that for most cases that have solutions, BOAS can find one solution in about one minute, either with MiniSat or PBS, thus would be helpful in practice. PBS outperforms MiniSat in finding first solutions generally, however, when exploring the whole space, MiniSat may do better. This is probably because the efficiency of MiniSat is more stable as the number of clauses grows.

Table 3. Running Time to Find First Solution

t	N	Para	Time (s)	
			MiniSat	PBS
3	48	6.2^7	0.27	0.01
3	48	$4.3.2^4$	0.02	0.00
3	48	4.2^{11}	42.33	0.02
3	56	14.2^3	0.03	0.03
3	56	7.2^6	2.47	1.03
3	64	16.2^3	0.15	1.14
3	64	$8.4.2^4$	8.15	0.00
3	64	4^6	0.09	1.11
3	64	$4^5.2^2$	0.09	1.10
3	64	$4^4.2^6$	0.01	0.02
3	64	$4^3.2^8$	150.47	510.71
3	72	$3^2.2^{12}$	-	89.04
3	81	3^{10}	-	12.14
3	81	9.3^4	0.00	0.00
3	96	$6.4^2.2^6$	-	1150.00
3	108	3^5	1.17	3.06

(A)

t	N	Para	Exists	Time (s)	
				MiniSat	PBS
4	64	4.2^6	YES	0.02	0.00
4	64	4.2^7	NO	3.21	11.65
4	64	2^8	YES	0.05	0.01
4	64	2^9	NO	99.66	97.58
4	80	2^7	NO	9.00	3.08
4	128	8.2^6	YES	20.15	0.31
4	128	$8.4.2^4$	NO	5.13	1.02
4	128	$4^3.2^3$	YES	0.00	0.05
4	128	$4^3.2^4$	NO	12.30	23.84
4	162	$3^5.2$	YES	0.01	0.00
4	162	3^6	NO	4003.05	4257.00
4	243	3^7	YES	227.03	110.28
5	128	2^9	YES	0.21	0.02
5	128	2^{10}	NO	870.06	868.73
5	160	5.2^6	YES	0.06	0.01
5	160	5.2^7	NO	645.59	1369.28

(B)

6 Related Works

In the last ten years, a few efforts have been made to develop algorithms for constructing OAs. Yamamoto et al. [16] proposed a constructive methodology. It is based on an algorithm obtaining modular representation vectors of an OA. The methodology is complex and restricted to pure-level OAs. It lacks implementation details. Xu [15] gave an approximate algorithm for constructing OAs and NOAs (nearly orthogonal arrays) through optimization of certain combinatorial criteria. His program can find solutions to some OAs of strength 2 and small runs with high probability, but for higher strengths, the chances seem slight.

Another trend in OA searching is employing global optimization algorithms such as simulated annealing, thresholding accepting and genetic algorithms. However, as discussed in [15], they are slow in convergence for the OA problem, failing to produce some OAs with quite moderate parameters. What is more, most of the algorithms mentioned above are inexact in the sense that they do not guarantee to provide a solution or draw the negative conclusion. In contrast, an exhaustive searching algorithm can always give a definite answer provided that there is enough memory and running time.

Snepvengers [13] developed a parallel implementation of an enumeration algorithm. In his program, isomorphisms are eliminated dynamically. Each time a new column is generated, all isomorphic operations are examined to get the lexicographically smallest solutions, therefore all isomorphisms can be eliminated. However, it may take a long time to perform isomorphism checking, and the largest factor level is restricted to 10. The parallel program was executed on a CRAY super-computer, and the running times were measured in hours. It is not convenient to compare our tool with that program directly. But it seems that we can solve some cases that are quite difficult. For example, to solve the case $(96, 6.4^2.2^6, 3)$, the parallel program ran on 16 processors for 6 days, but no result was obtained. Our program BOAS can find one solution in twenty minutes. In [13] OA($64, 4^2.2^5, 3$), OA($64, 4.2^8, 3$), OA($108, 4.3^4, 3$) and OA($108, 3^5, 3$) are claimed to be new results. It took their parallel program from 26.76 hours to 934.10 hours on a dozen of processors to obtain the results, while we can find these OAs within seconds. However, we feel it a bit unfair to perform such comparison since they aim at finding all non-isomorphic solutions (although they failed in three cases), while we try to find only one solution.

The cell-by-cell method for finding Covering Arrays (CAs) presented in [17] can also be extended to search for OAs. Currently we are investigating how to incorporate some of our symmetry breaking techniques into the framework.

7 Conclusion

As we mentioned in the Introduction, OAs (and MOAs) are quite important in combinatorics, and they also find applications in various areas such as software testing. It is desirable to have efficient tools for finding them. On the other hand, searching for such combinatorial objects poses interesting and challenging questions for automated reasoning research.

In this paper we proposed a framework for finding (M)OAs, which uses automated reasoning and constraint solving technology. And we also proposed some new symmetry breaking techniques for speeding up the search process. Our approach is a universal method in the sense that there is no restriction on the given parameters and that we can always get a conclusion after the search is completed. The experimental results show that our program is effective in solving cases with strengths higher than 2. The reason is that, in general the problems with more constraints are easier to solve for constraint solvers. They can take full advantage of constraint propagation techniques. But these cases (with high strengths) are more difficult for traditional mathematical methods.

The symmetry breaking techniques are quite effective in general. The SAT solver also contributes to the high efficiency of our program. But when the factor levels s_1, s_2, \dots, s_k are too small as compared with the run size N , we may get a large number of propositional clauses. Using PBS adds a level of modularity and makes it easier to program.

In the future, we will look further into various techniques, including more complete symmetry breaking, stronger constraint propagation, parallelization, direct search methods as well as mathematical results. We believe that they can be combined effectively. And we plan to implement a more powerful tool for finding (M)OAs.

References

1. Aloul, F.: The PBS Page, <http://www.eecs.umich.edu/~faloul/Tools/pbs/>
2. Bennett, F.E., Zhang, H.: Latin Squares with Self-Orthogonal Conjugates. *Discrete Mathematics* 284(1–3), 45–55 (2004)
3. Black, R.: *Pragmatic Software Testing*. Wiley, Chichester (2007)
4. Brouwer, A.E., Cohen, A.M., Nguyen, M.V.M.: Orthogonal Arrays of Strength 3 and Small Run Sizes. *J. Statistical Planning and Inf.* 136(9), 3268–3280 (2006)
5. Colbourn, C.J., Dinitz, J.H.: *Handbook of Combinatorial Designs*, 2nd edn. CRC Press, Boca Raton (2007)
6. Eén, N., Sorensson, N.: The MiniSat Page,
<http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat/>
7. Hedayat, A.S., Sloane, N.J.A., Stufken, J.: *Orthogonal Arrays: Theory and Applications*. Springer Series in Statistics (1999)
8. Flener, P., et al.: Breaking Row and Column Symmetries in Matrix Models. In: Van Hentenryck, P. (ed.) *CP 2002. LNCS*, vol. 2470, pp. 462–476. Springer, Heidelberg (2002)
9. Krishnan, R., et al.: Combinatorial Testing: Learnings from Our Experience. *ACM SIGSOFT Softw. Eng. Notes* 32(3), 1–8 (2007)
10. Law, Y.C., Lee, J.H.: Symmetry Breaking Constraints for Value Symmetries in Constraint Satisfaction. *Constraints* 11(2–3), 221–267 (2006)
11. Ma, F., Zhang, J.: Seaching for Orthogonal Arrays. *ISCAS-LCS-07-05* (2007)
12. Robert, B., Prowse, J., Phadke, M.S.: Robust Testing of AT&T PMX/Star MAIL using OATS. *AT&T Technical Journal* 71(3), 41–47 (1992)
13. Snepvangers, R.: Statistical Designs for High-Throughput Experimentation. Technical report, Stan Ackermans Institute (August 2006)

14. Walsh, T.: Symmetry Breaking using Value Precedence. In: Proc. ECAI 2006, pp. 168–172 (2006)
15. Xu, H.: An Algorithm for Constructing Orthogonal and Nearly-orthogonal Arrays with Mixed Levels and Small Runs. *Technometrics* 44, 356–368 (2002)
16. Yamamoto, S., et al.: Algorithm for the Construction and Classification of Orthogonal Arrays and Its Feasibility. *J. Combin. Inform. System Sci.* 23, 71–84 (1998)
17. Yan, J., Zhang, J.: Backtracking Algorithms and Search Heuristics to Generate Test Suites for Combinatorial Testing. In: Proceedings of COMPSAC, pp. 385–394 (2006); Extended version, to appear *J. of Systems and Software*
18. Zhang, J., Zhang, H.: SEM: A System for Enumerating Models. In: Proceedings of IJCAI 1995, pp. 298–303 (1995)

Statistical Model for Japanese Abbreviations

Norifumi Murayama¹ and Manabu Okumura²

¹ Interdisciplinary Graduate School of Science and Engineering

Tokyo Institute of Technology

murayama@lr.pi.titech.ac.jp

² Precision and Intelligence Laboratory

Tokyo Institute of Technology

oku@pi.titech.ac.jp

Abstract. We present a new approach to detect abbreviations given a root expression. The method is based on a statistical model combining two internal models: a generation and a verification model. The statistical model accounts for both the validity of abbreviations as a character sequence generated from a root (as learnt from the collection of abbreviation-root pairs) and their social validity, indicating how they are really used in the world (as obtained from a web search engine). The experimental results showed that our method outperforms traditional template-based methods. Specifically, using co-occurrence in the verification model yielded the best performance in our method.

1 Introduction

Technologies for identifying names have recently become a topic of interest for web services. Among them, those that detect the relation between a name and its abbreviations have received the most attention. Information about the relation can be used in many situations: for example, if someone wants to find information about “プレイステーション3” (*Purei-Suteisyon-Surii*¹, Play Station 3), the search engine should return documents that include not only “プレイステーション3” but also its abbreviation “プレステ3” (“Pure-Sute-Surii”). In this paper, we present a new method of detecting abbreviations from an original expression in Japanese.

We define an abbreviation as a shortened form of an original expression (called a root) that has the same meaning as the root. There are many types of abbreviations in Japanese: “東大” (*Tou-Dai*) from “東京大学” (*Toukyou-Daigaku*, Tokyo University) where the first character from each word is extracted (called an acronym), “コミケ” (*Komi-Ke*) from “コミックマーケット” (*Komikku-Maaketto*, comic market) where a character in the middle of one of the words is used, and “realm” (*Rearu*) from “realmadrid” (*Rearu-Madoriido*, Real Madrid) in which no characters from the second word are used.

¹ We append pronunciations for Japanese examples in alphabetical characters and represent the word boundaries of compound nouns by “-”.

There have been many studies on methods of detecting acronym-root pairs[1][2][3][4]. Acronyms are abbreviations that are formed using the initial letters of each word in a root. Most methods first detect acronym candidates in texts and then try to find root candidates from the text around the acronym candidates. Therefore, they assumed that abbreviations and their roots can be extracted from the text and that the extraction can be based on abbreviations as keys.

We think the latter assumption might not be applicable to Japanese. It is difficult to detect abbreviation candidates in Japanese texts because Japanese texts have no word boundaries and Japanese abbreviations have no obvious surface characteristics, such as using only upper case letters. These problems are also found in other languages, e.g. Chinese. Moreover, limiting the extraction of roots and their abbreviations to cases from within a text degrades coverage.

Therefore, we present a new approach to detect abbreviations given a root. The method is based on a statistical model combining two internal models. The respective models try to model the following information:

1. How natural the character sequence of an abbreviation is which is generated from the character sequence of a root, and
2. To what degree an abbreviation is socially recognized as a root.

Therefore, our statistical model accounts for both the validity of abbreviations as a character sequence generated from a root (as learnt from the collection of abbreviation-root pairs) and their social validity indicating how they are really used in the world (as obtained from a web search engine).

We think an approach based on a statistical model has at least the following merits:

- Based on the statistical model learnt from the collection of abbreviation-root pairs, we can acquire abbreviation-root pairs for known roots (existing names), but also find abbreviations for currently unknown roots (newly created names).
- An inputted root might not have any abbreviation or have several abbreviations. Since our method can yield abbreviations together with probabilistic scores for them, we expect that the appropriate number of abbreviations can be outputted for an inputted root by setting a suitable threshold.

Although we mainly experimented with Japanese abbreviations in this paper, we think that our approach can also be applied to abbreviations in other languages, such as English. English abbreviations are sometimes more complex than acronyms. For example, abbreviations, such as portmanteaus, are not formed only with uppercase letters (e.g. “Interpol” for “international police”), and abbreviations sometimes consist of multiple-word expressions (e.g. “Led Zep” for “Led Zeppelin”).

2 Related Work

As mentioned above, there have been many methods for detecting English acronym-root pairs by extracting them from documents [1][2][3][4]. Most

methods started by detecting acronym candidates in texts. Then, they searched for root candidates from the text around the acronym candidates. In the search for root candidates, they used templates such as acronym + “(” + root + “)” [1][2], edit distance[3], and machine learning[4].

Chang’s group presented an approach similar to ours for detecting Chinese abbreviations[5][6]. Their work is similar to ours in the following points: 1) There are common problems between Japanese and Chinese abbreviations. First, there are no word boundaries in Japanese and Chinese text, which makes it difficult to use the above-mentioned extraction approaches for acronyms. Moreover, the ways of abbreviating Chinese and Japanese are similar, especially in that abbreviations are constructed with Kanji characters (though Japanese abbreviations also use Hiragana and Katakana characters). 2) Both Chang’s group and our group use a statistical model, though their purpose is different: their approach is for “expanding”, whereas ours is for “abbreviating”.

In spite of these similarities, we think there are some important differences between the two approaches. One is the purpose of using the statistical model. As mentioned in section 1, since abbreviation candidates are difficult to detect in Japanese and Chinese texts, we think that abbreviations should not be the processing trigger.

Another is the formulation of the transformation rules in the statistical model, which we will describe later. As mentioned in the last section, since Japanese uses three character types (Kanji, Hiragana and Katakana), it has more forms of abbreviations than Chinese has. In Chang’s group’s work, rules were formulated with surface characters, and no generalization was taken into account in them. This sort of formulation might cause the rules in Japanese to become huge. Furthermore, more rules will cause a combinatorial explosion in computational cost.

Lastly, Chang’s group only dealt with a generation model. As we show in the experiments, we believe that a verification model is necessary for modeling abbreviations.

3 Characteristics of Japanese Abbreviations

In this section, we briefly describe the characteristics of Japanese abbreviations. As mentioned in section 1, there are many forms of Japanese abbreviation. Some typical examples are shown in figure 1.

First, abbreviations tend to be generated by extracting a sequence of characters from some words in a root. The extracted sequence of characters tend to come from the beginnings of the words (**a**, **c**, **d**, **e**) and **f**) in figure 1). Next, Japanese abbreviations are said to have a tendency for length. For example, if an abbreviation is made up only of Katakana characters, it tends to be three to five mora long² (**a**, **b**, **c**, **d**) and **f**). Moreover, as shown in **a**, **c**, and **d**, four-mora Katakana abbreviations tend to be created by combining two-mora

² The minimal unit of a syllable. In Japanese, mora are counted by the number of vowels (**a**, **i**, **u**, **e**, **o** and **n** in Roman characters).

	Root	Abbreviation
a)	Dōri i mu zu - Ka mu - Tou nu u ドリームズ カム トゥルー	Dōri - Ka mu ドリ カム
b)	Ko mi kku - Ma a ke tto コミック マーケット	Ko mi - Ke コミ ケ
c)	Cha a ri i - to - Cho ko re e to - Kou jou チャーリー と チョコレート 工場	Cha ri - Cho ko チャリ チョコ
d)	Tou kyou - Su ka - Pa ra da i su - O o ke su to ra 東京 スカ パラダイス オーケストラ	Su ka - Pa ra スカ パラ
e)	Tou kyou - Kou gyou - Dai gaku 東京 工業 大学	Tou - Kou - Dai 東 工 大
f)	Ke n ta kki i - -Fu ra i do - Chi kin ケンタッキー フライド チキン	Ke n ta Ke n ta kki i ケンタ or ケンタッキー

Fig. 1. Examples of abbreviations appearing in Japanese text

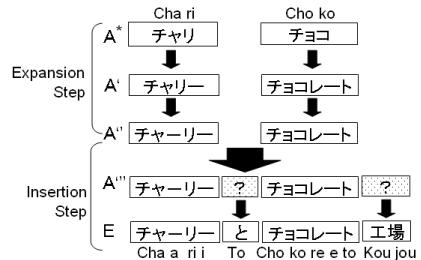


Fig. 2. Transformation Model

abbreviations. With Kanji characters, the length of an abbreviation tends to be two to three characters e).

Furthermore, if the root consists of Katakana and the extracted sequence includes the character “”³ or “_”⁴ in the middle, the character is dropped in the abbreviation c).

4 Overview of Our Method

4.1 Proposed Model

Our system receives a root and outputs its abbreviations in the order of probabilistic score. The task to identify the most probable abbreviation (character sequence) A for a root (word sequence) R is defined as follows:

$$A = \arg \max_{A^*} P(A^* | R) = \arg \max_{A^*} \frac{P(A^*, R)}{P(R)}. \quad (1)$$

In equation (1), $P(A^*, R)$, meaning the relevance of A^* to R , should be modeled from the following two viewpoints:

1. How natural the character sequence of A^* is which is generated from the character sequence of R , and
2. To what degree the expression A^* is socially recognized as the expression R .

As mentioned in the last section, four-mora Katakana abbreviations tend to be created by combining two-mora abbreviations. Such tendencies can be learnt with a statistical model from training data of abbreviation-root pairs. We call the first model the ‘generation’ model and explain it in section 5.

The second viewpoint indicates how popular A^* is when it has the same meaning as R . To model this idea, we use a search engine to get the information

³ The specific character which expresses a short pause before the next sound.

⁴ The specific character which expresses prolonged sound.

on word usage from the web. We call the second model the ‘verification’ model and explain it in section 6.

Assuming these two models are independent, $P(A^*, R)$ in equation (1) can be expanded into the following:

$$P(A^*, R) \simeq P_{gen}(A^*, R) * P_{ver}(A^*, R). \quad (2)$$

Equation (1) can in turn be expanded into the following:

$$A \simeq \arg \max_{A^*} P_{gen}(A^*, R) \frac{P_{ver}(A^*, R)}{P(R)}. \quad (3)$$

4.2 System Flow

The following is the procedure of our system that outputs abbreviations for an inputted root.

1. Segment the root into a sequence of words by using the Japanese morphological analyzer Mecab⁵,
2. Generate abbreviation candidates with the generation model, and rank them according to the scores of the model,
3. Apply the verification model to the top-N generated candidates, and rerank them according to the combined scores of the generation and verification models.

We need to use a web search engine for calculating the verification model. Since the number of possible abbreviation candidates might be huge (approximately the number of subsequences of characters of a root), it is unrealistic to calculate the verification model for all candidates. Therefore, we take a two-step approach to reduce the number of web searches. In the experiments described in section 7, we chose the top 30 candidates for verification.

5 Generation Model

We use the noisy channel model as our generation model. The noisy channel model is a popular statistical model for machine translation [7], text summarization [8], etc. As the work on text summarization indicates, this model naturally formalizes the process where a shortened sequence is obtained from the original.

The generation model in equation (3) can be transformed into the following:

$$P_{gen}(A^*, R) = P_{gen}(A^*)P_{gen}(R|A^*). \quad (4)$$

In equation (4), $P_{gen}(R|A^*)$ is called a “transformation model” from A^* to R , and $P_{gen}(A^*)$ is called a “language model” of the abbreviation that indicates the appropriateness of A^* .

⁵ <http://mecab.sourceforge.net/>

5.1 Transformation Model

$P(R|A^*)$ indicates a probability where root R is generated from abbreviation A^* . We model this generation process in two steps: expansion and insertion. Figure 2 shows a sample generation process for “チャーリーとチョコレート工場” (Chaarii-To-Chokoreeto-Koujou, Charlie and the Chocolate Factory⁶).

First, in the expansion step, each fragment of the abbreviation is expanded to the corresponding word in the root. In the insertion step, the words in the root which have no corresponding fragments in abbreviations are inserted to yield the original root.

The two steps are each divided into two sub-steps: In the expansion step, front and back characters are added to each fragment of the abbreviations (transformation from A to A'), and then “ \circ ” or “ $-$ ” is inserted if needed (transformation from A' to A''). In the insertion step, the positions for inserting words to yield the root are first decided (transformation from A'' to A'''), and the actual words are then inserted at the positions (transformation from A''' to R).

Assuming the transformation in each step is independent and each probability of the transformation is only dependent on the previous state, the whole transformation process can be represented as follows:

$$P(R|A) \simeq P(R|A''')P(A''''|A'')P(A''|A')P(A'|A). \quad (5)$$

Expansion Step. In the first sub-step, that is the transformation of A to A' , we model the tendencies when a fragment of the abbreviation is extracted from a word in the root, as mentioned in section 3. We model the probability of this sub-step as

$$P(A'|A) = \prod_{i=0}^N P(a'_i|a_i) \quad (6)$$

$$\simeq \prod_{i=0}^N P(\text{headnum}, \text{char} | \text{type}, \text{abnum}), \quad (7)$$

where N is the number of fragments in the abbreviation, and a_i and a'_i are the fragments of A and A' , respectively. In equation (6), the probability of the transformation is gotten by multiplying the probabilities of the transformation for each fragment. Equation (7) is the generalized form of equation (6). The elements in equation (7) are as follows:

- *type*: character type of a_i (Hiragana, Katakana, Kanji, roman alphabetic, numerals, or combination of these)
- *abnum*: number of characters (or moras in the case of Hiragana and Katakana characters) in a_i
- *headnum*: number of characters added to the beginning of a_i when a_i is transformed into a'_i
- *char*: characters added to the end of a_i .

⁶ The title of a movie.

As mentioned in section 2, since there are many types of Japanese abbreviations, we think the generalization is inevitable.

In the example shown in figure 2, the probability of this sub-step is modeled as follows:

$$P(\text{チャリー} | \text{チャリ}) P(\text{チョコレート} | \text{チョコ}) = P(0, \text{—} | \text{Kana}, 2) P(0, \text{レ} | \text{Kana}, 2).$$

The next sub-step, the transformation from A' to A'', models the unique phenomena concerning “” and “—” that were described in section 3. We model the probability of this sub-step as

$$P(A'' | A') = \prod_{i=0}^N P(a''_i | a'_i) \simeq \prod_{i=0}^N P(\text{wordch} | \text{type}, \text{abbch}), \quad (8)$$

where

- *abbch*: which of “—” or “” is included in a'_i
- *wordch*: “” or “” to be inserted when a'_i is transformed into a''_i .

In the example shown in figure 2, the probability of this sub-step is modeled as

$$P(\text{チャーリー} | \text{チャリー}) P(\text{チョコレート} | \text{チョコレート}) = P(\text{—} | \text{Kana}, \text{—}) P(\text{none} | \text{Kana}, \text{—}).$$

Insertion Step. In the insertion step, we model the words to be selected from the root. The first sub-step, which indicates the transformation of A'' into A''', models the position of the words to be selected in the root. The probability of this sub-step is modeled as

$$P(A''' | A'') = P(\text{beginning}, \text{middle}, \text{end} | \text{type}, \text{abbfragnum}). \quad (9)$$

The elements in equation (9) are

- *type*: character type in A''
- *abbfragnum*: number of fragments in A''
- *beginning*: number of words inserted at the beginning of A'''
- *middle*: number of words inserted in the middle of A'''
- *end*: number of words inserted at the end of A'''.

In the example shown in figure 2, the probability of this sub-step is modeled as

$$P(\text{チャーリー} [?] \text{チョコレート} [?] | \text{チャーリー} \text{—} \text{チョコレート}) = P(0, 1, 1 | \text{Kana}, 2).$$

The second sub-step, which indicates the transformation from A''' into R, models the insertion (selection) of words in the root. The probability of this sub-step is modeled as

$$P(R | A''') = \prod_{r_i \in R^-} P(r_i | a''_i) \simeq \prod_{r_i \in R^-} P(\text{wordtype}, \text{wordnum} | \text{type}, \text{location}). \quad (10)$$

R^- is a set of words in the root that have no corresponding fragments in the abbreviation, and a''_i is an unknown word that is inserted in the above internal step as the generalized expression of r_i . The elements in equation (10) are

- *type*: character type in A'''
- *location*: insertion location of a_i''' (beginning, middle or end)
- *wordtype*: character type in r_i
- *wordnum*: number of characters in r_i .

In the example shown in figure 2, the probability of this sub-step is modeled as

$$\begin{aligned} P(\text{チャーリーとチョコレート工場} \mid \text{チャーリー } [?] \text{ チョコレート } [?]) \\ = P(Kana, 1 \mid Kana, \text{middle})P(Kanji, 2 \mid Kana, \text{bottom}). \end{aligned}$$

5.2 Language Model

$P(A^*)$ in equation (4) indicates the appropriateness of abbreviation A^* . With this model, we capture tendencies in the form of an abbreviation, as discussed in section 3.

We define $P(A^*)$ as

$$P(A^*) \simeq P(type, length, fragnum) \prod_{a_i} P(fraglength, fragtype). \quad (11)$$

The elements in equation (11) are

- *type*: character type in A^*
- *length*: number of characters (or moras) in A^*
- *fragnum*: number of fragments in A^*
- *fragtype*: character type of a_i
- *fraglength*: number of characters in a_i .

The first term in equation (11) contains the global information for the abbreviation, and the product contains information for each fragment of the abbreviation.

In the case of figure 2, the probability of the language model is calculated as $P(\text{チャリチョコ}) = P(Kana, 4, 2)P(2, Kana)P(2, Kana)$.

6 Verification Model

To detect “correct” abbreviations in the abbreviation candidates generated from a root, it is necessary to verify whether they are actually used as the original root. To implement the verification, we use a web search engine and calculate the scores of association between an abbreviation and its original root. The association can be modeled in two ways: by using **similarity** or by using **co-occurrence**. The **similarity** of abbreviation-root pairs indicates whether the abbreviation is considered to have the same meaning as the root, and the **co-occurrence** of abbreviation-root pairs indicates whether the abbreviation tends to co-occur with the root.

In the experiments in section 7, we used Yahoo! web search APIs[9] as the web search engine and the top 50 retrieved snippets for calculating the similarity or co-occurrence.

6.1 Modeling Similarity

When an abbreviation has the same meaning as its root, the pair will likely have a similar context. To implement this idea, we compare snippets retrieved with the abbreviation and ones retrieved with the root.

The verification model in equation (3) can be transformed into the following:

$$\frac{P_{ver}(A, R)}{P(R)} = \frac{P_{ver}(R|A)P(A)}{P(R)} = P_{ver}(R|A)\frac{P(A)}{P(R)}. \quad (12)$$

Here, representing the total number of web documents as $|D|$, the number of documents that contain the abbreviation and the root as $|A|$ and $|R|$, respectively, the latter term in equation (12) can be transformed into the following:

$$\frac{P(A)}{P(R)} = \frac{|A|/|D|}{|R|/|D|} = \frac{|A|}{|R|}. \quad (13)$$

$|A|$ and $|R|$ can be calculated as the numbers of hits with queries A and R, respectively. The verification model in equation (3) can finally be transformed into the following:

$$\frac{P_{ver}(A, R)}{P(R)} = P_{ver}(R|A)\frac{|A|}{|R|}. \quad (14)$$

We calculated $P_{ver}(R|A)$ as the generation probability of R from the language model induced by A using the language-model approach proposed in [10].

$$\begin{aligned} P(R|A) &\simeq P(S_R|S_A) \simeq (\prod_{w \in S_R} P_{gen}(w|S_A))^{\frac{1}{|S_R|}} \\ &\simeq (\prod_{w \in S_R} \frac{tf(w, S_A)}{\sum_{w' \in S_A} tf(w', S_A)})^{\frac{1}{|S_R|}} \end{aligned} \quad (15)$$

In equation (15), S_R and S_A respectively represent the sets of retrieved snippets for R and A , and w and w' indicate the words occurring in these sets. The estimation of the language model from S_A uses Laplace smoothing with 0.00001 as the value of δ .

6.2 Modeling Co-occurrence

The second model tries to capture the characteristics of an abbreviation frequently co-occurring with its root in the same document. However, mere co-occurrence in a document might not show that the abbreviation originates in the root. Therefore, in our model, we take into account the distance between the two expressions (abbreviation and its root) in a snippet, assuming that near co-occurrence indicates a strong semantic relationship between the words.

The verification model in equation (3) can be transformed into the following:

$$\frac{P_{ver}(A, R)}{P(R)} \simeq \frac{distance(R, A)^{\frac{|R, A|}{|D|}}}{\frac{|R|}{|D|}} = \frac{distance(R, A)|R, A|}{|R|}. \quad (16)$$

$Distance(R, A)$, ranging from 0 to 1, was calculated as follows:

1. Use the top 50 snippets retrieved with a query of an abbreviation and its root
2. Measure the distance between the abbreviation and the root in bytes in a snippet
3. Average the distances for 50 snippets
4. $Distance(R, A)$ is calculated as $(100 - \text{the average}) / 100$.

7 Experiments

7.1 Data Set

In this section, we explain our training/testing data set. The training data is the set of correct abbreviation-root pairs used for training the transformation model in section 5.1.

We used the template-based approach for collecting the training data. First, we listed named entities in the Japanese version of Wikipedia⁷, such as names of TV programs, books, movies, and bands, as root candidates. We prepared six templates for acquiring abbreviation-root pairs in Japanese, such as “Root の 略語はAbbreviation”, which means “An abbreviation of Root is Abbreviation”. Then, we filled these templates with those named entities from Wikipedia and made a search with the templates as queries by using Yahoo! web search APIs. We used a simple method based on the DP-matching algorithm to extract the abbreviation candidates from the snippets of the search results.

We used the entities gathered from Wikipedia on August 24th, 2007. The number of entities was 167,429. Manual evaluation of the extracted abbreviation-root pairs using the template-based method yielded 13,685 correct pairs. 13,359 correct abbreviation-root pairs were used for the training data, excluding pairs selected as test data.

To prepare the test data, we selected the subset of the named entities by random sampling. Ideally, all abbreviations should be enumerated in the test data. However, since enumeration is almost impossible, we adopted a ‘pooling’ method with our proposed method and the template-based method. That is, we manually evaluated all the outputs of both methods and regarded the acquired set of correct pairs as the whole set. The number of sampled entities was 1,000. In those entities, 248 entities had the correct pairs, and there were 326 correct abbreviation-root pairs. The remaining 752 entities were judged to have no abbreviations.

7.2 Evaluation Measure

First, we evaluated how many correct pairs our method could correctly identify. Since our method can output an abbreviation for a root with its score⁸, we chose

⁷ <http://ja.wikipedia.org/wiki/>

⁸ In the template-based approach to be compared with ours, the number of occurrences can be considered as a score.

MAP(Mean Average-Precision) and MRR(Mean Reciprocal Rank) as our evaluation measures: these measures are frequently used in evaluations of information retrieval and question answering. MAP and MRR are calculated as follows:

$$MRR = \frac{1}{|R|} \sum_R \frac{1}{|CorrectAbbs|} \sum_{CorrectAbbs} 1/RANK$$

$$MAP = \frac{1}{|R|} \sum_R \frac{1}{|CorrectAbbs|} \sum_{CorrectAbbs} Precision,$$

where R is the set of roots, $RANK$ is the rank of correct abbreviations, and $Precision$ is the precision at the rank of correct abbreviations. The results also include the precision-recall graph. In this evaluation, therefore, 326 correct pairs were used.

In the second evaluation, we tried to evaluate how correctly our method could judge whether the inputted root had an abbreviation or not. Specifically, a root was judged to have an abbreviation if the score of the top-ranked abbreviation exceeded a threshold. The precision-recall graph was plotted by changing the threshold. The second evaluation used all 1000 roots.

7.3 Experimental Results

The experiments compared the following four methods:

- **sim.** Our method (Verification with similarity model),
- **cooc.** Our method (Verification with co-occurrence model),
- **gen.** Our method without a verification model,
- **temp.** Template-based method.

Table 1 shows the MAPs and MRRs for all methods. The table indicates our method outperformed the template-based method. It also indicates the verification model is needed, and co-occurrence outperformed similarity as the verification model.

The precision-recall graph for all methods is shown in figure 3. Figure 3 shows that our method outperformed the template-based method in terms of MAP and MRR, since our method significantly improved recall, and the results indicate that our method had broader coverage on abbreviations.

The precision-recall graph for judging the existence of abbreviations is shown in figure 4. The baseline in the figure is the one when all the roots are judged to have an abbreviation. The graph, unfortunately, indicates that our method does not always function as a good judge of the existence of abbreviations.

Table 1. MAP and MRR for the methods

Method	MAP	MRR
sim.	0.433	0.424
cooc.	0.525	0.509
gen.	0.353	0.340
temp.	0.214	0.259

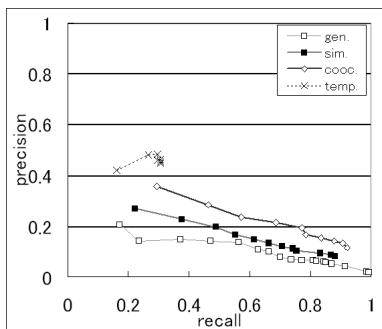


Fig. 3. Precision-recall graph

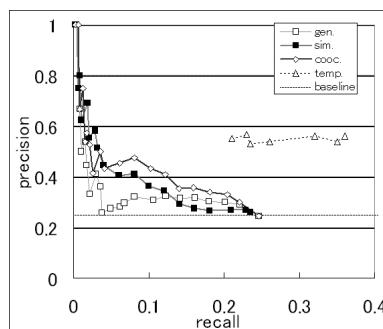


Fig. 4. Precision-recall graph as a means of judging abbreviation existence

8 Conclusion

We presented a new approach of detecting abbreviations given a root. The method was based on a statistical model combining two internal models. The respective models try to model the following information:

1. How natural the character sequence of the abbreviation is which is generated from the character sequence of a root, and
2. To what degree an abbreviation is socially recognized as a root.

Therefore, our statistical model takes into account both the validity of an abbreviation as a character sequence generated from a root, (as learnt from the collection of abbreviation-root pairs), and its social validity as to whether the abbreviation is really used in the world (as obtained from a web search engine).

The experimental results showed that our method outperformed the conventional template-based method. Specifically, using co-occurrence in the verification model yielded the best performance in our method.

Our next job will be to improve the validity of the score showing the existence of abbreviations. Furthermore, we will refine our verification model.

Acknowledgments

This work was supported by the 21st Century COE Program “Framework for Systematization and Application of Large-scale Knowledge Resources” of the Japan Society for the Promotion of Science.

References

1. Pustejovsky, J., Castano, J., Cochran, B., Kotecki, M., Morrell, M., Rumshisky, A.: Linguistic knowledge extraction from medline: Automatic construction of an acronym database. In: 10th World Congress on Health and Medical Informatics, Medinfo 2001 (2001)

2. Zahariev, M.: An efficient methodology for acronym-expansion matching. In: Proceedings of the International Conference on Information and Knowledge Engineering, IKE 2003, vol. 1 (2003)
3. Park, Y., Byrd, R.J.: Hybrid text mining for finding terms and their abbreviations. In: Conference on Empirical Methods in Natural Language Processing (EMNLP) (2001)
4. David, N., Peter, T.: A supervised learning approach to acronym identification. In: Proceedings 18th Conference of the Canadian Society for Computational Studies of Intelligence, pp. 319–329 (2005)
5. Chang, J.S., Lai, Y.T.: A preliminary study on probabilistic models for chinese abbreviations. In: Proceedings of the Third SIGHAN Workshop on Chinese Language Learning, ACL 2004, pp. 9–16 (2004)
6. Chang, J.S., Teng, W.L.: Mining atomic chinese abbreviation pairs: A probabilistic model for single character word recovery. In: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, COLING-ACL 2006, pp. 17–24 (2006)
7. Brown, P.F., Cocke, J., Pietra, S.A.D., Pietra, V.J.D., Jelinek, F., Lafferty, J.D., Mercer, R.L., Roossin, P.S.: A statistical approach to machine translation. Computational Linguistics 16, 79–85 (1990)
8. Daume III, H., Marcu, D.: A noisy-channel model for document compression. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002) (2002)
9. Yahoo Japan Corporation: Yahoo! developer network (2007),
<http://developer.yahoo.co.jp/>
10. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of SIGIR, pp. 275–281 (1998)

A Novel Heuristic Algorithm for Privacy Preserving of Associative Classification

Nattapon Harnsamut and Juggapong Natwichai

Computer Engineering Department, Faculty of Engineering
Chiang Mai University, Chiang Mai, Thailand
harnsamut@gmail.com, juggapong@chiangmai.ac.th

Abstract. Since individual data are being collected everywhere in the era of data explosion, privacy preserving has become a necessity for any data mining task. Therefore, data transformation to ensure privacy preservation is needed. Meanwhile, the transformed data must have quality to be used in the intended data mining task, i.e. the impact on the data quality with regard to the data mining task must be minimized. However, the data transformation problem to preserve the data privacy while minimizing the impact has been proven as an NP-hard. In this paper, we address the problem of maintaining the data quality in the scenarios which the transformed data will be used to build associative classification models. We propose a novel heuristic algorithm to preserve the privacy and maintain the data quality. Our heuristic is guided by the classification correction rate (CCR) of the given datasets. Our proposed algorithm is validated by experiments. From the experiments, the results show that the proposed algorithm is not only efficient, but also highly effective.

1 Introduction

Privacy is an important issue in every data processing task including data mining. When data are to be released to another business collaborator for data mining purpose, the issue must be addressed. Essentially, all identifiers (e.g., Name and ID) must be removed. Unfortunately, the released dataset can still “link” to another dataset by common attributes between them. For example, consider the datasets in Table 1 and 2. Suppose that the dataset in Table 1, which is released from a hospital, is to be used to build a classifier by a data analysis company. While, another dataset as shown in Table 2 is released publicly for voting purpose. By considering dataset from Table 1 alone could misjudge that the privacy of the individuals containing in this dataset has been already preserved due to the removal of the identifiers. However, if an adversary wants to find private information about a man named “Somchai” who lives in an area with postal code “50200”, and his age is approximately 50 years-old. The adversary can link the dataset in Table 1 to the dataset in Table 2 together using postal code and age attributes, subsequently, his medical condition will be disclosed.

Table 1. A released dataset

Postal code	Age	Sex	Disease
50211	1-25	Female	Fever
50211	1-25	Female	Fever
50202	26-50	Male	Flu
50202	26-50	Male	Flu
50200	26-50	Male	Cancer

Table 2. A public dataset

Name	Postal code	Age	Sex
Manee	50211	19	Female
Wanthong	50211	24	Female
Tawan	50202	32	Male
Suttee	50202	41	Male
Somchai	50200	50	Male

k-Anonymity [1] is a well-known privacy model for datasets, in which a dataset is said to satisfy the anonymity, if for every tuple in the dataset, there are another *k*-1 tuples which are indistinguishable from the tuple for all “linkable” attributes. If a dataset does not satisfy the standard, we can transform such the dataset by generalizing it until the standard is reached. The *k*-Anonymity is simple and meaningful, then, its has been applied in a lot of work [2,3,4,5,6,7,8,9]. For example, the dataset in Table 1 can be generalized into a 2-Anonymity dataset by changing the last digit of the postal code as shown in Table 3. However, we must address data quality issue in the transformation processes, i.e. the transformed datasets should have enough quality to be used by the designate data processing which is decided at the first place. As in our example, the transformed dataset should be able to be used to build classifiers accurately.

Table 3. 2-Anonymity dataset

Postal code	Age	Sex	Disease
5021*	1-25	Female	Fever
5021*	1-25	Female	Fever
5020*	26-50	Male	Flu
5020*	26-50	Male	Flu
5020*	26-50	Male	Cancer

In this paper, we address the problem of privacy preservation when a type of classification, associative classification [10,11], is to be applied to the released datasets. Such the classification model works on support and confidence scheme as association rules [12], but having a designate attribute as class label. As data transformation for privacy preservation problem is proven to be an NP-hard

problem in [5], we propose a novel heuristic algorithm called Minimum Classification Correction Rate Transformation algorithm (MCCRT). The proposed algorithm transforms the datasets based on the prediction accuracy denoted as classification correction rate (CCR) of the linkable attributes of the given dataset. Given a testing dataset, the CCR is computed by classifying each tuple into the predicted class label from the model, subsequently, comparing the predicted class label with the actual class label, finally determining the ratio between the number of the correct prediction and the total number of the tuples. The higher CCR value means the classification model can be used to classify the target data better.

The organization of this paper is as follows. Section 2 presents the problem definition. The proposed algorithm for such the problem is presented in Section 3. Our work is validated by experiments in Section 4. Finally, we presents the conclusion in Section 5.

2 Problem Definition

In this section, the basic notations of the problem and problem definition are presented.

Definition 1 (Dataset). Let a dataset D be a collection of tuples defined on a schema \mathbf{A} , $D = \{d^1, d^2, \dots, d^n\}$. For each attribute $A_j \in \mathbf{A}$, its domain is denoted as $\text{dom}(A_j) \subseteq \mathcal{N}$, where \mathcal{N} is the set of natural number. For each $d^i \in D$, $d^i(\mathbf{A}) = (d^i(A_1), d^i(A_2), \dots, d^i(A_k))$, denoted as $(d_1^i, d_2^i, \dots, d_k^i)$. Note here that tuples in a table is not necessary to be unique.

Let C be a set of class labels, such that $C = \{c_1, c_2, \dots, c_o\}$, each $c_m \in C$ is a natural number. The class label of a tuple d^i is denoted as $d^i.\text{Class}$.

Definition 2 (Associative Classification). A literal p is a pair, consisting of an attribute A_j and a value v in $\text{dom}(A_j)$. A tuple d^i will **satisfy** the literal $p(A_j, v)$ iff $d_j^i = v$.

For all $l \in L$, $r_l : \bigwedge p \rightarrow c_m$, where p is the literal, and c_m is a class label. The left hand side (LHS) of the rule r_l is the conjunction of the literals, denoted as $r_l.\text{LHS}$. The right hand side (RHS) is a class label of the rule r_l , denoted as $r_l.\text{RHS}$.

A tuple d^i **satisfies** the classification rule r_l iff it satisfies all literals in $r_l.\text{LHS}$, and has a class label c_m as $r_l.\text{RHS}$.

A tuple d^i which satisfies the classification rule r_l is called **supporting tuple** of r_l . The **support** of the rule r_l , denoted as $\text{Sup}(r_l)$, is the ratio between the number of supporting tuples of r_l and the total number of tuples. The **confidence** of rule r_l , denoted as $\text{Conf}(r_l)$, is the ratio between $\text{Sup}(r_l)$ and the total number of tuples which satisfy all literals in LHS of r_l . Given a dataset D , a set of class labels C , a minimal support threshold minsup , and a minimal confidence threshold minconf , a set of classification rules $R = \{r_1, r_2, \dots, r_q\}$ can be derived.

Generally, the set of attributes of a dataset which can “link” to another dataset is called “quasi-identifier”. The linkage process is also called “re-identifying” as it can identify the de-identified data. A quasi-identifier attribute may not be non-sensitive attribute, i.e. it can be disclosed but may be used to re-identify individuals.

Definition 3 (Quasi-Identifier). *A quasi-identifier of the dataset D , written Q_D , is the minimal subset of the attributes \mathbf{A} that can re-identify the tuples in D by using external data.*

Definition 4 (k -Anonymity). *A dataset D with a schema \mathbf{A} and a quasi-identifier Q_D satisfies k -anonymity iff each tuple $d^i \in D$ there exist $k - 1$ other tuples $d^{i^1}, d^{i^2}, \dots, d^{i^{k-1}} \in D$ such that $d_j^i = d_j^{i^1} = d_j^{i^2} = \dots = d_j^{i^{k-1}}, \forall A_j \in \mathbf{A}$.*

Definition 5 (Generalization). *Let a domain $\text{dom}^*(A_j) = \{P_1, P_2, \dots\}$ be a generalization of a domain $\text{dom}(A_j)$ of an attribute A_j where $\bigcup P_{jt} = \text{dom}(A_j)$ and $P_{jt} \cap P_{jt'} = \emptyset$, for $jt \neq jt'$. For a value v in $\text{dom}(A_j)$, its generalized value P_{jt} in a generalized domain $\text{dom}^*(A_j)$ is denoted as $\phi_{\text{dom}^*(A_j)}(v)$.*

Let \prec_G be a partial order on domains, $\text{dom}(A_j) \prec_G \text{dom}^*(A_j)$ iff $\text{dom}^*(A_j)$ is a generalization of $\text{dom}(A_j)$.

For a set of attributes $\mathbf{A}', \mathbf{A}' \subseteq \mathbf{A}$, let $\text{dom}^*(\mathbf{A}')$ be a generalization of a domain $\text{dom}(\mathbf{A}')$. A dataset D can be generalized to D^* by replacing the values of $d^i(\mathbf{A}')$ with a generalized value $\phi_{\text{dom}^*(\mathbf{A}')} (d^i(\mathbf{A}'))$ to get a new tuple d^{i*} . The tuple d^{i*} is defined as a generalization of the tuple d^i .

For an attribute, the set of generalization domains on it forms a hierarchy. Typically, we can derive hierarchies from the priori knowledge of the given data. From the dataset in Table 1, we can apply the hierarchies shown in Figure 1a) and 1b). Also, we can simply partition the domain into intervals for the numerical attribute as shown in Figure 1c).

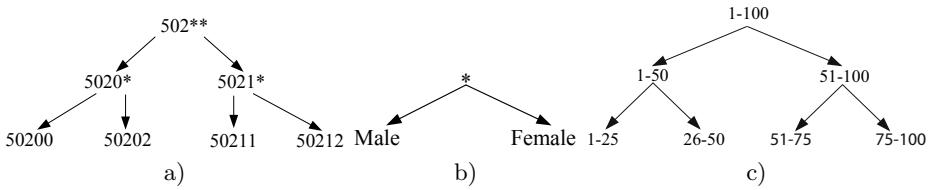


Fig. 1. Postal Code, Sex, and Age Hierarchies

Definition 6 (k -Anonymization). *k -Anonymization is transformation of D into D' where D' is generalization of D which satisfies the k -Anonymity property.*

After the basic definitions were defined, we formalize the problem of privacy preserving for associative classification as follow.

Problem 1. Given a dataset D with a set of class label C , a quasi-identifier Q_D , a minimal support threshold minsup , a minimal confidence threshold minconf ,

a k value, find D' which satisfies k -anonymity property by anonymization such that the impact on data quality with regards to the associative classification, C_{FCM} , is minimized. Such the impact metric proposed in [13] is defined as follows.

$$C_{FCM} = \frac{\sum_{fp \in FP} \frac{hp_{fp}}{\text{Full Generalization}} \times Sup(fp)}{\sum_{fp \in FP} Sup(fp)} \quad (1)$$

where fp is a frequency-pair of (p, c_m) which $Sup(fp) \geq minsup$, p is a literal, c_m is a class label, the attribute in a literal p is in the quasi-identifier Q_D . FP is the set of all fp from D . hp_{fp} is the height of the generalization value of pair fp .

The domain of the frequency-based metric is in range $[0,1]$ in which the larger the value is the greater impact on the result dataset. The metric will penalty any transformation which degrades pairs (p, c_m) whose their support equal or greater than $minsup$. This is because such the pairs will be used to derive the associative classification rules, therefore, preserve this type of property will also preserve the characteristics of the classifier. When comparing generalization by two attributes, the attribute with higher hierarchy will cause less penalty since it introduces less generalized values. Also, any transformation which selects to degrade pairs with high support will be penalized more than those which degrade less support pairs.

3 Algorithm

In this section, we present a heuristic algorithm to solve the problem of privacy preserving when the given datasets are intended to derive associative classification. The algorithm transforms the given datasets to satisfy k -Anonymity, and also tries to minimize the impact C_{FCM} .

Figure 2 shows the pseudo code of the proposed algorithm, MCCRT. It begins with sorting all quasi-identifier attributes using their CCRs increasingly. Prior to the sorting, the CCR of each attribute is determined as follows. For each attribute, we derive the set of one-literal classification rules satisfying $minsup$ and $minconf$. Subsequently, the algorithm classifies each tuple into the predicted class label from the derived rules, comparing the predicted class label with the actual class label. Finally, the ratio between the number of the correct prediction and the total number of the tuples is computed as the CCR of such the attribute.

For the sake of clarity, we present an example of the CCR computation. Considering the input dataset in Table 4, we have 2 attributes which are A_1 and A_2 . The class label is denoted as C . Suppose that the $minsup$ is set at 0.43 (3 tuples), and the $minconf$ is set at 75%. For the CCR of A_1 , the algorithm begins by scanning the dataset once to discover the set of one-literal rules satisfying $minsup$ and $minconf$. We can see that such the rules are $(A_1 = 0) \rightarrow 1$ which its support is 3 tuples with 100% confidence, and $(A_1 = 1) \rightarrow 0$ which has 3 supporting tuples with 75%. Subsequently, the dataset is scanned for another time to predict the class labels. Such the set of rules can correctly predict 6

Input: D : a dataset Q_D : a quasi-identifier $minsup$: a minimal support threshold k : a condition for k -Anonymity $dom^*(Q_D)$: a generalization on Q_D **Output:**

D' : the output dataset, which satisfy the k -Anonymity property,
and the frequency-based classification impact on data quality
is potentially minimal.

Method:

- 1 Sort all attribute into sequence S by the CCR of each attribute increasingly
- 2 Let A_j be the first attribute in S .
- 3 **while** D does not satisfy k -Anonymity
- 4 generalize D to D' using $dom^*(A_j)$,
- 5 where $dom^*(A_j) \succ_G dom(A_j)$ and there is no other $dom'^*(A_j)$,
- 6 $dom'^*(A_j) \prec_G dom^*(A_j)$.
- 7 **if** the height of $dom^*(A_j)$ reaches the height of the hierarchy of A_j **then**,
- 8 Let A_{j+1} be next attribute in S .
- 9 **end while**

Fig. 2. Minimum Classification Correction Rate Transformation algorithm**Table 4.** An example dataset

Tuple ID	A_1	A_2	C
1	0	0	1
2	0	1	1
3	0	0	1
4	1	1	0
5	1	0	0
6	1	1	0
7	1	1	1

tuples out of 7. Therefore, the CCR of A_1 is $6/7$ or 86%. In this way, the CCR of A_2 which is $3/7$ or 57% can be computed. From this example, A_2 will be at the first rank of the sequence S followed by A_1 .

After the sorting, the algorithm selects an attribute to be generalized from the sorted sequence of the attributes. The generalization is performed in depth-first-search manner, i.e. it selects an attribute, subsequently, generalize the dataset until the dataset satisfies k -Anonymity. If the dataset has not been satisfied, but the selected attribute has reached the height of its hierarchy, the algorithm will select the next attribute from the sequence. The intuition behind this is to avoid excessive generalization of some important attributes, i.e. the attribute with higher CCR could generate the rules since its corresponding one-literal rules are satisfying the $minsup$ and $minconf$ thresholds. Therefore, such the attributes will be more important for associative classification model building.

If the algorithm has more than one choice of attributes to be generalized, it will generalize the attribute with less CCR.

The complexity of the algorithm is composed by two steps: sorting, and generalization. It can be seen that the sorting step requires $2 \times n \times k + k \log k$ cost, where n is the number of the tuples, and k is the number of the attributes. For the generalization step, the complexity of this algorithm is $h \times k \times (n + n \log n)$ where h is the height of the attribute with the highest hierarchy. The $n + n \log n$ cost comes from the k -Anonymity verification which the dataset is first sorted by the attribute values in Q_D . Subsequently, the dataset is scanned once for the anonymity verification. Finally, the algorithm requires $2 \times n \times k + k \log k + h \times k \times (n + n \log n)$ cost which is $O(n \log n)$, when n is large.

4 Experimental Validation

In this section, we present the experiments to validate the proposed algorithm both in terms of effectiveness and efficiency. The effectiveness of the proposed algorithm is validated by the C_{FCM} and the CCR. For the efficiency, the execution time of the proposed algorithm is evaluated. Note here that the results reported in this section are five-time-average results.

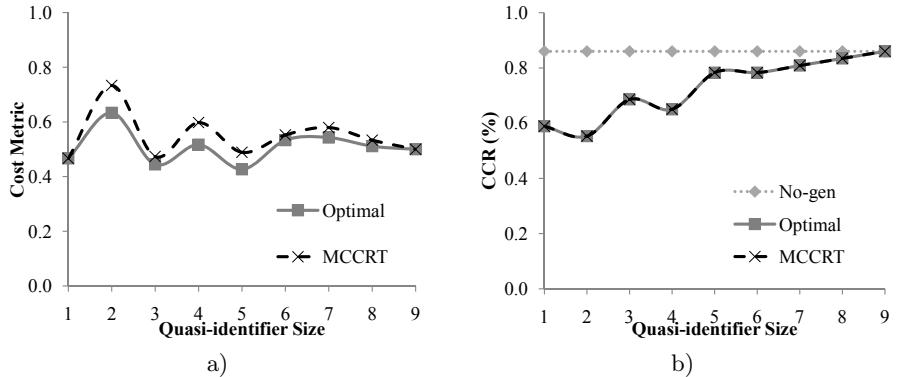
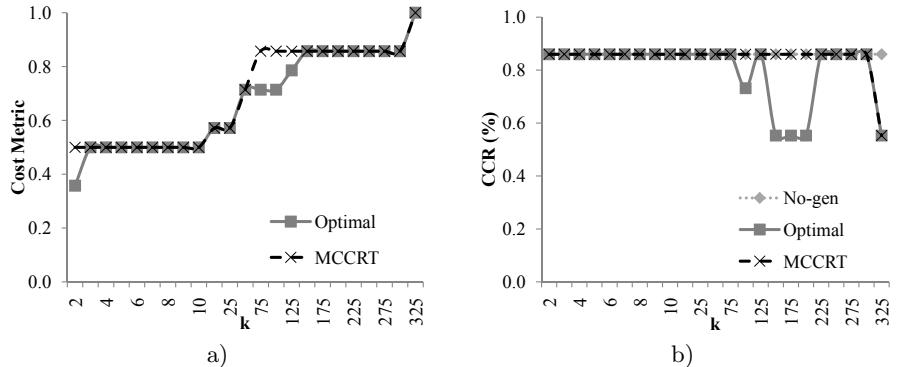
We evaluate our proposed work using the “crx” dataset from UCI repository [14]. The dataset is pre-processed by removing the tuples with unknown values, and discretizing continuous attributes. The crx dataset has 9 attributes for quasi-identifier, and 671 tuples.

The experiments are conducted on an 3.00 GHz Intel Pentium D PC with 3 gigabytes main memory running Microsoft Window Vista. We compare the proposed algorithm with an algorithm which generates optimal solutions (minimum impact C_{FCM}). Both the optimal algorithm and the proposed algorithm are implemented by using JDK 5.0 based on Weka Data Mining Software.

4.1 Effectiveness

In this section, we investigate the the effectiveness of the algorithm by the size of quasi-identifier and the k value. Both algorithms transform the datasets on a specific quasi-identifier until reach a specific k value. Then, we present the corresponding impact metric of each experiment. For the CCR, we build the set of associative classification rules from the transformed dataset to classify the testing datasets as explained in [11]. The CCR of the generalized data will be compared with the “No-gen” CCR where the classifiers are derived from the datasets without any transformation. In all experiments, the $minsup$ are fixed at 30% and $minconf$ at 50%.

Figure 3 shows the results when the size of quasi-identifier is varied, in this experiment, k is fixed at 4. In Figure 3a), we can see that the impact metric C_{FCM} of the proposed algorithm is slightly higher than the optimal algorithm. While, in Figure 3b), it can be seen that our proposed $n \log n$ algorithm performs as effectively as the optimal algorithm which requires exponential run-time. From

**Fig. 3.** Effect of the size of QI**Fig. 4.** Effect of the k value

the figure, we can see that the CCRs of both algorithms are improved when the size of quasi-identifier increases. This is because, with large size of quasi-identifier, the algorithm has more choices of attribute to be generalized.

We present the effect of the k value in Figure 4, in this experiment, the size of quasi-identifier is set at the maximum value. From Figure 4a), we can see that when the k value is increased, the impact metric also increases. The ration behind this is because the crx dataset is very dense. From Figure 4b), we can see that the impact metric could be even 1.00 when the k value is set at 325. This means that all attributes that are related to frequency-pairs have been generalized to the highest level.

From Figure 4b), we can see that the k value affects the CCR significantly, i.e. when the k increases, the CCR is decreased sharply. From the results, the CCR of the optimal algorithm starts to decrease after the k value has been increased to 75. Subsequently, the CCR is not very consistent. This is because of there is an important with regard to the classification in the dataset. As the optimal algorithm treats it as equal as the other attributes in Q_D , when it is

excessive generalized, the CCR will be dropped sharply. However, the CCR of the proposed algorithm starts to drop when k has been increased to 325 which is higher than the optimal algorithm. This is because the proposed algorithm has put the important attribute at the last element of the sequence. Therefore, when the k value has been increased to 325, the proposed algorithm has no choice, but has to generalize the attribute to the level which degrades the CCR. To summary here, the results in term of the CCR of the proposed algorithm is better than the optimal algorithm when the k value is increased.

4.2 Efficiency

In this section, the efficiency of the proposed algorithm, i.e. the execution time is considered. We investigate the efficiency to see the size of quasi-identifier attributes, the k value and the size of input dataset. In these experiments, when the effect of the size of the quasi-identifier is considered, k value is fixed at 4. The size of quasi-identifier is set at maximum value, when we consider the effect of the k value. And, when the effect of the size of input dataset is considered, we fix the k value at 20, and the size of quasi-identifier is set at the maximum value.

In Figure 5a), the proposed algorithm uses much less time comparing with the optimal algorithm when the size of quasi-identifier increases. The ration behind this is because the optimal algorithm must explore the whole search space to find the solution. When the size of quasi-identifier is increased, the search space will also be increased exponentially. From the figure, we can see that the execution time of the proposed algorithm also increases slightly, this effect is caused by the constant in the complexity of the algorithm. In Figure 5b), the effect of the k value is shown. We can see that the execution time of the optimal algorithm is much higher than the execution time of the proposed algorithm. This reason, which the execution time of the proposed algorithm is slightly increased when the k value is increase, is also the constant in the complexity as the result in Figure 5a). Finally, the effect of the size of the dataset is presented in Figure 5c). We can see that the execution time of the proposed algorithm is in the order of $n \log n$ while the execution time of the optimal algorithm is in exponential time order.

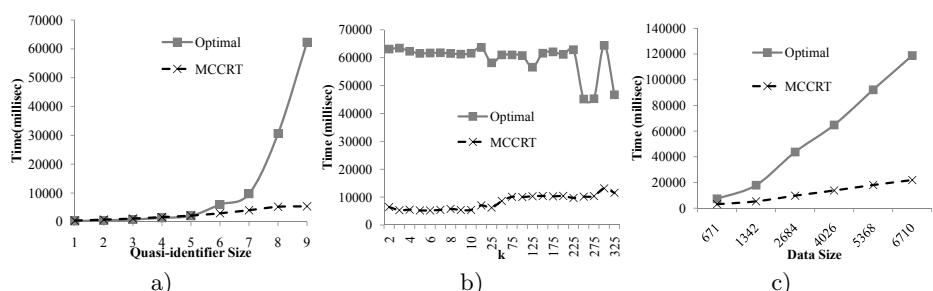


Fig. 5. Effect of the execution time

5 Conclusion

In this paper, we have addressed the problem of privacy preservation using k -Anonymity privacy model in the context of associative classification. We have proposed the heuristic algorithm, MCCRT, for such the problem. The algorithm, which is guided by the CCR of the attributes in the given dataset, have been investigated by the experiments both in terms of effectiveness and efficiency. The experiment results have shown that the proposed algorithm can produce the transformed datasets with data quality. Also, the CCRs of the transformed datasets are very high comparing with the optimal solution. For the efficiency, the proposed algorithm, which its complexity is in the order of $O(n \log n)$, uses much less time than the optimal algorithm in terms of the size of quasi-identifier, the k value, and the size of the dataset.

Acknowledgement

Our work was funded in part by Computer Engineering Department, Faculty of Engineering, Chiang Mai University, as well as Postharvest Technology Innovation Center, Thailand.

References

1. Sweeney, L.: k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* 10, 557–570 (2002)
2. Fung, B.C.M., Wang, K., Yu, P.S.: Top-down specialization for information and privacy preservation. In: Proceedings of the 21st International Conference on Data Engineering, pp. 205–216. IEEE Computer Society, Los Alamitos (2005)
3. Bayardo Jr., R.J., Agrawal, R.: Data privacy through optimal k-anonymization. In: Proceedings of the 21st IEEE ICDE International Conference on Data Engineering, pp. 217–228. IEEE Computer Society, Los Alamitos (2005)
4. Li, J., Wong, R.C.W., Fu, A.W.C., Pei, J.: Achieving k-anonymity by clustering in attribute hierarchical structures. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2006. LNCS, vol. 4081, pp. 405–416. Springer, Heidelberg (2006)
5. Meyerson, A., Williams, R.: On the complexity of optimal k-anonymity. In: Proceedings of the Twenty-third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 223–228. ACM, New York (2004)
6. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* 10, 571–588 (2002)
7. Truta, T.M., Campan, A.: K-anonymization incremental maintenance and optimization techniques. In: SAC 2007: Proceedings of the 2007 ACM symposium on Applied computing, pp. 380–387. ACM, New York (2007)
8. Wang, K., Fung, B.C.M.: Anonymizing sequential releases. In: KDD 2006: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 414–423. ACM Press, New York (2006)
9. Wang, K., Yu, P.S., Chakraborty, S.: Bottom-up generalization: A data mining solution to privacy protection. In: Proceedings of the 4th IEEE International Conference on Data Mining, pp. 249–256. IEEE Computer Society, Los Alamitos (2004)

10. Li, W., Han, J., Pei, J.: Cmar: Accurate and efficient classification based on multiple class-association rules. In: Proceedings of the 2001 IEEE ICDM International Conference on Data Mining, Washington, DC, USA, pp. 369–376. IEEE Computer Society, Los Alamitos (2001)
11. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proceedings of the fourth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 80–86. AAAI Press, Menlo Park (1998)
12. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: SIGMOD 1993: Proceedings of the 1993 ACM SIGMOD international conference on Management of data, pp. 207–216. ACM Press, New York (1993)
13. Harnsamut, N., Natwichai, J., Sun, X., Li, X.: Data quality in privacy preserving for associative classification. In: Proceedings of the Third International Conference on Advanced Data Mining and Applications (to appear, 2008)
14. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)

Time–Frequency Analysis of Vietnamese Speech Inspired on Chirp Auditory Selectivity

Ha Nguyen and Luis Weruaga

Commission for Scientific Visualisation*

Austrian Academy of Sciences

Donau-City Strasse 1, 1220 Vienna, Austria

{ha.nguyen,luis.weruaga}@oeaw.ac.at

<http://www.viskom.oeaw.ac.at/>

Abstract. In speech analysis, the pitch or fundamental frequency is usually considered as parameter for characterizing the vocal chord excitation, but it plays nearly no role in the very time–spectral analysis of the speech signal. In this paper, we present a novel speech analysis approach in which pitch (and its variation over time) play a leading role. The computation of the pitch and the pitch rate is carried out in-segment, by means of the minimization of Huber’s loss over the short-time correlation according to a second-order polynomial fitting law. The proposed method is integrated within the Fan-Chirp transform and the Spectral All-Pole Estimation method, both proposed previously by the authors. The results over Vietnamese speech reveal the advantages of the proposed analysis methodology versus the popular linear prediction estimation. The paper discusses finally the possible impact of the proposed method in speech coding, this representing the upcoming research work.

Keywords: Pitch-driven time–frequency analysis, frequency-selective AR estimation, speech coding.

1 Introduction

In most of the speech coding techniques, the long term prediction is carried out after the spectral analysis step [1]. For instance, in CELP coders and its variants, the pitch or fundamental frequency is obtained from the resulting linear-prediction-coding (LPC) residue, while in sinusoidal or harmonic coding, it is obtained from the Fourier spectrum of the segment based on its harmonic structure. However, this way of proceeding presents some drawbacks, namely, the LPC analysis delivers inaccurate results with the presence of periodicity within the analysis [2], and in-segment variations of the pitch cause a blurry harmonic

* This work has been partially supported by Grant 812120-SCK/SAI of the *Österreichische Forschungsförderungsgesellschaft*.

representation in the Fourier spectrum [3], this compromising the accurate representation of the speech segment by means of plain sinusoids. These limitations are especially severe with austroasiatic languages, such as Vietnamese, which possess a comparatively large number of vowels, in which semantic meaning depends also on the “tone”, i.e., pitch contour within the vowel. Thus, the pitch information should be used as input to the spectral analysis stage, rather than just a result thereof. Pitch-dependent spectral analysis has not received much attention so far, due to the simplicity and reasonable performance of LPC and Fourier analyses (we can cite the work [4], an elaborate pitch-synchronous harmonic analysis, as the most popular among them.)

A pitch-driven time–frequency analysis technique developed by one of the authors in [3], the fan-Chirp transform (FChT), is an interesting alternative to the Fourier analysis. The FChT delivers a fine spectral representation of voiced speech segments with fast variation of the fundamental frequency, this method requiring the value of the pitch rate (relative change of the pitch over time). The FChT has an analysis basis composed of harmonically-related linear chirps, thus emulating the chirp sensitivity that takes actually place in the auditory system [5]. The FChT has been recently used in harmonic speech coding [6] and single-channel speech separation [7]. On the other hand, the mentioned inaccuracy of the LPC analysis of harmonic signals can be alleviated by the Spectral All-Pole Estimation (SAPE) technique [8], proposed by the authors. SAPE is a frequency-selective autoregressive (AR) estimation method, suitable for harmonic signals such as speech voiced utterances. Loosely speaking, it obtains the AR model that best fits at the spectral location of the harmonics, which are apart from each other by the fundamental frequency. The combination of both techniques for speech analysis has not been proposed so far.

This work addresses the use of the FChT and SAPE in the analysis of Vietnamese speech. Vietnamese was chosen due to its reach variety of tones and voiced sounds, which represent a handicap to the classical LPC and Fourier analyses. Although the theoretical background of both techniques has been well studied, its combination, as well as the joint estimation of the pitch and the pitch rate, need study. The estimation of the pitch is a scrutinized problem [1]. However that of the pitch rate, i.e., its variation over time, has not been sufficiently explored. We propose here a novel in-segment approach for the estimation of both parameters over short segments (20-30 ms) of voiced speech, based on combining correlation among segment sub-blocks and Support Vector Machine (SVM) theory. SVMs, increasingly popular in artificial intelligence, are suitable here for its ability at discriminating outliers that may be present in the long-term information provided by the correlation. The paper is divided as follows: Sec. 2 contains a review of the FChT and SAPE techniques; Sec. 3 presents the SVM-based pitch/-rate in-segment estimation method; Sec. 4 illustrates the performance of the proposed method on real Vietnamese speech; Sec. 5 presents an analysis on the pros and cons of the method, discussing as well the adequacy of using pitch information in the analysis stage of a speech coder.

2 Background

2.1 Pitch-Variant Speech Model and Fan-Chirp Transform

A pitch-variant voiced speech segment is described in a simplified way as

$$s(t) = \sum_i h(t - t_i) \quad (1)$$

where t is time, $h(t)$ is the impulse response of the vocal tract/glottal pulse, and t_i represents the instants of glottal pulse production ($t_i > t_{i-1}$). By adopting the assumption of linear variation of the pitch within the segment, the instants t_i must fulfill the following second-order polynomial rule

$$f_o \left(1 + \frac{1}{2}\gamma t_i\right) t_i = \eta_i \quad (2)$$

where γ is the so-called pitch rate, and f_o represents the “instantaneous” fundamental frequency at $t = 0$. We assume that segment $s(t)$ is centered at $t = 0$, index i running from small negative integers to positive integers, while the period order η_i is such that $\eta_i - \eta_{i-1} = 1$.

The non-stationary model (1) has been proven to describe accurately short segments of naturally-intonated speech [3]. Assuming that the values f_o and γ are known (Sec. 3 presents a method to estimate them from segment $s(t)$), the fan-chirp transform, defined as

$$X(f, \gamma) = \int_{-\infty}^{\infty} s(t) \sqrt{|1 + \gamma t|} e^{-j2\pi f(1 + \frac{1}{2}\gamma t)t} dt \quad (3)$$

is proven to deliver a very detailed harmonic representation in comparison to the Fourier transform. The fan-chirp transform yields intrinsically the marginalization of the time–frequency space according to a fan geometry, as illustrated in Fig. 1. This geometry is similar to that of short segments of naturally-intonated

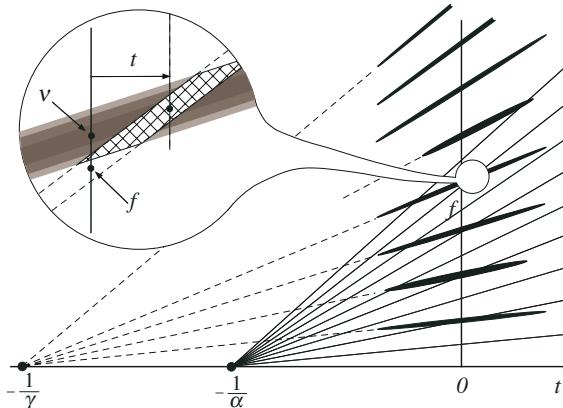


Fig. 1. Fan-chirp transform and marginalization of the time–frequency plane

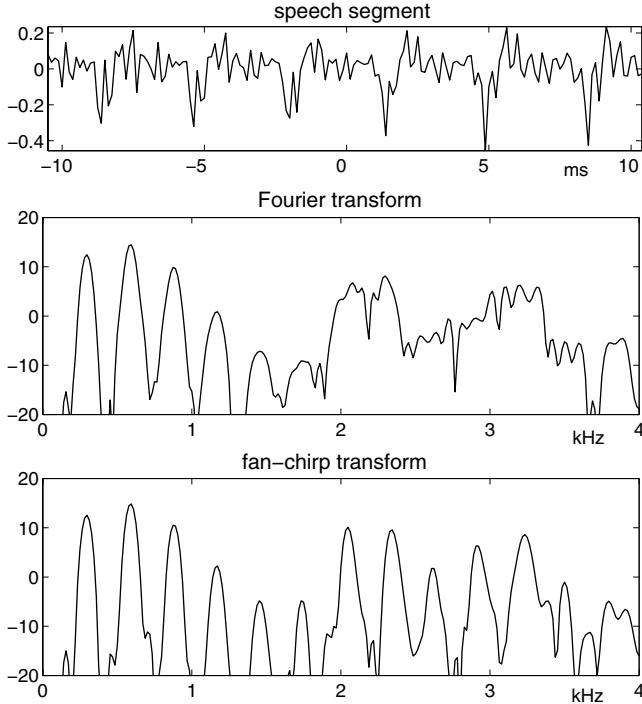


Fig. 2. Fourier versus fan-chirp transform

speech, the FChT being thus especially indicated in its analysis. This fact is illustrated in Fig. 2 on a real example. Although the FChT basis is composed of linear chirps, its implementation can be carried out with a Fourier transform over the time-warped segment. This hint points to the efficient numerical evaluation of the FChT with discrete-time signals (see [3] for detailed information).

2.2 Spectral All-Pole Estimation

The spectrum of voiced speech segments are characteristic by its harmonic structure, which yields the undersampling of the vocal tract spectral envelope. This case of “missing” spectral information is well-known to be a source of inaccuracy in autoregressive estimation [2]: the inter-harmonic areas, which do not contain valuable energy, harm implicitly the LPC solution. This undesirable effect is clearly more pronounced in high-pitched speech segments. Rather than using all spectral samples, one would wish to use only those samples with valuable information, i.e., those corresponding to harmonic locations. This frequency-selective AR estimation can be approached with the spectral all-pole estimation (SAPE) [8]. SAPE solution results from the minimization of the Whittle likelihood [9]

$$\mathcal{L} = \int_{-\pi}^{\pi} A(\omega) \left(|S_\gamma(e^{j\omega})|^2 |A(e^{j\omega})|^2 - \log |A(e^{j\omega})|^2 \right) d\omega \quad (4)$$

where $S_\gamma(e^{j\omega})$ is the discrete-time fan-chirp transform of the speech analysis segment, $A(z)$ is the P -order normalized linear residue predictor

$$A(z) = \sum_{k=0}^P a_k z^{-k} \quad (5)$$

and $\Lambda(\omega)$ is the so-called spectral mask. The spectral mask is non-negative, containing the relevance of each spectral sample. Since the relevant information here is located at the harmonics, the spectral mask is set to

$$\Lambda(\omega) = \sum_k \delta(\omega - k\omega_o) \quad (6)$$

where $\delta(\omega)$ is the Dirac delta and ω_o represents the effective pitch in the segment.

The minimization of \mathcal{L} with respect to AR coefficients a_k is a convex problem and can be solved numerically with a computationally efficient algorithm. Details of the numerical implementation can be found in [8]. The proposed combination of FChT and SAPE turns out to be an alternative to the LPC estimation in case of voiced sounds with in-segment pitch contours. In this regard Fig. 3 provides an illustrative example on real speech: the LPC solution clearly

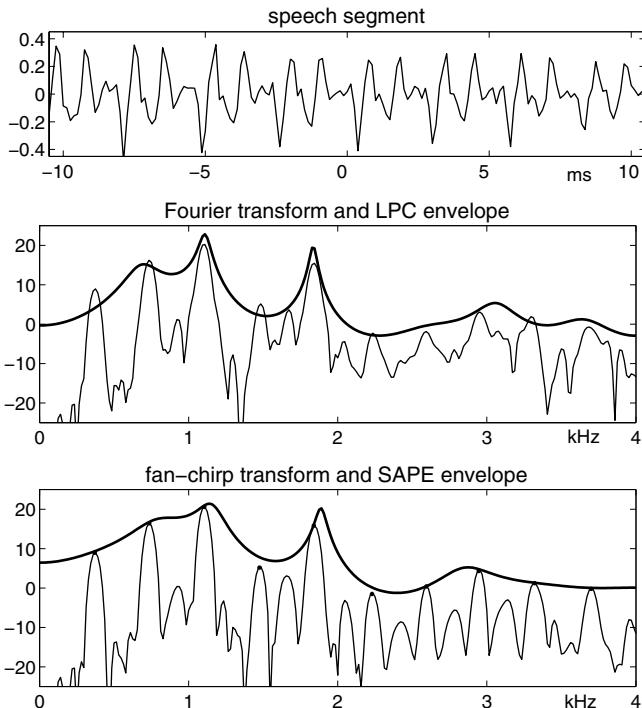


Fig. 3. Classical analysis versus pitch-driven speech analysis

overestimates the dominant harmonics, while SAPE delivers an AR model that delineates consistently the harmonic energy.

3 Pitch and Pitch-Rate In-Segment Estimation

The pitch and pitch rate within the segment are input parameters required in the FChT and SAPE techniques. An intra-frame technique to estimate both was proposed in [10]: the pitch rate was simply obtained as the difference between the pitch of two consecutive segments. This approach is however prone to errors especially in onsets and offsets of the voiced sounds, and it implies a look-ahead processing, not desirable in real-time operation.

This section presents an in-frame technique thereto, that is, pitch and pitch rate are obtained only from the information in the very short analysis segment. The estimation of pitch according to the non-stationary model (1),(2) implies estimating simultaneously the pitch rate γ . This estimation problems reduces to obtaining at least three time glottal instants t_i , since the coefficients of the best fit second-order polynomial (2) yields immediately the values of pitch and pitch rate. Thus, the main focus is the detection of the actual glottal time instants t_i , and the final computation of f_o and γ thereof.

3.1 Selecting Candidate Glottal Instants

Despite the analysis speech being a discrete-time signal, it results from the sampling of the analogue signal $s(t)$, and thus the glottal instants need not correspond to integer samples. Therefore, for the sake of simplicity, we will use the continuous formulation henceforth. In speech coding, the long-term prediction lag is commonly obtained from the signal or the perceptually-enhanced LPC residue by cross-correlation among subblocks

$$\rho_\ell(t) = \left| \frac{\int_0^B s(t + \tau) s(\ell + \tau) d\tau}{\sqrt{\int_0^B s(t + \tau)^2 d\tau \int_0^B s(\ell + \tau)^2 d\tau}} \right| \leqslant 1 \quad (7)$$

where B is the length of the subsegment, located at time instant $t = \ell$, and $\lfloor \cdot \rfloor_0$ denotes low-clipping by 0.

In voiced segments, $\rho_\ell(t)$ presents several spikes of high value (closer to one), related to the glottal pulse occurrence. Unfortunately, not all peaks necessarily correspond to that desired case:

- the vocal tract impulse response $h(t)$ exhibits a sinusoidal-like shape, which may yield moderate levels of cross-correlation and thus spurious peaks,
- impulsive noise can hide or attenuate actual glottal pulses, and
- the peak related to two periods away may exhibit comparable cross-correlation levels than the contiguous one [11].

The previous facts difficult the localization of the actual glottal instants. Instead of taking a hard decision on the maximum of the cross-correlation (7), we proceed as follows:

1. Select the first glottal instant as $\tau_0 = \arg \max\{|s(t)|, t\}$ (in a voiced segment, this rule points out very likely to the beginning of a glottal pulse).
2. Compute the cross-correlation with the subblock at $\ell = \tau_0$.
3. Select the L largest peaks to the left and right (L is usually set to three).
4. Stop if there are no peaks or the segment limit has been reached.
5. Compute the cross-correlation with the subblocks placed on the previous peak locations. Go to step 3.

The set of candidate positions resulting from the previous process along with their respective cross-correlation values $\rho_\ell(t_i)$ hold the actual information of the pitch and pitch rate in the segment. The question now lies on how to disregard the spurious peaks and detect the actual ones, so that the fitting rule (2) can be properly applied. The solution to this problem is studied in the next section.¹

3.2 SVM-Based Pitch and Pitch-Rate Estimation

Let us consider the peaks resulting in a given subframe cross-correlation $\rho_\ell(t)$. The maximum clearly corresponds at $\rho_\ell(\ell)$, while the candidate lags correspond to two cases, those located at the right-hand side of that center, $t_i^+ > \ell$, and those at the left, $t_i^- < \ell$. Note that at most one of the pulses on each side should correspond to the long-term prediction lag, but at this point one does not know which. According to (2), the correct lag must fulfill

$$f_o \left(1 + \frac{1}{2}\gamma t^+\right) t^+ = \eta - 1 \quad (8a)$$

$$f_o \left(1 + \frac{1}{2}\gamma t^-\right) t^- = \eta + 1 \quad (8b)$$

where η is the period order of the central instant ℓ , that is,

$$f_o \left(1 + \frac{1}{2}\gamma\ell\right) \ell = \eta. \quad (9)$$

From (8) and (9), we can define the error of the peak at t_i as

$$e_i = y_i \mathbf{w}^T \mathbf{x}_i - 1 \quad (10)$$

where T denotes transpose, and

$$y_i = \text{sign}(t_i - \ell) \quad (11a)$$

$$\mathbf{w} = [\gamma f_o, f_o] \quad (11b)$$

$$\mathbf{x}_i = \left[\frac{1}{2}t_i^2 - \ell^2, t_i - \ell \right]. \quad (11c)$$

¹ One previous remark on the continuos-time formulation and the fact that the analysis signal is discrete-time: the peak positions are actually computed with subpixel accuracy from the discrete cross-correlation; the details on the interpolation technique are omitted here since they represent a straightforward implementation exercise.

By considering the information from all cross-correlation instances, the estimation of f_o and γ can be thus revamped as the minimization of the following functional

$$\mathcal{J} = \sum_{i,\ell} \rho_\ell(t_i) \mathcal{H}_\sigma \left(y_i^{(\ell)} \mathbf{w}^T \mathbf{x}_i^{(\ell)} - 1 \right) + \lambda \gamma^2 \quad (12)$$

where superscript (ℓ) is associated to the results from the cross-correlation $\rho_\ell(t)$, the coefficient $0 \leq \rho_\ell(t_i) < 1$ represents the confidence of each candidate, λ is the regularization constant, and $\mathcal{H}_\sigma(e)$ denotes Huber's robust loss

$$\mathcal{H}_\sigma(e) = \begin{cases} \frac{1}{2\sigma}(e)^2, & \text{if } |e| \leq \sigma \\ |e| - \frac{\sigma}{2}, & \text{otherwise} \end{cases}. \quad (13)$$

Functional \mathcal{J} has two terms: the first one pursues fitting the second-order model (2) to the dataset while simultaneously discarding the outliers or anomalous peak locations (a well-known property of the Huber's loss [12]), and the second term forces the solution to correspond to a moderate γ value, as is the case of naturally-intonated speech in practice. The margin in the Huber's loss is set to a low value $0 < \sigma \ll 1$.

Functional (12) resembles closely the primal function in support-vector regression (SVR) [12].² Given that the size of the dataset exceeds by far the dimension of the plane $\mathbf{w} = [w_1, w_2]$ (only two parameters), we choose to use the Iterative Re-Weighted Least Squares (IRWLS) [13] rather than the computationally-expensive quadratic programming. The IRWLS is based on expressing (14) in the form of a least-square (LS) functional as

$$\mathcal{J} = \sum_{i,\ell} \alpha_i^{(\ell)} \left(y_i^{(\ell)} \mathbf{w}^T \mathbf{x}_i^{(\ell)} - 1 \right)^2 + \hat{\lambda} w_1^2 \quad (14)$$

where

$$\hat{\lambda} = \lambda / w_2^2 \quad (15a)$$

$$\alpha_i^{(\ell)} = \frac{\rho_\ell(t_i)}{|y_i^{(\ell)} \mathbf{w}^T \mathbf{x}_i^{(\ell)} - 1|_\sigma}. \quad (15b)$$

Here $|_\sigma$ denotes low-clipping by σ . The IRWLS achieves iteratively the solution by solving the LS problem (14) and then recomputing the weights according to (15). The final value of \mathbf{w} yields the estimated pitch f_o and pitch rate γ according to (11). In the practice, few iterations suffice to reach the solution.

4 Examples: Analysis of Vietnamese Speech

We conducted several experiments with the proposed method on Vietnamese speech. The analysis segment was set to 24 ms. This segment was extended with

² The only difference refers to the regularization term, which does not actually correspond to $\|\mathbf{w}\|^2$, but to $\gamma^2 = (w_1/w_2)^2$.

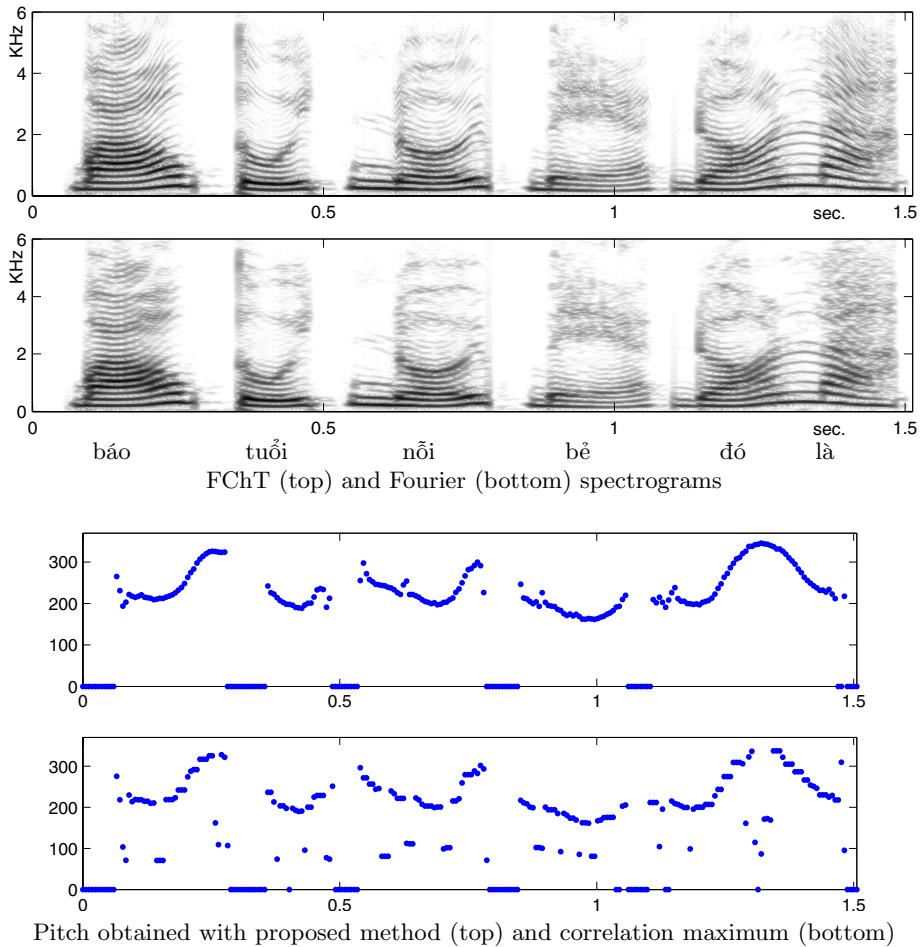


Fig. 4. Pitch-driven time–frequency analysis of female Vietnamese speech

the samples from the previous segment, in order to allow for the detection of the pitch and pitch rate in low-pitched segments. The parameters were set as follows: the regularization constant to $\lambda = 10^4$, and Huber's loss margin to $\sigma = 0.02$.

Fig. 4 and Fig. 5 show graphical results on Vietnamese female and male speech respectively. On each figure the Fourier- and FChT-spectrograms are shown, in which each time instance corresponds to a time shift of 3 ms. The FChT-spectrogram is driven by the output of the pitch and pith rate estimation mechanism, whose results are show below the spectrograms.

Visual analysis on the spectrograms reveals immediately better representation of the fine-spectral representation of Vietnamese speech achieved by the FChT analysis, this pointing out to an accurate estimation of the pitch rate for each signal frame. On the other hand, the pitch estimation of the proposed method turns out very reliable and precise when compared to the classical method of

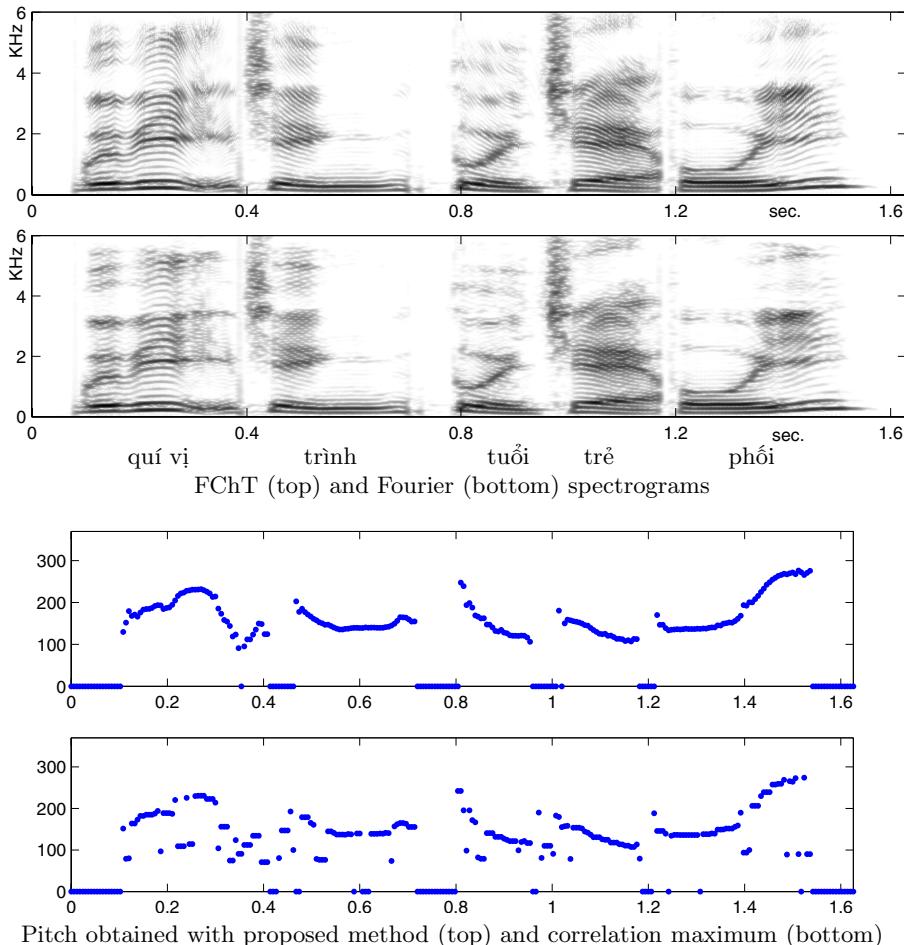


Fig. 5. Pitch-driven time-frequency analysis of male Vietnamese speech

correlation maximum, commonly used in some speech coding algorithms. It is interesting to remark the continuity of the estimated pitch, this not being the case in the classical method, especially in low-pitched segments (Fig. 5). In case of sudden utterance transitions and on- and offsets the proposed method achieves reasonable performance (further explanations on Sec. 5).

The pitch-driven sampled FChT spectrum along was fed into the SAPE spectral estimation method. In Fig. 6 we bring illustrative results on one time instance of the male record (for $t \simeq 0.5$). Apart from the known fine representation of the FChT versus Fourier, the SAPE and LPC estimation differ mainly in the estimation of the formants: the LPC overestimates the energy of the formant, while SAPE provides a natural interpolation of the harmonic energy.

Although the combination of FChT and SAPE turns out promising in speech analysis, it is necessary to point out its main drawback: the FChT yields a

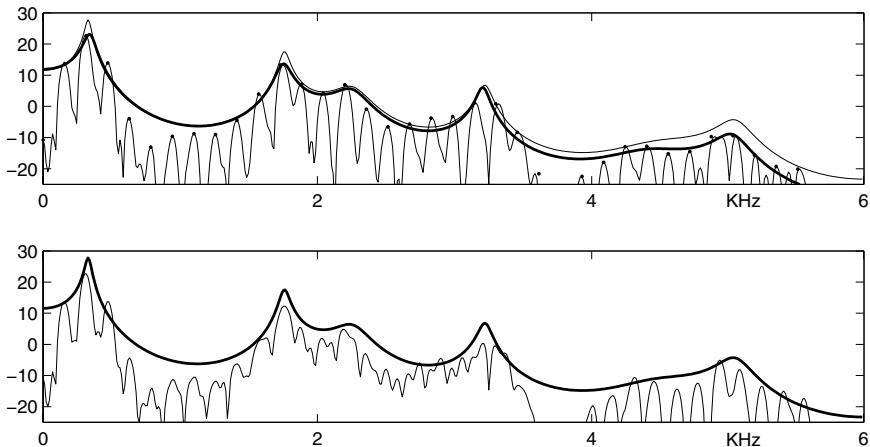


Fig. 6. Spectral analysis of Vietnamese speech: top – FChT, and SAPE (thick) and LPC (thin) AR estimations; bottom – Fourier transform and LPC AR estimation

smoothed vocal tract spectral envelope, especially in high-frequency formants (this fact has been documented in [10]), which harms the estimation of the vocal tract as an AR model. This point is not only noticeable in Fig. 6, but in other segments with large pitch variation. Next section discusses the tentative solution to this last point as well as our further research lines.

5 Discussion

Two issues are worth discussing here: the performance of the proposed pitch/rate estimation technique, and the (undesired) intrinsic spectral envelope smoothing caused by the FChT. This last point represents a serious drawback on the vocal tract estimation with fast pitch contours, which is the case of Vietnamese speech. We are currently approaching a solution thereto, based on a different warping rule in the FChT that does not distort the vocal tract impulse response. While conclusive results are not available at this point, our preliminary investigations point out to a promising outcome.

Regarding the proposed pitch and pitch rate estimation method, the main point to work on refers to the pitch estimation on utterance transitions and on/offsets. In fact, the proposed method (as others based on correlation) estimate pitch by exploiting periodicity in the signal; however, periodicity vanishes when the vocal tract impulse response abruptly changes, even if the pitch remains constant. Based on this reasoning we are currently working on an improved technique for pitch (rather than periodicity) and pitch rate estimation. Just mentioning that the commonly-used enhanced LPC residual does not represent a de-facto solution in the abrupt utterance transition.

The previous background techniques are planned to be integrated in a speech coder, with special emphasis on increasing robustness to background noise. Not

only the FChT delivers a low entropic spectral representation, but the spectral mask in SAPE can be set conveniently to disregard noise-corrupted spectral regions. The performance of the resulting noise-robust speech coder is planned to be assessed with commercial and in-house speech recognition systems.

References

1. Kondoz, A.M.: Digital speech: Coding for low bit rate communication systems. John Wiley & Sons, Chichester (2004)
2. Quatieri, T.F.: Discrete-Time Speech Signal Processing. Prentice-Hall, Englewood Cliffs (2001)
3. Weruaga, L., Képesi, M.: The fan-chirp transform for nonstationary harmonic signals. *Signal Processing* 87, 1504–1522 (2007)
4. Kawahara, H., et al.: Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. *Speech Communication* 27, 187–207 (1999)
5. Mercado, E., Myers, C.E., Gluck, M.A.: Modeling auditory cortical processing as an adaptive chirplet transform. *Neurocomputing* 32(33), 913–919 (2000)
6. Dunn, R., Quatieri, T.F.: Sinewave analysis/synthesis based on the fan-chirp transform. In: Proc. IEEE WASPAA, pp. 247–250 (2007)
7. Li, P., Guan, Y., Xu, B., Liu, W.: Monaural speech separation based on computational auditory scene analysis and objective quality assessment of speech. In: Proc. IEEE ICASSP, pp. 2014–2023 (2008)
8. Weruaga, L.: All-pole estimation in spectral domain. *IEEE Trans. Signal Processing* 55, 4821–4830 (2007)
9. Whittle, P.: Gaussian estimation in stationary time series. *Bull. Intl. Stat. Instit.* 39, 105–130 (1961)
10. Képesi, M., Weruaga, L.: Adaptive chirp-based time-frequency analysis of speech signals. *Speech Communication* 55, 474–492 (2006)
11. Marques, J.S., et al.: Improved pitch prediction with fractional delays in CELP coding. In: Proc. IEEE ICASSP, pp. 665–668 (1990)
12. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press, Cambridge (2002)
13. Rojo-Álvarez, J.L., et al.: A robust support vector algorithm for nonparametric spectral analysis. *IEEE Signal Processing Lett.* 10, 320–323 (2003)

Meta-level Control of Multiagent Learning in Dynamic Repeated Resource Sharing Problems

Itsuki Noda^{1,2,3} and Masayuki Ohta¹

¹ Information Technology Research Institute

National Institute of Advanced Industrial Science and Technology

1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan

{i.noda, m-ohta}@aist.go.jp

² School of Information Science

Japan Advanced Institute of Science and Technology, Japan

³ Department of Systems Innovation, The University of Tokyo, Japan

Abstract. In this article, we propose two methods to adapt parameters in multi-agent reinforcement learning (MARL) for repeated resource sharing problems(RRSP). Resource sharing problems (RSP) are important and widely-applicable frameworks on MARL. RRSP is a variation of RSP in which agents select resources repeatedly and periodically. We have been proposing a learning method called Moderated Global Information (MGI) for MARL in RRSP. However, we need carefully adapt several parameters in MGI, especially temperature parameter T in Boltzmann selection in agent behavior and modification parameter L , to converge the learning into suitable states. In order to avoid this difficulty, we propose two methods to adjust these parameters according to the performance of each agent and statistical behaviors of agents. Results of several experiments tell us that the proposed methods are robust against changes of environments and force agent-behaviors to the optimal situation.

1 Introduction

Suppose that there are two possible routes to commute daily to our office or school. We may explore both route several times and fix a route for the daily use. In such case, we believe that a chosen route is more comfortable than another. Generally, we learn such a choice and reinforce the belief about its superiority through experience. However, that assessment of comfortable choice might vary according to other people's changes of choices; for example, such changes might cause new traffic congestion patterns or jam-packed trains on our previously chosen route. If many people change their minds often, it is difficult to expect lasting comfort from a choice of any route, so that we give up planning daily commutes and start to behave transiently. Nevertheless, people tend to fix their daily routes in towns where traffic is functioning smoothly. As a result, that collection of fixed behaviors forms implicit rules that support a well-functioning society.

In this article, we focus on the process for the society to achieve such a well-functioning situation. Generally, people decide their choice based on their own

experiences and information given from others like hearsay and traffic reports. We simplify such experience-/information-based decision making as reinforcement learning of agent policy, and discuss how the learning influences and is controlled by the society.

First of all, we formalize these cases as “Repeated Resource Sharing Problems” (RRSPs), that is a kind of simplified congestion games in which each agent chooses one of multiple resources to obtain an increasingly great benefit or reward that varies according to the number of agents who choose that resource (figure 1).

When we consider an RRSP as a control problem, there are two salient aspects: consumer-side aspects and supplier-side aspects. Generally, an RRSP is handled from the consumer-side aspect, in which we specifically address mechanisms to manage agents’ behaviors to reach a social optimum where the total benefit or reward of agents is maximized [1,2]. Related to this aspect, the main issue is how to satisfy agents’ requirements that they believe that they have chosen the best resource. A possible solution for this issue is a Nash equilibrium, wherein we assume that each agent is selfish.

Generally, it is difficult to determine the equilibrium for partially observable open systems like the internet. Therefore, learning or adaptation of agents is required to reach an equilibrium. As the RRSP matures, the agent-learning in RRSP becomes a multi-agent reinforcement learning (MARL) problem, in which each agent learns and changes its behavior using its own and others’ experiences. If each agent can refer to another’s experiences (we call such information ‘global information’) for learning, an *oscillation problem* might occur because all agents tend to choose the same resource when they share their experiences. Consequently, the total benefit of the agents decreases seriously. Such oscillation problems are apparent in several domains like choices of attractions at theme parks, load-balancing in computer-networks, and so on.

Several mechanisms are proposed [3,4,5] for several situations to avoid such problems. We also have proposed *Moderated Global Information* (MGI) method [6], which can be applied to general cases of the oscillation problem. Using MGI method, we can stabilize the agents’ choice to produce a Nash equilibrium. However, these methods do not provide convergence from variable cases or stability against a dynamic environment.

On the other hand, in the supplier-side aspect, the salient issue of RRSP is how to let agents’ choices stable. As shown in the example described above, if all agents fix their choices, it is easy to anticipate the number of agents choosing each resource by observation. Such expectation enables effective re-design and modification of resource allocation or resource logistics. On the other hand, in the case where agents change their choices repeatedly, such re-design becomes difficult so that the total effectiveness might decrease.

To attack issues of both aspects, we focus tuning learning parameters as a control problem of exploration and exploitation in MARL, and propose a meta-level mechanism to control the additional information for a more general and dynamic environment. In the remainder of this paper, we formalize the resource-sharing problem in section 2 and propose two frameworks to control agent learning in

section 3 and section 4. We also discuss about related works and relation between our works and them in section 5, and summarize in section 6.

2 Resource Sharing and Learning Agents

2.1 Repeated Resource Sharing Problem

An RRSP is defined as follows. Presume that there exists a set of resources $\mathbf{R} = \{r_1, r_2, \dots, r_n\}$ and a set of agents $\mathbf{A} = \{a_1, a_2, \dots, a_m\}$. In every discrete time step, the following procedure is executed:

- S.1 Each agent $a_i \in \mathbf{A}$ chooses a resource $r_j \in \mathbf{R}$ solely according to its policy.
- S.2 Each agent a_i who chooses r_j receives a utility that is calculated by the utility function (U_j) of r_j and the number of agents (n_j) who choose r_j .

Here, “solely according to its policy” means that all agents are selfish and introverted. A decision of an agent is never influenced by another’s simultaneous behaviors or intention. We also presume that the policy is without memory, i.e., no decision is influenced directly by past decisions and experiments of the agent. Instead, the policy can be denoted as a probabilistic function $\pi_i(r_j)$, a probability that agent a_i chooses resource r_j . Each agent is supposed to learn its policy using reinforcement learning based on its experiment:

- S.3 Each agent a_i changes its policy $\pi_i(r_j)$ according to the utility $U_j(n_j)$ that the agent obtains.

For simplicity, we use the following framework as the learning mechanism: Each agent a_i has its own estimated utility of resource r_j , which is denoted as $V_i(r_j)$. When the agent receives a utility u_j from resource r_j , the agent modifies its estimation as

$$V_i(r_j) = (1 - \alpha)V_i(r_j) + \alpha u_j, \quad (1)$$

where α ($0 < \alpha < 1$) is the learning rate. The policy of agent $\pi_i(r_j)$ is calculated from $V_i(r_j)$ using Boltzmann softmax function as $\pi_i(r_j) = e^{V_i(r_j)/T} / \sum_{r_k} e^{V_i(r_k)/T}$, where T ($T > 0$) is a temperature parameter. Note that we presume that the utility function U_j of resource r_j is a monotonically decreasing function, because the problem is resource sharing.

2.2 Moderated Global Information

One problem of multi-agent reinforcement learning is the number of learning examples and the variation of situations. Generally, each agent can be considered as an environment for other agents. On the other hand, the learning of an agent who uses only its own experience is slow and only slightly reaches the optimum. In fact, Figure 2 shows an exemplary process of MARL in RRSP. In this figure, the dotted line shows the performance of learning with one’s own experience.

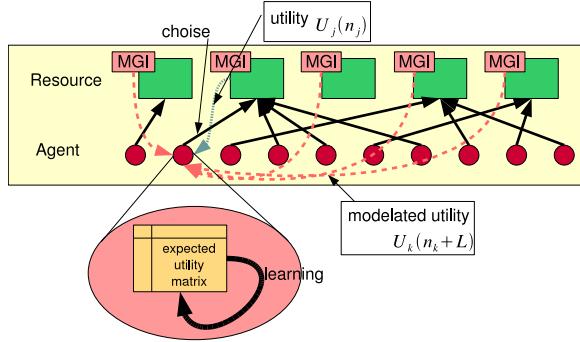


Fig. 1. Repeated Resource Sharing Problem (RRSP) and Moderated Global Information (MGI)

The line climbs gradually (it takes about 50 cycles to become greater than 0.57) and never converges to the optimum (0.5785 in this graph).

Using experiences of others is one solution to speed up learning. In this case, each agent, who is choosing resource r_j , can refer others' experience about utility u_k of resource r_k ($k \neq j$) at this time. Using u_k , the agent can update its estimated utility $V_i(r_k)$ using the same manner of 2.1. However, using crude experiences of others causes another problem, as discussed in the section 1. In this case, all agents learn and use the same information to update $V_i(r_j)$ for all j , so that they come to choose the same resource, whose value $V_i(r_j)$ is the maximum. As a result, the average utility of each agent degrades rapidly because of the congestion to one resource. The thick line in figure 2 shows such phenomena of learning through other's experiences. The line initially inclines quickly (it takes less than 20 cycles to reach 0.57) but declines after 200 cycles.

In order to overcome this problem, we have proposed the usage of *moderated global information* (MGI) [6] (figure 1). In MGI, in addition to learning through self-experimentation, as eq. (1), each agent also adjusts the estimated utilities of resources that the agent does not choose, as

$$V_i(r_k) = (1 - \alpha')V_i(r_k) + \alpha'U_k(n_k + L), \quad (2)$$

where α' ($0 < \alpha' < 1$) is the learning rate, and L ($L \geq 1$) is the moderation factor.

In the case of $L = 1$, $U_k(n_k + L)$ means the utility the agent will receive from resource r_k if only the agent change the resource from r_j to r_k but other agents do not change their resources. In [6], we already have shown theoretically that policies of learning agents can converge into a Nash equilibrium and that they are stabilized at the equilibrium. Using this effect, we demonstrated that the moderated information can reduce the adverse effects caused by rumors of selfish agents.

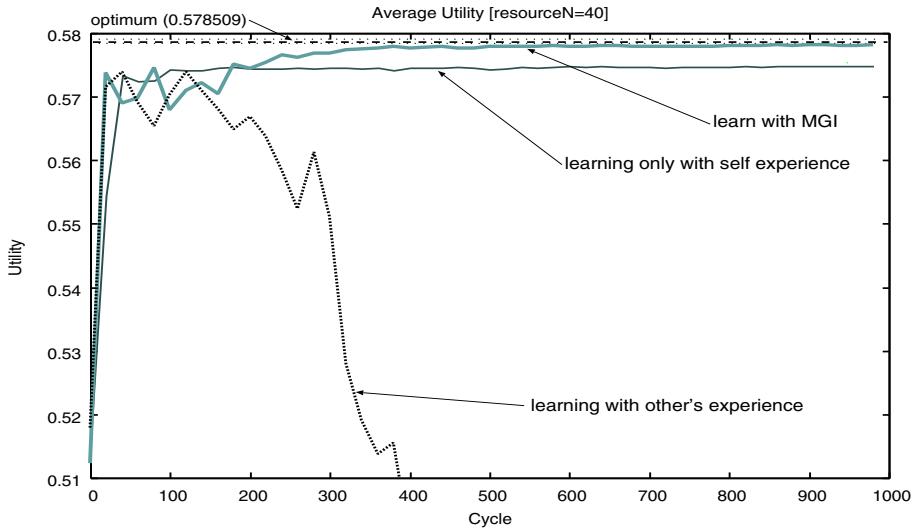


Fig. 2. Comparison of Learning with Experiences: Self, Others and Moderated

2.3 Exploration and Exploitation

The MGI retains an open issue: it does not guarantee the convergence of agents' policies into Nash equilibrium, especially in the case of dynamic environments and swiftly learning agents. Actually, there are two cases where the agents can not reach Nash equilibrium by MGI:

- Agents explore various resources so often so that each agent can not get stable and suitable information about utilities for each resources.
- Agents explore resources so rarely so that the agents can not test enough combinations to find the equilibrium.

This issue presents a kind of balancing problem between exploration and exploitation in multi-agent learning [7]. One solution to this problem is to change several learning parameters over time. For example, [8] applied simulated annealing technique to reinforcement learning to balance exploration and exploitation. In their method, a temperature parameter in a Boltzmann-softmax function drops toward zero over time. Finally, each agent obtains a deterministic policy in which there is no randomness in the agent's choice. Although such deterministic behavior is desirable from the perspective of stability, the monotonic dropping method might cause a serious oscillation problem when the environment changes after the conversion. Figure 3 shows a typical oscillation problem when we use a monotonic dropping method in learning. This figure shows the changes of the number of agents who choose one of three resources. In the beginning (left in the graph), agents choose a resource randomly, but learn to achieve a Nash equilibrium. But, the agents start to wander among resources at step 100

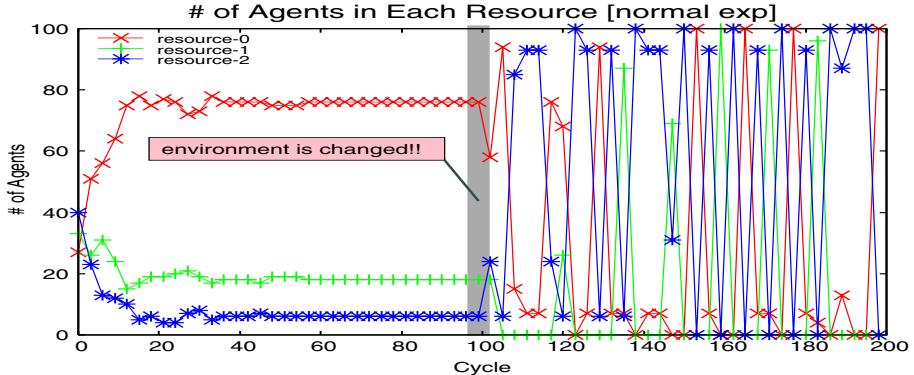


Fig. 3. Typical Oscillation Problem in Resource Sharing

when the environment (utility functions of resources) changes. Subsequently, the wandering never stops: oscillation persists.

Another idea related to this issue is the Win or Learn Fast (WoLF) principle [9]. By this principle, learning agents use different learning parameters for winning and losing phases: Agents use a set of parameters to enforce the learning faster than in the winning phase if an agent is losing. Using this principle, they show that learning agents can avoid oscillation problems and reach an equilibrium.

We generalize the concept of the WoLF principle and derive a framework to change learning parameters by which we control the learning process for convergence to an equilibrium.

3 Adaptive Temperature Control

3.1 Abstracted WoLF Principle

It is difficult to apply the WoLF principle directly to RRSP for the following reasons. The original WoLF method subsumes that each agent separately learns Q values (estimated utilities) and probabilities of actions, although we presume that the agent uses the Boltzmann softmax function to determine the probabilities.

- (1) Let $\alpha \in (0, 1]$, $\alpha' \in (0, 1]$, $\rho \in (0, 1]$, $T_i > 0$ and $\lambda \in (0, 1]$. Initialize $V_i(r_j)$ randomly for each i, j .
- (2) Repeat,
 - (a) Choose resource r_j according to Boltzmann softmax.
 - (b) Receiving utility u_j , $V_i(r_j) \leftarrow (1 - \alpha)V_i(r_j) + \alpha u_j$.
 - (c) Receiving MGI $u'_k (= U_k(n_k + L))$, where $k \neq j$, $V_i(r_k) \leftarrow (1 - \alpha')V_i(r_k) + \alpha' u'_k$.
 - (d) If $r_j = \operatorname{argmax}_{r_k} V_i(r_k)$ and $u_j > \rho V_i(r_j)$ or $r_j \neq \operatorname{argmax}_{r_k} V_i(r_k)$ and $u_j < V_i(r_j^*)$, $T_i \leftarrow \lambda T_i$

Fig. 4. Procedure of Learning Estimated Utility and Adapting Temperature

To overcome this difference from original WoLF settings, we attempt to abstract the concept of WoLF principle from the perspective of ‘exploration and exploitation’, which is a main issue of RRSP, as shown in the previous section. In WoLF, an agent learns slowly when it is receiving a greater reward than expected (winning phase), but it learns quickly when it obtains a worse reward (losing phase). We interpret and abstract this principle as follows:

An agent should exploit/explore in the winning/losing phase.

3.2 Adaptive Temperature Control Based on WoLF

As described in section 2, we suppose that each agent makes its choice according to Boltzmann softmax. Therefore, *exploration* and *exploitation* can be defined simply as

- *exploration*: to increase the temperature T .
- *exploitation*: to decrease the temperature T .

We also define *winning* as the following: Agent a_i is winning in the following cases:

1. When it chooses resource r_j whose expected utility $V_i(r_j)$ is the best in V_i , and it receives more utility than $V_i(r_j)$.
2. When it chooses non-best resource r_k , and it receives less utility than $V_i(r_j)$ (we suppose that r_j is the best resource according to V_i).

That definition is reasonable because the learning of the agent will enhance the current best policy ($V_i(r_j)$). As described in the section 1, we specifically address the stability of agents’ behaviors. Therefore, we take only the best choice of each agent into account because only the best choice is meaningful in a stabilized situation. Finally, we obtain the procedure shown in figure 4.

3.3 Experiment 1

We conducted an experiment on RRSP to show the effects of the proposed method.

As the utility function in the experiment, we use the following function:

$$U_j(n) = 1 - \epsilon_j^{\frac{C_j}{n}}, \quad (3)$$

where ϵ_j and C_j respectively denote the error rate and the capacity of the resource r_j . The meaning of this function is as follows: Consider a kind of information service in which an agent can obtain required information in the probability $(1 - \epsilon)$ for a query. The agent can repeat the query $\frac{C_j}{n}$ times in one cycle until it obtains the required answer. The agent can try many times when the capacity C_j

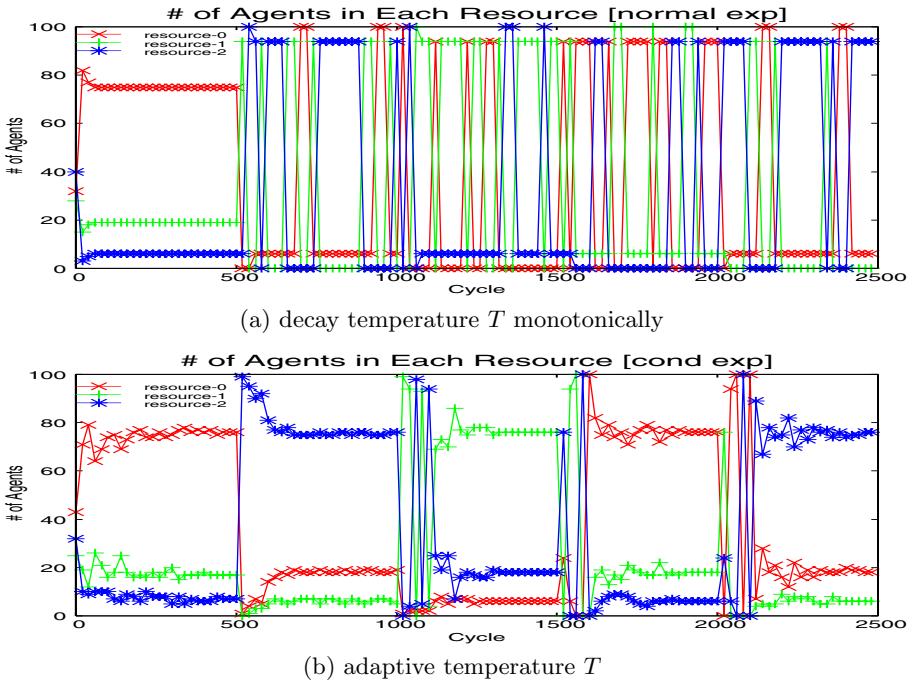


Fig. 5. Exp.1: Changes of # of Agents Who Choose Each Resource under Adaptive Temperature Control

is large. On the other hand, when numerous agents use the resource, the agent has few chances to try its query within a certain time period. Therefore, the utility function described above shows the probability that an agent can obtain the correct information within one cycle. In the following experiment, we use the following parameters: the number of agents n : 100, the number of resource m : 3, the capacities of resource $\{C_j\}$: $\{20, 5, 2\}$. Note that eq. (3) is just a sample for the experiment. We can apply the proposed method to other general utility functions.

Similarly to the preliminary experiment in section 2.3 and figure 3, we train agents in the environment, which changes the capacities of resources C_j in a certain period. For this experiment, we rotate the capacities of resources every 500 cycles. We use $L = 3$ as the moderation factor for MGI learning. Figure 5 shows the result of changes in the number of agents in each resource through learning. The graph (a) of this figure shows the case in which the agent decreases the temperature monotonically. As in preliminary experiments, although the agents can achieve the equilibrium in the first phase ($\text{cycle} < 500$), they commence oscillating (up-and-down) behaviors at the first change of the environment. Subsequently, the oscillation never ceases. On the other hand, when we use adaptive temperature control based on WoLF ((b) in figure 5), we can see that agents reach an equilibrium in each phase of 500 cycles. We confirmed that duration

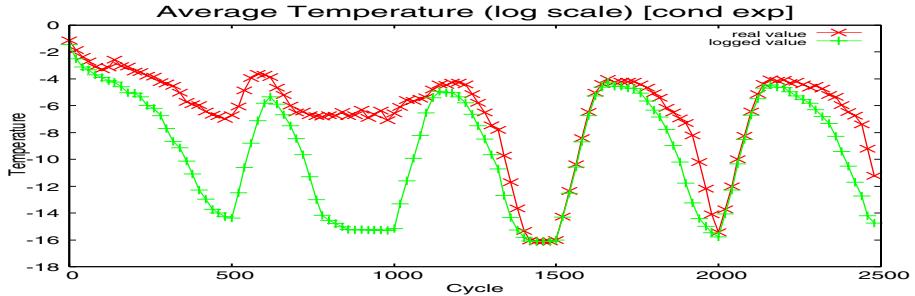


Fig. 6. Exp.1: Changes of Average Temperature under Adaptive Temperature Control

by investigating the changes of average temperatures of all agents. The figure 6 indicates changes of average of temperatures of all agents in adaptive temperature methods in log-scale. In the graph, both arithmetic and geometrical means are plotted. This graph illustrates that agents raise the temperature when the environment is changed so that they start to explore a new equilibrium. They drop their temperature gradually when the behaviors approach the equilibrium. Consequently, the temperatures become sufficiently low that most agents never change their choice until the environment is changed.

4 Adaptive Moderation Control

4.1 Features and Adaptation of Moderation Factor

As described in section 2.2, agents can reach a Nash equilibrium using MGI in learning. However, the convergence to the equilibrium is fragile when the state (agents' estimated utility V_i) is far from the equilibrium because agents tend to behave randomly in such cases. Consequently, MGI imparts a bad influence on the learning of V_i . For example, (a) in figure 8 shows the changes of the number of agents who choose a resource in the case where the moderation factor $L = 1$. In this case, the learning can not reach an equilibrium, so that there remain unstable behaviors of agents. The learning becomes more robust and reaches closer to the equilibrium when we choose the moderation factor $L = 5$ ((b) in figure 8). However, some perturbation remains in agent behavior in this result. Generally, the larger the moderation factor we use, the increasingly robust the learning is.

On the other hand, when we give a large value to the moderation factor, we can not guarantee that the learning can reach the Nash equilibrium shown in [6]. The meaning of the moderation factor L is: If an agent changes its choice from resource r_j to r_k , the other $L - 1$ agents may also follow to change their choice to the resource r_k . As a result, the number of agents who choose r_k may increase L , so that the utility the agent obtain may decrease to $U_k(n_k + L)$. In other words, large L forces agents to be pessimistic about choosing other resources instead of the current choice. This feature is useful when too many agents change

- (1) Let $\beta \in (0, 1]$ and $\mu > 0$. Initialize $\bar{n}_j \leftarrow 0.0$ and $\hat{n}_j \leftarrow 0.0$ for each j .
- (2) In each cycle of the agents' decision (step (2) in figure 4),
 - (a) Count n_j , the number of agents who use resource r_j .
 - (b) Let $\bar{n}_j \leftarrow \frac{\bar{n}_j + \beta n_j}{1+\beta}$, $\hat{n}_j \leftarrow \frac{\hat{n}_j + \beta n_j^2}{1+\beta}$, and $L \leftarrow 1 + \mu \sqrt{\hat{n}_j - \bar{n}_j^2}$.
 - (c) Calculate utilities and MGI: $u_j = U_j(n_j)$, $u'_j = U_j(n_j + L)$.
 - (d) Provide u_j to the agents who use resource r_j , and inform u'_j to other agents.

Fig. 7. Procedure of Adapting a Moderation Factor

their choices, because pessimistic information about other resources inhibit the changes of choices.

But, if agents are too pessimistic to try other choices, they can not explore enough combinations to find true equilibrium. In order to avoid such situations, we should use small L .

To solve this dilemma, we introduce a mechanism to adapt the factor L according to agents behaviors. As investigated above, L should be large when too many agents explore, while it should be small when the most of agents fix choices. In order to measure such agents' behaviors, we use standard deviation σ_j of the number of agents that choose resource r_j . Using σ_j , we determine the value of moderation factor L_j for r_j as $L_j = 1 + \mu\sigma_j$, where $\mu > 0$ is an amplification factor.

The merit using standard deviation is that we need not introduce a central control mechanism to measure agents behaviors and control the moderation. The standard deviation σ_j can be calculated only using history of n_j that is already measured at each resource to determine its utility. This feature is important for large-scale open systems because it is difficult to handle whole behaviors of all agents in such systems.

Finally, we obtain the procedure shown in figure 7 for the control of moderation factors.

4.2 Experiment 2

In the second experiment, we show the convergence performance of adaptive moderation control compared to the fixed value of moderation in MGI learning.

As described in section 4.1, the learning is fragile with the small moderation factor, although the learning can not be guaranteed to reach a Nash equilibrium with the large moderation factor. The graphs (a) and (b) in figure 8 shows the respective changes of the number of agents using each resource in the cases of fixed moderation factors $L = 1$ and $L = 5$. On the other hand, (c) shows the results of adaptive moderation control proposed in section 4.1. Here, it is apparent that the agents' behaviors converge completely into the equilibrium. These figures illustrate that agents can not reach an equilibrium, but rather continue exploration in the case $L = 1$ (a), whereas most of the agents reach equilibrium and stop exploration in the case of $L = 5$ (b) and the adaptive moderation (c). The case of adaptive moderation reaches the stable state completely. Therefore,

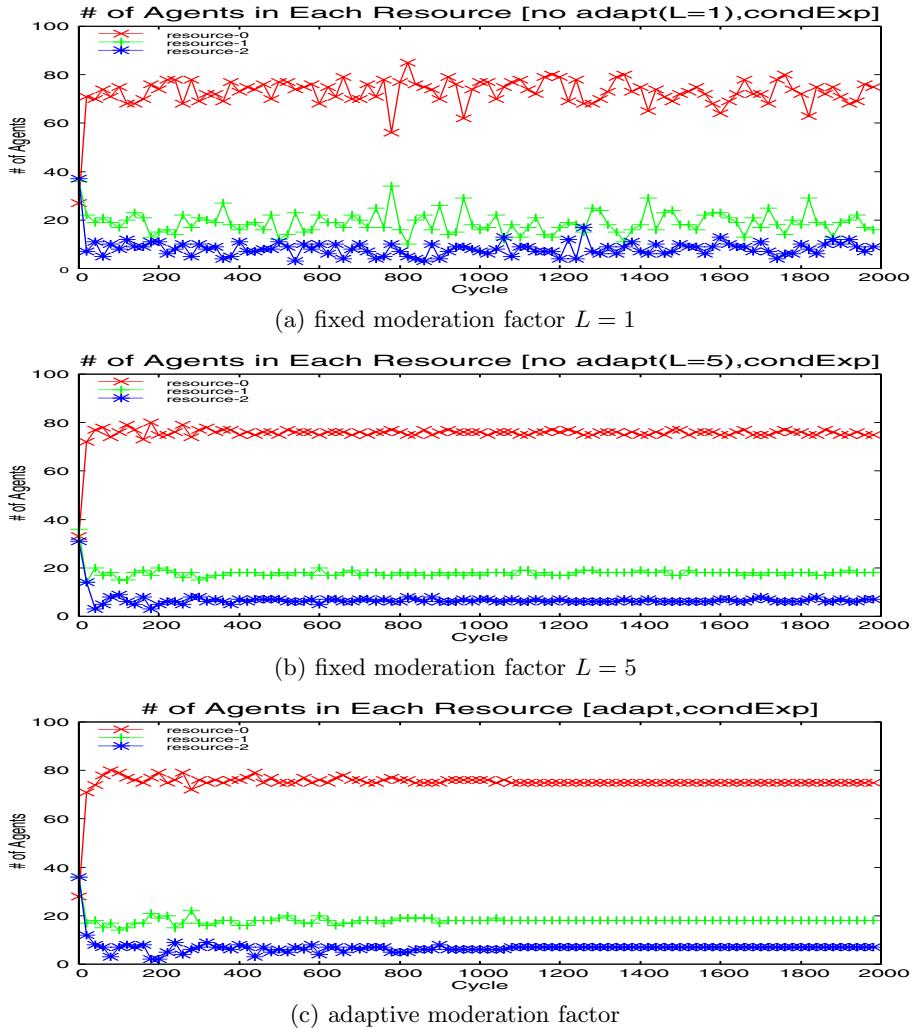


Fig. 8. Exp.2: Changes of Number of Agents Who Choose Each Resource under Adaptive Moderation Control

it is guaranteed that the agents' choice is one of Nash equilibrium. Therefore, the state is a kind of optimum so that the average of utilities that agents obtain is maximized locally. Actually, the final value of the total utilities in the adaptive moderation control is 46.292, which is greater than those of the other cases (46.239 in $L = 1$ and 46.258 in $L = 5$). Figure 9 shows the changes of average utilities that each agent obtains through learning. As similarly shown in figure 8, each agent can obtain stable and high utility in the case of the adaptive moderation control (c), but the utility sometimes drops in the case of fixed moderation factors because of perturbation.

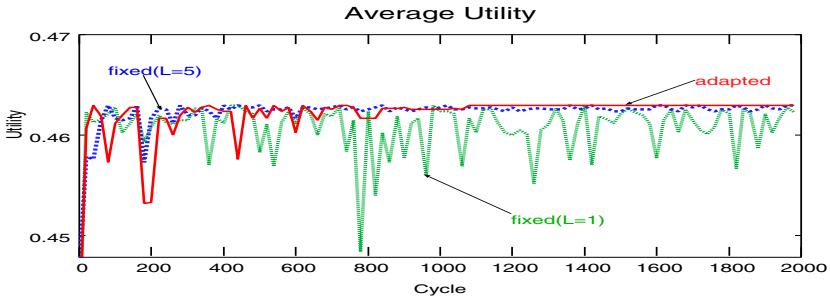


Fig. 9. Exp. 2: Changes of Average Utility under Adaptive Temperature Control

5 Related Works and Discussion

Several researchers attack MARL for (repeated) resource sharing problems [10,11,12].

Abdallah and Lesser [10] attacked task allocation games and introduced weighted policy learning algorithm that can robustly converges for various game setups. Main difference to these works is a usage of global information to speed-up learning. As described in section 2.2, learning only by direct experiences of each agent is too slow to explore whole situations in MARL when the numbers of resources and agents are large. Our method can speed-up the learning and also provide robustness of convergence. Such feature is useful when the environments of agents are dynamic and open.

MG1 itself is tightly related with regret-based approaches like [12]. Introducing $U_k(n_k + L)$ in eq. (2) is similar idea to calculate the regret values. Instead of calculating regret (difference of utilities of chosen and not-chosen resources), MG1 utilizes the utilities to estimate Q values. Therefore, our methods to adjust parameters T and L will be able to be applied to these regret-based approaches to enhance convergence and speed-up of learning.

Works of Bowling and Veloso [9] inspired many ideas to this work. As described in section 3, the adaptive mechanism for temperature is directly derived from the concept of WoLF in their works.

6 Concluding Remark

The study described in this article investigated RRSP from the viewpoints of balancing exploration and exploitation of MARL, and proposed methods to control two learning parameters, temperature in Boltzmann softmax function, and the moderation factor in moderated global information. The experimental result shows the advantage of the method in robustness against changes of environment and steadiness of learning.

Although we show the experimental results of a small variation of setting, the proposed methods are applicable to several environments. For example, we can use different utility functions that satisfy the monotonically decreasing condition.

It is also possible to introduce internal states in the agent to adapt Markov decision processes.

Several open issues remain in relation to the proposed methods:

- theoretical analysis of convergence and comparison to other conventional methods.
- how to set remaining learning parameters suitably, especially the amplification factor μ in the adaptive moderation control.
- the relation between the convergence speed and the robustness of learning and control.
- how to formalize hierarchical social structures as a resource-sharing problem under which groups of agents make choices.

References

1. Tan, M.: Multi-agent reinforcement learning: Independent vs. cooperative agents. In: Proc. of the Tenth International Conference on Machine Learning, pp. 330–337 (1993)
2. Garland, A., Alterman, R.: Learning procedural knowledge to better coordinate. In: Proc. of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI, pp. 1073–1079 (2001)
3. Yamashita, T., Izumi, K., Kurumatani, K., Nakashima, H.: Smooth traffic flow with a cooperative car navigation system. In: Proc. of the Fourth International Joint Conference on Autonomous Agents and Multi Agent Systems, AAMAS, pp. 478–485 (2005)
4. Wolpert, D.H., Tumer, K., Frank, J.: Using collective intelligence to route internet traffic. Advances in Neural Information Processing Systems 11, 952–958 (1999)
5. Kluegl, F., Bazzan, A.L.C., Wahle, J.: Selection of information types based on personal utility - a testbed for traffic information markets. In: Proc. of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems, AAMAS, pp. 377–384 (2003)
6. Ohta, M., Noda, I.: Reduction of adverse effect of global-information on selfish agents. In: Antunes, L., Takadama, K. (eds.) Seventh International Workshop on Multi-Agent-Based Simulation (MABS), Hakodate, AAMAS 2006, Shinko, pp. 7–16 (May 2006)
7. Carmel, D., Markovitch, S.: Exploration strategies for model-based learning in multiagent systems. Autonomous Agents and Multi-agent Systems 2, 141–172 (1999)
8. Guo, M., Liu, Y., Malec, J.: A new q-learning algorithm based on the metropolis criterion. IEEE Transactions on Systems, Man and Cybernetics, Part B 34(5), 2140–2143 (2004)
9. Bowling, M., Veloso, M.: Multiagent learning using a variable learning rate. Artificial Intelligence 136, 215–250 (2002)
10. Abdallah, S., Lesser, V.: Learning the task allocation game. In: Proc. of AAMAS 2006, IFAAMAS, pp. 850–857 (May 2006)
11. Abdallah, S., Lesser, V.: Multiagent reinforcement learning and self-organization in a network of agents. In: Proc. of AAMAS 2007, IFAAMAS, pp. 172–179 (May 2007)
12. Marden, J.R., Arslan, G., Shamma, J.S.: Regret based dynamics: Convergence in weakly acyclic games. In: Proc. of AAMAS 2007, IFAAMAS, pp. 194–201 (May 2007)

Ontology-Based Natural Query Retrieval Using Conceptual Graphs*

Tho Thanh Quan¹ and Siu Cheung Hui²

¹ Faculty of Computer Science and Engineering, Hochiminh City University of Technology
Hochiminh City, Vietnam
qttho@cse.hcmut.edu.vn

² School of Computer Engineering, Nanyang Technological University
Singapore
asschui@ntu.edu.sg

Abstract. As compared to the classical library model of printed materials, digital library offers a more efficient way to browse and search for scholarly information in a networked environment. Currently, the most common way of searching and retrieving information from digital libraries is still by means of keyword-based queries. Over the past many years, there have been many attempts to enhance the query formalism to allow users to retrieve information in a more effective manner. Among them, natural query that is expressed using natural language is obviously the most natural form of search requests. However, typical NLP techniques for natural query retrieval suffer from high cost of complexity. In addition, they are also not very effective when dealing with grammatically imprecise search requests. In this paper, we propose a novel ontology-based approach for natural query retrieval of scholarly information using conceptual graphs. The paper will present the proposed ontology-based approach and its experimental results. The proposed approach has achieved some promising initial results.

1 Introduction

Digital library is an organized repository of recorded knowledge which can be accessed in a networked environment [1]. A digital library can be in various forms such as an open access e-print archive, cross-archive search service, digital collection of intellectual works or Web Portal with search functions provided. Generally, the major advantage of digital library is its capabilities of organizing knowledge and retrieving information in a distributed environment such as the World Wide Web (WWW).

One of the major functions facilitated by digital library, as compared to the classical library model, is its search utility that allows users to find their intended documents and other scholarly information easily and effectively. Currently, the most common search mechanism offered by digital libraries is based on user keywords.

* This work is completed with the financial support of the research project T2008-KHMT-14, funded by Hochiminh City University of Technology, Vietnam National University Hochiminh City.

Although keyword-based search techniques have been proven to be useful, the quality of search results is still far from satisfactory. It is because keywords alone are not powerful enough to represent the semantics conveyed in both documents and queries.

Over the past many years, there have been many attempts to improve the accuracy of information retrieved from digital libraries by means of extracting or encoding semantics into documents and queries. In [2], a system for generating query-specific document summarization was proposed. On the other hand, much research have also been conducted on enhancing the submitted queries to support more informative forms of search queries. For instance, in [3], fuzzy query was proposed to handle uncertainty information. In another approach [4], queries were transformed into a hybrid form to capture the associated metadata. However, among the various formalisms proposed to represent queries with precise semantics, natural language is still the most desirable one to many users. In BT Digital Library [5], a question-answering mechanism was proposed based on the knowledge inferred from an ontology. However, lacking a well-defined semantic representation for the submitted query, this system can only reason according to a fixed set of pre-defined natural query forms.

To process natural queries, conceptual graph (CG) [6, 7] is regarded as an effective technique to capture and represent the semantics in linguistic structures. Queries represented by CG can further be processed by appropriate tools such as SESAME [8] to obtain the precise answers retrieved from a knowledge base. In [9], an effort on supporting retrieval using natural query over CG-represented documents was reported. However, automatic translation of natural queries into the corresponding CGs is still a complex and challenging problem. More recent work on this issue rely mostly on natural language processing (NLP) techniques, i.e. making use of grammars and corpus to construct CGs from textual data [10, 11]. However, these techniques would suffer from poor performance when dealing with incomplete or imprecise queries in terms of grammatical structures, which are likely to occur in many practical situations.

In this paper, we propose a new approach on natural query retrieval based on CG using domain ontology. Our work is motivated from previous research [12], which aimed at generating CGs from Vietnamese documents using a simple grammar combined with some heuristics rules. We do not intend to deal with all possible forms of natural queries. Instead, we only attempt to support natural query retrieval according to the domain interest of users when searching for information. As such, it makes our approach more practical and realistic. In this paper, we apply our proposed approach to the Scholarly Information Ontology which encompasses the domain knowledge stored in a digital library. The natural query will be processed according to the domain ontology and converted into the corresponding CG for scholarly information retrieval.

The rest of this paper is organized as follows. Section 2 discusses the Scholarly Information Ontology for our digital library. Section 3 discusses the basic concept of CG and its representation for natural query. Section 4 presents our proposed approach for generating conceptual graph from natural query. Section 5 gives some initial experimental results. Finally, Section 6 concludes the paper and discusses the direction for our future research.

2 Scholarly Information Ontology

Currently, most of the digital libraries are accessible from the WWW. With the recent advancement of Semantic Web [13], which offers strong capabilities of knowledge sharing and exchanging, there are much research on developing digital library systems over the Semantic Web environment [14]. In the Semantic Web, ontology [15] is adopted as the standard for knowledge representation due to its strong capability of reflecting real-life knowledge in a machine-understandable manner. Thus, Semantic Web-based digital libraries can improve the performance of data searching, browsing and personalization. Ontology is basically a conceptualization of a domain into a human understandable, machine-readable format consisting of entities, attributes, relationships and axioms. Ontology uses classes, which contain attributes, to represent concepts. Ontology also supports taxonomy and non-taxonomy relations between classes.

Formally, an ontology can be defined as follows. An ontology O consists of four elements (C, A^C, R, X) . C represents a set of concepts. A^C represents a collection of attributes sets, one for each concept. $R = (R_T, R_N)$ represents a set of relationships, which consists of two elements: R_T is a set of taxonomy relationships and R_N is a set of non-taxonomy relationships. Each concept c_i in C represents a set of objects, or instances, of the same kind. Each object o_{ij} of a concept c_i can be described by a set of attribute values denoted by $A^C(c_i)$. Each relationship $r_i(c_p, c_q)$ in R represents a binary association between concepts c_p and c_q , and the instances of such a relationship are pairs of (c_p, c_q) concept objects. X is a set of axioms. Each axiom in X is a constraint on the concept's and relationship's attribute values or a constraint on the relationships between concept objects.

For example, the Scholarly Information Ontology $O_S = (C, A^C, R, X)$ is an ontology with each component defined as follows:

$$\begin{aligned}
 C &= \{\text{"Document", "Research Area"}\} \\
 A^C(\text{"Document"}) &= \{\text{"Name", "Author", "Title", "Keywords", "Abstract", "Body", "Publisher", "Publication Date"}\} \\
 A^C(\text{"Research Area"}) &= \{\text{"Name", "Keyword"}\} \\
 R_T &= \{\text{superarea-of("Research Area", "Research Area"), subarea-of("Research Area", "Research Area")}\} \\
 R_N &= \{\text{belong-to("Document", "Research Area"), consist-of("Research Area", "Document")}\} \\
 X &= \{ \\
 &\quad \text{Implies(Antecedent(consist-of(I-variable(x1) I-variable(x2))),} \\
 &\quad \quad \text{Consequent(belong-to(I-variable(x2) I-variable(x1)))))} \\
 &\quad \text{Implies(Antecedent(belong-to(I-variable(x1) I-variable(x2))),} \\
 &\quad \quad \text{Consequent(consist-of(I-variable(x2) I-variable(x1)))))} \\
 &\quad \text{Implies(Antecedent(superarea(I-variable(x1) I-variable(x2))),} \\
 &\quad \quad \text{Consequent(subarea(I-variable(x2) I-variable(x1)))))} \\
 &\quad \text{Implies(Antecedent(subarea(I-variable(x1) I-variable(x2))),} \\
 &\quad \quad \text{Consequent(superarea(I-variable(x2) I-variable(x1)))))} \\
 &\}
 \end{aligned}$$

The Scholarly Information Ontology has two concepts (or classes): *Document* and *Research Area*. The attributes of the class *Document* are major properties of a

scientific publication such as *author*, *title*, etc. A *Research Area* is indicated by a set of appropriate keywords. Taxonomy relation set R_T defines hierarchical relations between research areas, in which a research area may be a super-area or sub-area of others. Non-taxonomy relation set R_N consists of relations between *Document* and *Research Area* implying that a document can belong to some research areas and a research area may consist of different scientific documents. The axiom set X contains some basic rules that imply the inverted relations of defined relations. For example, if a document belongs to a research area, then the research area contains that document and vice-versa.

As can be observed from the above example, ontology is of high precision when representing knowledge in a certain domain. In this research, we employ the Scholarly Information Ontology to represent the knowledge of a digital library. Scholarly Information Ontology can be generated automatically from scientific documents by capturing and reflecting conceptual entities and relations among the documents. The details on the automatic generation of the Scholarly Information Ontology can be found in [16]. Figure 1 gives the conceptual schema of the Scholarly Information Ontology.

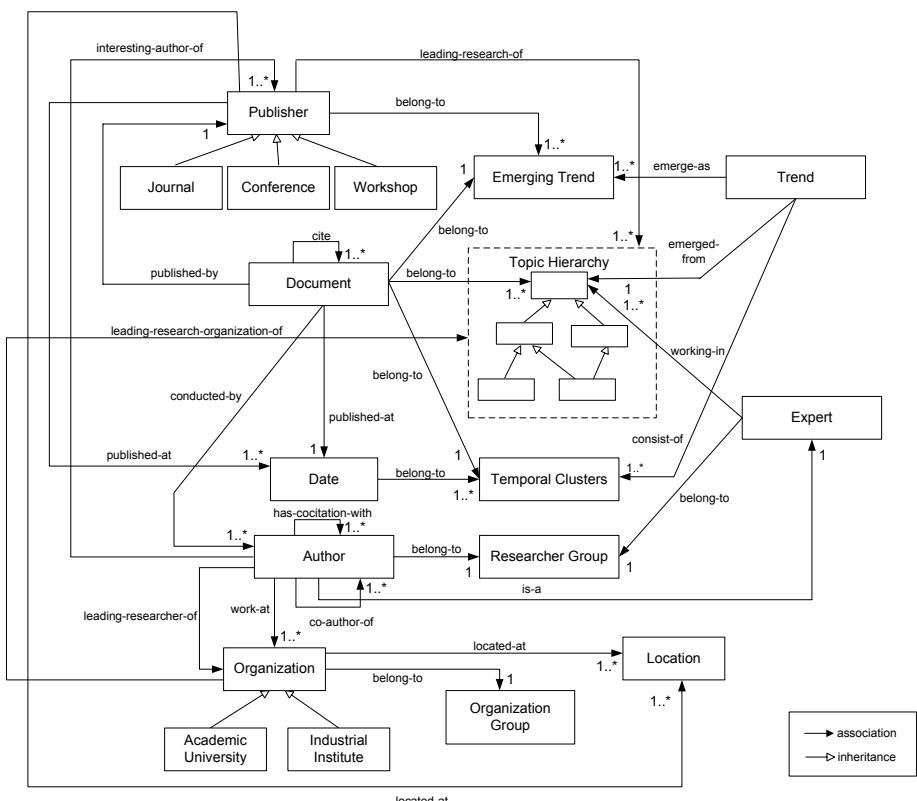


Fig. 1. Conceptual Schema of the Scholarly Information Ontology

3 Query Representation Using CG

A conceptual graph (CG) is a notation for logic based on existential graphs and semantic networks in artificial intelligence. A CG has a displayed form as a directed graph whose nodes can be a *concept node* or *relation node*. A concept node implies an individual of a concept, while a relation node indicates the relationships between individuals. Figure 2 shows an example of a CG, whose concept nodes are presented in boxes and relation nodes in ovals. The fact conveyed by this CG is that the author *Tim Berners Lee* has written the paper entitled “*The Semantic Web*”. In this CG, the individuals *Tim Berners Lee* and “*The Semantic Web*” are called *individual referents* of the concepts *Author* and *Document* respectively.



Fig. 2. An example conceptual graph

Hence, CG has rendered a powerful formalism for describing the world of logic. CG is also an effective formalism to represent queries in natural language. Figure 3 shows a CG representing the query “Find the author who published documents about Semantic Web.” Notice that the “?” symbol indicates a *query referent* of the CG, which specifies the object that will be searched. The “*” symbol indicates a *generic referent*, which means that no specific individual of the concept *Document* is mentioned explicitly in the query.

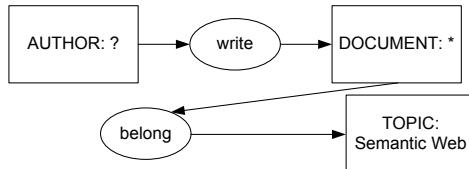


Fig. 3. A natural query represented by conceptual graph

4 Automatic Generation of CG-Based Query

In this section, we present an approach for automatic generation of CG-based query from natural query using the knowledge captured from domain ontology (e.g. Scholarly Information Ontology). The automatic generation process consists of the following two steps:

- *Concept Generation* – It identifies ontological concepts and individuals in the query and converts the identified concepts and individuals into the concept nodes of the CG.

- *Relation Construction* – It constructs relation nodes between the concept nodes according to the domain ontology to obtain the final CG. During construction, some additional concept and relation nodes may need to be generated and added into the final CG.

4.1 Concept Generation

This step aims to parse the submitted query in order to identify the ontological concepts and individuals. As domain ontology contains well-defined concepts and individuals, it provides a powerful means for recognizing these concepts and individuals automatically from a query. For example, let's consider the following query:

(Q1): “Who is the author of the document entitled “The Semantic Web”? ”

Using the Scholarly Information Ontology given in Figure 1, we can recognize the ontological concepts *Author* and *Document* and the ontological individual “*The Semantic Web*” of the ontological class *Document* from the query.

Note that when constructing domain ontology, the ontology engineer can define an *ontology vocabulary* that helps to recognize different keywords that may be relevant to certain ontological concepts and individuals. For example, the ontology engineer may define that the question term *who* should refer to an *author*, whereas the concept *Document* may be referred to by some linguistic terms such as *document*, *paper*, *article*, *publication*, etc. Thus, if the natural query is changed to another form like “*Who wrote papers about the Semantic Web?*”, the same ontological concepts and individuals will be recognized.

After recognizing ontological concepts and individuals in a query, we will map them into the corresponding concept nodes in the final CG. To do this, the following heuristic rules are used:

- An ontological individual will be mapped as an individual referent.
- An ontological concept will be mapped as a query referent if there is no individual of this concept recognized in the query.

Therefore, for the query (Q1), the individual of “*The Semantic Web*” will be mapped as an individual referent. Between the two ontological concepts *Author* and *Document*, we only preserve the concept *Author* as a query referent since the concept *Document* has the corresponding individual (i.e. “*The Semantic Web*”) found in the query.

Let's consider another query:

(Q2): “I'm interested in the scientific documents that are written by the famous author Tim Berners Lee and published in the year of 2001. ”

With the Scholarly Information Ontology given in Figure 1, the ontological concepts identified are *Document*, *Author* and *Date*, whereas the ontological individuals are *Tim Berners Lee* and *2001*. Since *Tim Berners Lee* and *2001* are individuals of the concepts *Author* and *Date* respectively, these concepts will not be further considered as query referents. Thus, only one query referent of *Document* and two individual referents of *Tim Berners Lee* and *2001* are identified in this query.

4.2 Relation Construction

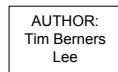
As discussed in Section 3, the identified query referents and individual referents will correspond to concept nodes in the final CG. In this step, we construct the relations between these concept nodes. First, for each pair of query referent and individual referent identified, we find a path between the corresponding ontological concepts in the conceptual schema of the domain ontology. Then, we unify all the paths to form the final CG-based query.

For example, in query (Q1), there is one query referent of *Author* and one individual referent of *Document* identified. The corresponding path between these concepts in the Scholarly Information Ontology is given in Figure 4, which is also the final CG-based query generated for the query.



Fig. 4. CG-based query generated from query (Q1)

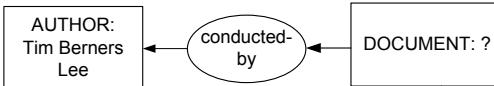
In another query (Q2), there is one query referent of *Document* and two individual referents of *Author* and *Date*. Figure 5(a) gives the identified path between *Document* and *Author*, while Figure 5(b) shows the path between *Document* and *Date*. Finally, Figure 5(c) presents the unification of these paths, producing the final CG-based query.



(a) Path between *Document* and *Author*



(b) Path between *Document* and *Date*



(c) Final CG-based query

Fig. 5. CG-based path generated from query (Q2)

Consider the following query:

(Q3): “I want to know the emerging trend of the year 2007.”

There are one query referent *Emerging Trend* and one individual referent *Date* identified. Figure 6 shows the path between these concepts in the Scholarly Information Ontology, and the corresponding CG-based query. Note that in the identified path,

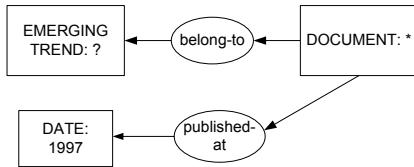


Fig. 6. The CG-based path generated from query (Q3)

an additional concept *Document*, which is mapped as a generic referent in the CG, is also created.

4.3 Uncertainty Resolution

When parsing natural query to identify ontological concepts and individuals as discussed in Section 4.1, it is sometimes difficult to be certain about the concepts and individuals associated with a keyword, even when the ontology vocabulary is used. Here, we discuss some strategies that can be used to handle such uncertainties.

Multiple individuals of the same ontological concept. When submitting a query, the user may want to find information relevant to multiple individuals of the same ontological class instead of a single one. In such case, we can just simply generate a single node corresponding to these individuals. For example, in the following query:

(Q4): “Find documents published by the ACM Press and IEEE Press.”

there are two individuals *ACM Press* and *IEEE Press* of the same concept *Publisher* identified. Figure 7 shows the final CG-based query.



Fig. 7. The CG-based query for query (Q4).

Multiple concepts of the same keyword/phrase. A certain keyword or phrase in the natural query may be identified as indicators for different ontological concepts. As a result, there are multiple CG-based queries generated accordingly. Our solution to this problem is to consider all the generated CG-based queries, since all of them should be able to provide answers for the intended questions from users. For example, in the following query:

(Q5): “Find researchers working in the field of the Semantic Web.”

the phrase “*Semantic Web*” may be identified relevant to two concepts *Document* and *Publisher* (In fact, we have a journal named *Semantic Web* defined in Scholarly Information Ontology). Thus, there are two CG-based queries generated as shown in Figure 8. Obviously, the results obtained when processing these queries should be relevant to the request from users.

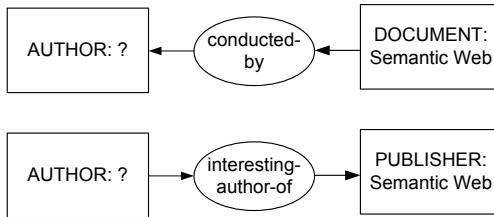


Fig. 8. The CG-based queries for query (Q5)

Multiple paths of the same pair of query/individual referents. When finding a path between concepts in the domain ontology, which are corresponding to a query/individual referents recognized in the original natural query, we sometimes end up with more than one possible path. In such case, we will select the most reasonable path whose relation nodes are most similar to the query keywords. The similarities between the relation nodes and the keywords once again rely on the vocabulary defined by the ontology engineering when constructing ontological relations in the domain ontology. For example, in the following query:

(Q6): “Find researchers working in USA.”

there is one query referent *Author*, and one individual referent *USA* of concept *Location*. In the Scholarly Information Ontology, there are two possible paths between the concepts *Author* and *Location*, as depicted in Figure 9(a) and Figure 9(b). However, the term *working* in the query should be relevant to the ontology vocabulary defined for the ontological relation *work-at* in the Scholarly Information Ontology. Thus, the first path, i.e. the path depicted in Figure 9(a), should prevail and be selected as the appropriate CG-based query for the given query.

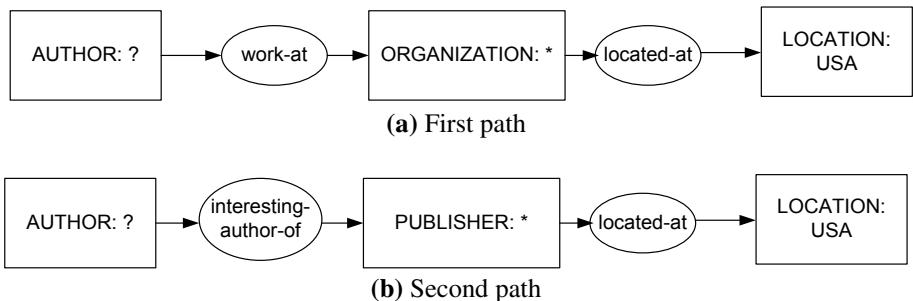


Fig. 9. Two paths identified for query (Q6)

5 Experimental Results

In this section, we present an initial experiment to evaluate the effectiveness of the proposed ontology-based approach for natural query retrieval. In the experiment, we use the Scholarly Information Ontology which was generated from a citation database

as domain ontology. The citation database was constructed from a collection of 1400 scientific documents. These documents are downloaded from the Institute for Scientific Information's (ISI) website (<http://www.isinet.com>) based on the research area "Information Retrieval" published from 1987-1997. The downloaded documents are pre-processed to extract related information such as the title, authors, citation keywords, and other citation information. The Scholarly Information Ontology was given in Figure 1.

We then conducted a survey on common natural queries that users may wish to submit when looking for scholarly information from a digital library. Students from the Faculty of Computer Science and Engineering, Hochiminh City University of Technology (<http://www.cse.hcmut.edu.vn>) had participated the survey. As a result, 346 queries are collected. Table 1 lists the 30 most common natural queries. We then process the natural queries using our proposed approach to generate the corresponding CG-based queries in order for evaluating its effectiveness for information retrieval. From the 346 queries collected, our approach is able to produce correct answers for 323 cases. Therefore, the proposed approach has achieved an accuracy of 93.35%. In addition, we have also compared the performance of our approach with other typical information retrieval (IR) techniques in terms of typical IR measures like recall, precision and F-measure, using the same query set as depicted in Table 1. Two

Table 1. Examples of common natural queries

-
1. Find paper about "Information-retrieval system"
 2. Find paper about "Information-retrieval system" published in 1993?
 3. Document about "micro computers"?
 4. Search for document about "micro computers" written by J. Beaumont
 5. Document about "micro computers" or "computer graphic"
 6. Research papers related to "security"
 7. Search article related to "security"
 8. I want to find researchers in the field of "Semantic Network"?
 9. I want to find expert in "Semantic Network" whose works are published in "Information Processing & Management"
 10. Identify author of paper published in "Information Processing & Management"
 11. Who are researching in "Computer Network"?
 12. Search paper about "Computer Network" and written in Chinese
 13. Who have documents published in German about Digital Libraries?
 14. Journal of Semantic Web in German?
 15. Find researcher working in China
 16. Find author research about "information-retrieval" currently living in China?
 17. Who did write about "information-retrieval"?
 18. Who did write about "information-retrieval system"?
 19. Who did write about "Bibliographic retrieval" and living in UK?
 20. Find paper published from the LIBRI institute
 21. Semantic Web?
 22. Work on Information system published in 1993
 23. Find documents have keyword "information retrieval"?
 24. Paper written by Morgan and Young
 25. Who have documents published in "Information Processing & Management" about "advanced algorithms"?
 26. Find some good paper about "Neural-Network"?
 27. I'm only keen on "Neural -Network" paper published in "British Journal of Psychiatry"
 28. Who did conduct some works about "Retrieval" and now working in China
 30. Give me some information about what is going on in Semantic Web today
-

techniques are used for comparison. The first technique applies the typical *tfidf* vector-space-model (VSM) to retrieve information from the input queries. In the second technique, we first cluster the data before applying the VSM for retrieval. Since the scholarly data are multi-dimensional and the queries are multi-objective, the multi-clustering technique [17] is adopted here. Table 2 presents the performance comparison of the different techniques based on recall, precision and F-measure.

Table 2. Performance comparison on retrieval

Techniques	Recall	Precision	F-measure
Vector-space-model (VSM)	78%	87%	82%
VSM + Multi-clustering	92%	94%	93%
CG-based Queries	98%	93%	95%

As can be seen in Table 2, when combined with the multi-clustering technique, the performance of the VSM-based retrieval technique has been improved significantly in terms of both recall and precision. It is because when data are clustered using the multi-clustering technique, information can be represented better in clusters, thereby enhancing the retrieval performance. As compared to the VSM-based multi-clustering technique, the precision obtained by the CG-based query processing method is slightly lower. However, the recall is better since if the CG-based queries are generated precisely, the retrieval performance can achieve with almost absolute accuracy. As a result, the CG-based query processing technique has achieved the best performance in terms of the F-measure.

6 Conclusions

This paper has proposed an ontology-based approach for natural query retrieval using conceptual graphs. We have applied the proposed approach for the retrieval of scholarly information in digital libraries, thereby enabling the sharing and exchanging of knowledge over the Semantic Web environment. The initial experimental results have shown that our proposed approach is capable of handling effectively most of the typical search requests in natural language. By avoiding using a fixed grammar to process natural query, our approach seems flexible since it can handle queries in different forms, ranging from a short phrase to a complete complex sentence. In addition, minor grammatical errors, which may probably occur in queries submitted casually by users in many practical situations, can also be tolerated reasonably.

However, the lack of grammar handling capabilities in our approach makes it difficult in processing queries which contain highly precise semantics such as “I want to find papers that are *not* related to Semantic Web” and “Recent papers on Semantic Web published *after* 2004”, i.e. queries whose concept individuals are associated with some *language operators*. In these two cases, the former does rarely occur in real practical scenarios of searching information from digital libraries, whereas the latter can be handled by applying some advanced language processing techniques. This is also the direction for our future research.

References

1. Saracevic, T., Dalbello, M.: A Survey of Digital Library Education. *Proceedings of the American Society for Information Science and Technology* 38, 209–223 (2001)
2. Varadarajan, R., Hristidis, V.: A system for query-specific document summarization. In: *Proceedings of the 15th ACM international conference on Information and knowledge management*, USA, pp. 622–631. ACM Publisher, New York (2006)
3. IntraText Digital Library, <http://www.intratext.com/CERCA/Aiuto.htm>
4. Kim, S.S., Myaeng, S.H., Yoo, J.M.: A Hybrid Information Retrieval Model Using Metadata and Text. In: *Digital Libraries: Implementing Strategies and Sharing Experiences*
5. Cimiano, P., Haase, P., Sure, Y., Völker, J., Wang, Y.: Question answering on top of the BT digital library. In: *Proceedings of the 15th International Conference on World Wide Web*, Scotland, pp. 861–862. ACM Publisher, New York (2006)
6. Sowa, J.F.: Conceptual structures: Information Processing in Mind and Machine. Addison-Wesley, Reading (1984)
7. Sowa, J.F.: Matching logical structure to linguistic structure. In: Houser, N., Roberts, D.D., Van Evra, J. (eds.) *Studies in the Logic of Charles Sanders Peirce*, pp. 418–444. Indiana University Press (1997)
8. Kampman, A., Harmelen, F., Broekstra, J.: SESAME: a generic architecture for storing and querying RDF and RDF schema. In: Horrocks, I., Hendler, J. (eds.) *ISWC 2002*. LNCS, vol. 2342, pp. 54–68. Springer, Heidelberg (2002)
9. Shady, S., Karray, F., Kamel, M.: Enhancing text retrieval performance using conceptual ontological graph. In: *Proceedings of the 6th IEEE International Conference on Data Mining*, pp. 39–44 (2006)
10. Hensman, S., Dunnion, J.: Using linguistic resources to construct conceptual graph representation of texts. In: Sojka, P., Kopeček, I., Pala, K. (eds.) *TSD 2004*. LNCS (LNAI), vol. 3206, pp. 81–88. Springer, Heidelberg (2004)
11. Zhang, L., Yu, Y.: Learning to generate CGs for domain specific sentences. In: Delugach, H.S., Stumme, G. (eds.) *ICCS 2001*. LNCS (LNAI), vol. 2120, pp. 44–57. Springer, Heidelberg (2001)
12. Hong, D.T., Cao, T.H.: Automatic Translation of Vietnamese Queries to Conceptual Graphs. *Vietnamese Journal of Computer Science and Cybernetics* 23, 272–283 (2007)
13. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web, *Scientific American* (2001), <http://www.sciam.com/2001/0501issue/0501berners-lee.html>
14. Lim, E.P., Sun, A.: Web Mining - The Ontology Approach. In: *Proceedings of The International Advanced Digital Library Conference (IADLC 2005)*, Nagoya, Japan (August 2005)
15. Guarino, N., Giaretta, P.: Ontologies and Knowledge Bases - Towards a Terminological Clarification. Toward Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing (1995)
16. Quan, T.T., Hui, S.C., Fong, A.C.M., Cao, T.H.: Automatic fuzzy ontology generation for Semantic Web. *IEEE Transactions on Knowledge and Data Engineering* 18(6), 842–856 (2006)
17. Quan, T.T., Hui, S.C., Fong, A.C.M.: Mining multiple clustering data for knowledge discovery. In: Grieser, G., Tanaka, Y., Yamamoto, A. (eds.) *DS 2003*. LNCS (LNAI), vol. 2843, pp. 452–459. Springer, Heidelberg (2003)

Optimal Multi-issue Negotiation in Open and Dynamic Environments

Fenghui Ren and Minjie Zhang

School of Computer Science and Software Engineering
University of Wollongong, Australia
`{fr510,minjie}@uow.edu.au`

Abstract. Multi-issue negotiations can lead negotiants to the “win-win” (optimal) outcomes which is not applicable in single issue negotiations. Negotiants’ preferences on all negotiated issues in multi-issue negotiations impact the negotiation results a lot. Most of existing multi-issue negotiation strategies are based on the situation that all negotiants have fixed preferences, and very little work has been done on situations when negotiants modify their preferences during the negotiation. However, as the negotiation environment becomes open and dynamic, negotiants may modify their preferences dynamic for higher profits. The motivation of this paper is to propose a novel optimal multi-issue negotiations approach to handle the situation when negotiants’ preferences are changed. In this model, agents’ preferences are predicted dynamic based on the historical records of the current negotiation, then optimal offers are generated by employing the predicted preferences so as to lead the negotiation to “win-win” outcomes (if applicable). The experimental results indicates the proposed approach can improve negotiants’ profits and efficiency considerably.

1 Introduction

Multi-issue negotiation is an active research direction in the area of multi-agent systems. Literature [1] [2] [3] indicates great achievements in this area. In [4], Fatima et. al. pointed out that the procedure of multi-issue negotiation plays a critical role to the negotiation results. In general, there are three main procedures in multi-issue negotiation [5], which are the *package deal* procedure, *simultaneous* procedure and the *sequential* procedure. In *package deal* procedure, all issues are bundled and discussed together; in *simultaneous* procedure all issues are discussed simultaneously but independently of each other; and in *sequential* procedure all issues are discussed one after another. By considering the time complexity and optimality, generally the *package deal* procedure is highly encouraged since it can outperform other two procedures in most situations.

The most significant feature of multi-issue negotiations by use of the *package deal* procedure is that it may lead the negotiation results to the “win-win” (optimal) outcomes which otherwise cannot be achieved by others [1] [6]. This feature in reality makes multi-issue negotiation important and valuable. Many researchers had paid attention on optimal outcomes in multi-issue negotiation

and some approaches had been produced and developed [2] [4] [5]. However we noticed that the existing approaches to the optimal outcomes were only based on the static negotiation environments where negotiants' preferences on negotiated issues were been fixed. Therefore, a novel optimal multi-issue negotiation approach is introduced in this paper in order to handle negotiants' preferences dynamic and lead the negotiation results to the "win-win" outcomes in open negotiation environments.

In the proposed multi-issue negotiation approach, an regression approach is proposed firstly in order to help agents to predict opponents' preferences dynamic during the negotiation. The major difference between our approach and others' works [7] [8] [9] [10] is that the prediction, here, does not require additional training process and only uses the historical offers of the current negotiation to estimate opponents' behaviors. Therefore the proposed prediction approach is more suitable for the dynamic negotiation environment by considering its facility and flexibility. Furthermore, we propose an optimization approach to generate optimal offers dynamic by considering negotiants' preferences and lead the negotiation results to the "win-win" outcomes (if applicable) finally.

The rest of this paper is organized as follows: Section 2 introduces the preference prediction approach; Section 3 introduces the optimal offer generation approach; Section 4 illustrates and discusses the experimental results; Section 5 compares this research with related works; and Section 6 concludes this paper and outlines our future works.

2 Preferences Prediction

In this section, the prediction approach on opponents' preferences is introduced based on the regression approach [11]. Subsection 2.1 introduces the prediction approach for single-issue negotiation, and Subsection 2.2 extends the approach to multi-issue negotiation.

2.1 Behaviors Prediction in Single Issue Negotiation

In general, three kinds of negotiation strategies can be employed by agents during a single-issue negotiation, which are *Boulware*, *Conceder* and *Linear* [2]. In Boulware strategies, agents look towards to the maximum profits. Therefore they will not give a great concession until the negotiation deadline. In Conceder strategies, agents normally like to make the deal with opponents as soon as possible, so they will make a great concession at the beginning of the negotiation. In Linear strategies, agents normally make a concession smoothly and sequentially throughout the negotiation. In order to simulate these common negotiation behaviors, we propose the following quadratic regression function.

$$O(t) = a \times t^2 + b \times t + c \quad (1)$$

where $O(t)$ is the predicted opponent's offer at t^{th} ($1 \leq t \leq \tau$, where τ is the deadline) negotiation round, a , b and c are coefficients, and are all independent

on t . It is noticed that the proposed quadratic regression function can simulate the three common negotiation behaviors mentioned above by assigning different values to the coefficients as follows:

- *Boulware* ($a > 0$): the rate of change in the slope is increasing, corresponding to smaller concession in the early rounds but large concession in later rounds.
- *Conceder* ($a < 0$): the rate of change in the slope is decreasing, corresponding to large concession in early round but smaller concession in later rounds.
- *Linear* ($a = 0$ and $b \neq 0$): the rate of change in the slope is zero, corresponding to making constant concession throughout the negotiation.

The aim of this prediction approach is to employ opponents' historical offers to generate a particular function $O(t)$ to predict the opponent's offer generation function. It must be ensured that the differences between the predicted offer and the real offer in all negotiation rounds are minimized. Let set $R = \{\hat{O}(t')\}$ ($t' \in [1, t]$) be the opponent's real historical offers in the previous t rounds. Because the function $O(t)$ is the regression results on set $R = \{\hat{O}(t')\}$, so all distances $\varepsilon(t')$ between the real offers $\hat{O}(t')$ and the predicted offers $O(t')$ in all negotiation rounds should obey the Normal Distribution. Let $\varepsilon(t') = \hat{O}(t') - O(t')$, the joint probability density function for all $\varepsilon(t')$ in the previous t negotiation rounds is:

$$\begin{aligned} L_{\varepsilon(t)} &= \prod_{t'=0}^t \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2} [\hat{O}(t') - O(t')]^2\right\} \\ &= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^t \exp\left\{-\frac{1}{2\sigma^2} \sum_{t'=0}^t [\hat{O}(t') - O(t')]^2\right\} \end{aligned}$$

where $L_{\varepsilon(t)}$ indicates the joint probability that all real offers $\hat{O}(t')$ may happen. Because each $\hat{O}(t')$ comes from the historical record, so we must let $L_{\varepsilon(t)}$ to be its maximum value. Obviously, in order to maintain $L_{\varepsilon(t)}$ to the maximum value, $\sum_{t'=0}^t [\hat{O}(t') - O(t')]^2$ should achieve its minimum value. Let

$$\begin{aligned} Q(a, b, c) &= \sum_{t'=0}^t [\hat{O}(t') - O(t')]^2 \\ &= \sum_{t'=0}^t [\hat{O}(t') - at'^2 - bt' - c]^2 \end{aligned} \tag{2}$$

Then we calculate the partial derivative for $Q(a, b, c)$ on a , b and c , respectively and let their results equal to zero.

$$\begin{cases} \frac{\partial Q}{\partial a} = -2 \sum_{t'=0}^t (\hat{O}(t') - at'^2 - bt' - c)t'^2 = 0 \\ \frac{\partial Q}{\partial b} = -2 \sum_{t'=0}^t (\hat{O}(t') - at'^2 - bt' - c)t' = 0 \\ \frac{\partial Q}{\partial c} = -2 \sum_{t'=0}^t (\hat{O}(t') - at'^2 - bt' - c) = 0 \end{cases} \tag{3}$$

Then the above equations can be simplified to:

$$\begin{cases} (\sum_{t'=0}^t t'^4)a + (\sum_{t'=0}^t t'^3)b + (\sum_{t'=0}^t t'^2)c = \sum_{t'=0}^t t'^2 O(\hat{t}') \\ (\sum_{t'=0}^t t'^3)a + (\sum_{t'=0}^t t'^2)b + (\sum_{t'=0}^t t')c = \sum_{t'=0}^t t' O(\hat{t}') \\ (\sum_{t'=0}^t t'^2)a + (\sum_{t'=0}^t t')b + tc = \sum_{t'=0}^t O(\hat{t}') \end{cases} \quad (4)$$

Let C_u , C_a , C_b and C_c are the coefficient matrices for Equation 4, then:

$$C_u = \begin{vmatrix} \sum_{t'=0}^t t'^4 & \sum_{t'=0}^t t'^3 & \sum_{t'=0}^t t'^2 \\ \sum_{t'=0}^t t'^3 & \sum_{t'=0}^t t'^2 & \sum_{t'=0}^t t' \\ \sum_{t'=0}^t t'^2 & \sum_{t'=0}^t t' & t \end{vmatrix} \quad (5)$$

$$C_a = \begin{vmatrix} \sum_{t'=0}^t t'^2 O(\hat{t}') & \sum_{t'=0}^t t'^3 & \sum_{t'=0}^t t'^2 \\ \sum_{t'=0}^t t' O(\hat{t}') & \sum_{t'=0}^t t'^2 & \sum_{t'=0}^t t' \\ \sum_{t'=0}^t O(\hat{t}') & \sum_{t'=0}^t t' & t \end{vmatrix} \quad (6)$$

$$C_b = \begin{vmatrix} \sum_{t'=0}^t t'^4 & \sum_{t'=0}^t t'^2 O(\hat{t}') & \sum_{t'=0}^t t'^2 \\ \sum_{t'=0}^t t'^3 & \sum_{t'=0}^t t' O(\hat{t}') & \sum_{t'=0}^t t' \\ \sum_{t'=0}^t t'^2 & \sum_{t'=0}^t O(\hat{t}') & t \end{vmatrix} \quad (7)$$

$$C_c = \begin{vmatrix} \sum_{t'=0}^t t'^4 & \sum_{t'=0}^t t'^3 & \sum_{t'=0}^t t'^2 O(\hat{t}') \\ \sum_{t'=0}^t t'^3 & \sum_{t'=0}^t t'^2 & \sum_{t'=0}^t t' O(\hat{t}') \\ \sum_{t'=0}^t t'^2 & \sum_{t'=0}^t t' & \sum_{t'=0}^t O(\hat{t}') \end{vmatrix} \quad (8)$$

Because $C_u \neq 0$, then parameters a , b and c have an unique solution which is:

$$\begin{cases} a = C_a/C_u \\ b = C_b/C_u \\ c = C_c/C_u \end{cases} \quad (9)$$

Then by employing these parameters, we can find a particular function $O(t)$ to represent the opponent's historical offers in the previous t rounds. Furthermore, we can also predict the opponent's future offers in the next i rounds by replacing t by $t + i$.

2.2 Preferences Prediction in Multi-issue Negotiation

In this subsection, we introduce the approach to predict opponents' preferences in bilateral multi-issue negotiations. Let M be the total number of issues in a multi-issue negotiation, then for each single issue m ($m \in [1, M]$), we adopt the behavior prediction approach on the single issue negotiation to generate a particular regression function for the issue m as follows:

$$O(t)^m = a^m \times t^2 + b^m \times t + c^m \quad (10)$$

Since negotiants may have different preferences on the same issue and/or change their preferences as the negotiation environment changes. So by considering the reality, we make the following assumption.

Negotiants give concession to each single issue in a multi-issue negotiation based on the negotiant's preference. The more/less significant the issue is considered by the negotiant, the less/more concession is given by the negotiant on that issue.

Based on the above assumption, we can predict the opponent's concession on each issue and further predict the opponent's preferences. The opponent's possible concession on issue m can be represented by the derivative of the function $U(O(t)^m)^m$ as follows:

$$C(t)^m = \frac{\partial U(O(t)^m)^m}{\partial t} \quad (11)$$

where $U(O(t)^m)^m$ is the profit that the negotiant gained from the opponent's offer $O(t)^m$ on issue m . It is noticed that the greater the $C(t)^m$ is, the less significant the issue m is considered by the opponent, and the greater concession that the opponent would like to make on the issue m . Let $W(t)^m$ be the opponent's preference on the issue m at the round t , then $W(t)^m$ can be calculated as follows.

$$W(t)^m = \frac{1/C(t)^m}{\sum_{n=1}^M 1/C(t)^n} = \frac{\prod_{n=1, n \neq m}^M C(t)^n}{\sum_{n=1}^M (\prod_{p=1, p \neq n}^M C(t)^p)} \quad (12)$$

Then by calculating all $W(t)^m$ ($m \in [1, M]$) for all negotiated issues, we can outline the opponent's preferences at the round t . In the next section, we will employ the predicted preferences to help negotiants to achieve the "win-win" outcomes in the multi-issue negotiation.

3 Optimization Agreements

In this section, we introduce the approach to generate the optimal outcome in the bilateral multi-issue negotiation by employing the predicted preferences in previous Section.

Let agent p and agent q be the two negotiants. For agent p (same as for agent q), we assume that it already knows its own negotiation strategies, utilities functions and preferences, namely $O(t)_p^m$, $U(\text{offer})_p^m \in [0, 1]$ and $W(t)_p^m \in [0, 1]$, where $m \in [1, M]$, $t \in [1, \tau_p]$ and τ_p is agent p 's deadline. By employing the prediction approach introduced in Section 2, agent p can estimate its opponent's (agent q) preferences. Based on the predicted preferences, we introduce the approach on how to generate the optimal outcomes in this section.

Let $U(O(t)_q^m)_p^m$ be the profit which agent p gained from agent q 's offer $O(t)_q^m$ on the issue m at the round t , and $U(O(t)_q)_p$ be the overall profit that agent p gained from agent q by considering all negotiated issues at the round t , so

$$U(O(t)_q)_p = \sum_{m=1}^M [U(O(t)_q^m)_p^m \times W(t)_p^m] \quad (13)$$

Then the overall profit gained by agent q from the offer $O(t)_q$ can also be predicted by agent p as follows:

$$U(O(t)_q)_q = \sum_{m=1}^M [1 - U(O(t)_q^m)_p^m] \times W(t)_q^m \quad (14)$$

where $W(t)_q^m$ is agent p 's prediction on agent q 's preference which can be calculated by formula (12). $[1 - U(O(t)_q^m)_p^m]$ is agent p 's prediction on agent q 's profit. The reason we perform this kind of prediction is based on the “pie splitting” theory. According to the “pie splitting” theory, if the whole profit of an item is 1 and one negotiant claims u ($u \in [0, 1]$) out of 1, then the other negotiant's profit is $1 - u$. In the multi-issue negotiation, situations on each single issue can be treated similar to the “pie splitting” game, so $[1 - U(O(t)_q^m)_p^m]$ can be employed by agent p to represent agent q 's profit approximately.

It is proposed that the agent p 's optimal offer at the round t should satisfy two requirements: (1) the optimal offer can maximize agent q 's profits. The reason behind this condition is based on the real situation that all bargaining agents try to gain as much profits as they can during the negotiation; and (2) the optimal offer can minimize the opponent's loss. The reason behind this consideration is that the opponent definitely will not accept an offer which damages its profits too much. Therefore, in order to make the optimal offer more efficient, this consideration should also be satisfied as much as possible.

In order to find out the optimal offer, we firstly transfer the issue to an optimization problem and then get the optimal offer by solving the optimization problem. In this paper, we are going to employ the Lagrange Multipliers in the optimization problem solving process. Let $U(O(t+1)_p)_p$ be the function which agent p wants to maximize and equation $U(O(t+1)_p)_q - U(O(t)_q)_q = c$ is the constraint, where c indicates the benefit that agent q may lost. In order to minimize agent q 's loss, we set c 's default value as 0. If the solution for Lagrangian cannot be achieved, we can loosen the restriction and enlarge c 's value gradually according to predefined step, such as 0.1. So the Lagrangian, for agent p , is defined as follows:

$$\Lambda(t, \lambda) = U(O(t+1)_p)_p + \lambda[U(O(t+1)_p)_q - U(O(t)_q)_q] \quad (15)$$

By setting the partial derivative for $\Lambda(t, \lambda)$ on t and λ to zero respectively, we can get formula (16) as follows:

$$\begin{cases} \frac{\partial \Lambda(t, \lambda)}{\partial t} = \frac{\partial U(O(t+1)_p)_p}{\partial t} + \frac{\partial \lambda U(O(t+1)_p)_q}{\partial t} = 0 \\ \frac{\partial \Lambda(t, \lambda)}{\partial \lambda} = U(O(t+1)_p)_q - U(O(t)_q)_q = 0 \end{cases} \quad (16)$$

By solving the formula (16), we may get three possible results, which are:

1. No solution, so we should relax the restriction and enlarge the opponent's loss, then recalculate the optimal offer;

2. Single solution, t_o , which is the optimal solution;
3. Multiple solution, namely set $\mathbf{t_s}$. Then the optimal solution t_o can be defined as follows:

$$\forall t \in \mathbf{t_s}, \exists t_o \in \mathbf{t_s} \Rightarrow U(O(t_o + 1)_p) \geq U(O(t + 1)_p) \quad (17)$$

Finally, by employing the optimal solution t_o , agent p 's optimal offer $\mathbf{O(t_o + 1)_p}$ can be represented by formula (18) as follows:

$$\mathbf{O(t_o + 1)_p} = \{O(t_o + 1)_p^m\} \quad (18)$$

where $m \in [1, M]$ and $O(t_o + 1)_p^m$ is the optimal offer for the issue m only.

In this section, we introduced the approach to generate the optimal offer in bilateral multi-issue negotiation. During the negotiation, both sides of negotiators can employ this optimal approach alternatively until an optimal agreement (when the optimal agreement is applicable) is agreed by all negotiators or one negotiant quit the negotiation (when the optimal solution does not exist). In the following section, experiments on the proposed optimal approach is introduced.

4 Experiment

In this section, we exam our proposed prediction and optimization approaches through experiments and compare to the results of NDF negotiation approach [2].

4.1 Experimental Setup

In order to simply the experiments, we employ three agents (one seller and two buyers) in an two issues negotiation. Of course, the proposed approach can be applied on negotiation with any number of issues. The scenario is described as follows: the agent *seller* wants to sell a car and a GPS navigation system, both buyer agents *buyer1* and *buyer2* wants to purchase these two items from the *seller*. In order to make the negotiation results comparable, all agents have same acceptable range on the items' prices, which are from \$4000 to \$5000 for the

Table 1. Experimental cases

Case #	Seller's strategies	Buyer's strategies
1	Car (C), GPS (B)	Car (C), GPS (B)
2	Car (C), GPS (B)	Car (L), GPS (L)
3	Car (C), GPS (B)	Car (B), GPS (C)
4	Car (L), GPS (L)	Car (C), GPS (B)
5	Car (L), GPS (L)	Car (L), GPS (L)
6	Car (L), GPS (L)	Car (B), GPS (C)
7	Car (B), GPS (C)	Car (C), GPS (B)
8	Car (B), GPS (C)	Car (L), GPS (L)
9	Car (B), GPS (C)	Car (B), GPS (C)

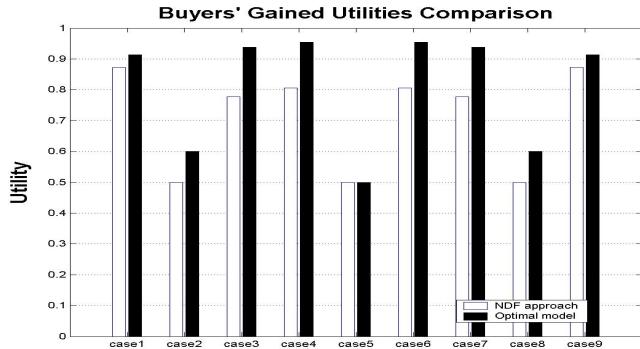


Fig. 1. The buyers' gained utilities comparison

car and from \$200 to \$300 for the GPS. The deadline for all negotiants are also same, which is the 10th negotiation round. All agents employ the *package deal* procedure [5] and the alternating offering protocol is employed in the negotiation. No agent wants to share its own private information with others. In order to simulate the open and dynamic negotiation environment, *seller* will change its preferences dynamic based on its profits, which is *Seller* would like to make less concession on the issue whose price is near its 'bottom line' and to make more concessions on other issues. During the negotiation, both *seller* and *buyer1* employs the NDF negotiation approach and *buyer2* employs the proposed negotiation approach. In order to make the negotiation results general enough, each agent can have three options on negotiation strategy on each negotiated item, which can cover almost all common situations in the multi-issue negotiation. The three common negotiation strategies are *Boulware* (B), *Conceder* (C) and *Linear* (L) (refer to Subsection 2.1). We progress the experiments according to the nine cases shown in Table 1.

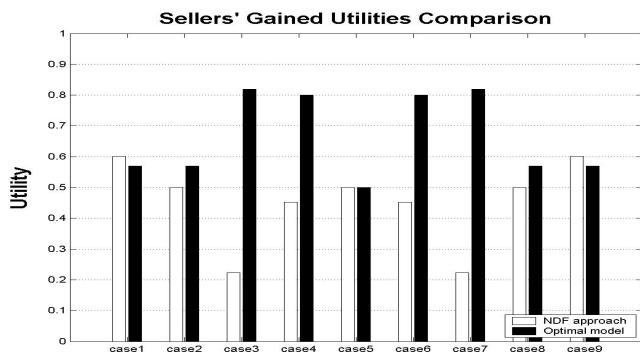
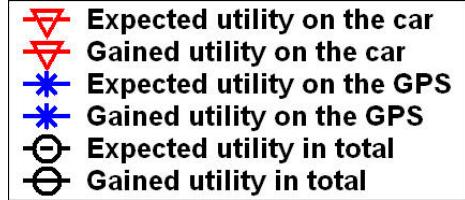


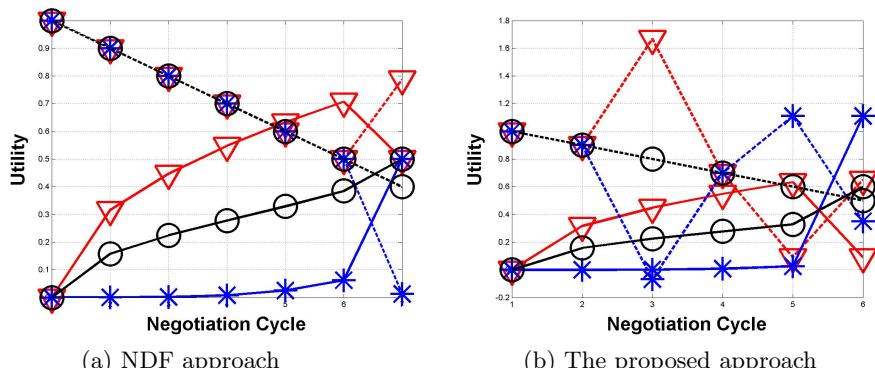
Fig. 2. The sellers' gained utilities comparison

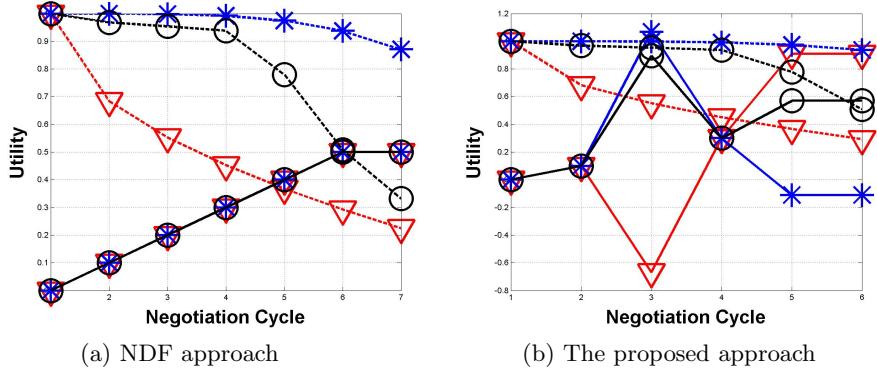
**Fig. 3.** Legend

4.2 Experimental Results

In the experiment, agents *buyer1* and *buyer2* negotiates with the agent *seller* according to the nine cases in Table 1. Both *buyer1* and *seller* employ the NDF negotiation approach, and *buyer2* employ the proposed optimal approach. We compare the two negotiation outcomes by considering negotiants' profits and time spending on the negotiation. The comparison results for both sellers and buyers in all nine cases are displayed in Fig. 1 and Fig. 2.

In Fig. 1, we compare two buyers' profits by employing different negotiation approaches. The x-axis represents the nine cases and the y-axis represents the profit agents gained from the negotiation. It can be seen that almost in all cases, *buyer2* can gain more profits (around 10%) than *buyer1*. In Fig. 2, we compare *seller*'s profits by employing two approaches. It can be seen that except case 1 and 9, *seller* gains much more profits by employing the proposed optimal approach. Especially in case 3 and 7, the advance is more than 60%. The explanation on this comparison results is that in the NDF negotiation approach, *buyer1* only considered its own profits but ignored *seller*'s profits. Therefore, most of *buyer1*'s offer are rejected by the *seller*. However, by employing the optimal approach, *buyer2* not only try to maximize its self's profits, but also try to minimize *seller*'s loss. So the *buyer2*'s offers are more acceptable for the *seller*, and the negotiation were led to the “win-win” outcome. Furthermore, according to experimental results, the proposed optimal approach save more than 10%

**Fig. 4.** Case 2: buyer's benefit

**Fig. 5.** Case 2: seller's benefit

negotiation time in average than the NDF approach. Therefore, to summarize the experimental results, the proposed optimal approach can achieve a better negotiation outcome and spend less negotiation time by comparing with the NDF negotiation approach in the bilateral multi-issue negotiation. Also, by analyzing the experimental results in all cases, we indicate the following discovery:

In the multi-issue negotiation, the possibility that negotiators can gain more profits from the optimal offer is direct ratio to differences between negotiators' preferences. The greater the differences are, the more profits negotiators can gain from the optimal offer, and vice versa.

Because of the pages limitation, we only illustrate the detail about the experimental results in case 2. In case 2, *seller* employs the *Conceder* negotiation strategy on the car and the *Boulware* negotiation strategy on the GPS. All buyers employ the *Linear* negotiation strategy on all negotiated items. The legend of the result figures are displayed in Fig. 3. By comparing the gained profits between two buyers (see Fig. 4), it can be seen that because *buyer2* can predict *seller*'s preferences, so *buyer2* gained 10% more profits than *buyer1*. For detail, at the 3rd negotiation round, *buyer2* makes a great concession on the GPS but enlarge its expected profits on the car to respond *seller*'s changes on the preferences. In the 5th round, *buyer2* noticed that *seller* would like to decrease its concession on the car but increase its concession on the GPS, so *buyer2* gave more concession on the car and requiring more benefits from the GPS. Through these kinds of modification, *buyer2* found the optimal offer to maximize own profits, and also to minimize *seller*'s loss. Furthermore, according to Fig. 5, *seller*'s profits is also increased by 10% through negotiating with *Buyer2* than with *Buyer1*.

5 Related Work

Some related works about searching the optimal agreements in multi-issue negotiations have been proposed by researchers [1] [2] [3] [5]. To list few of them,

in [4], Fatima et. al. discovered that the outcome of the multi-issue negotiation depends on the agenda and the negotiation procedure. They analyzed both package deal and issue-by-issue negotiation procedures and gave equilibrium strategies for both procedures. Furthermore, the authors introduced their proposed optimal strategy in two-issues negotiation. Based on whether a particular issue has or does not have a zone of agreement, the authors divided all possible situations in two-issues negotiation into four scenarios. For each scenario, the authors also introduced a particular method to achieve optimal outcomes. The contribution of their work is presenting an optimal two-issues negotiation strategy and pointing out a possible way for optimal multi-issue negotiation. However, their work only presented the situation of multi-issue negotiation in static negotiation environments and also did not consider the impact from agents' preferences.

In [6], Lai et. al. proposed a Pareto optimal model for multi-issue negotiations. In this model, the authors did not calculate the optimal agreement directly, but employed an enhancement process to update offers in each negotiation round and approximated to the optimal agreement gradually. The enhancement process works as follows: in each negotiation round, based on the original offer and the enhancement range, the agent can calculate another two enhancement offers (averages between the original offer and the lower and upper bounds of the enhancement range). Then the agent will send all enhancement offers with the original offer to its opponent. Based on the opponent's response, the agent will modify its count-offer in the next negotiation round and the count-offer will approximate to the optimal agreement gradually. The advantage of their optimal strategy is easy to implement. However, the disadvantages are also evident, i.e. in some situations, it is very difficult to find out the enhancement bounders. Also by considering the time limitation in negotiation, this enhancement process may not efficient enough to approximate an offer to the optimal agreement before the deadline. Comparing to this work, our proposed approach is more efficient and suitable to be employed in open and dynamic negotiation environments.

6 Conclusion and Future Work

In this paper, we proposed an optimal approach for bilateral multi-issue negotiations in open and dynamic environments. According to experimental results, the proposed approach can predict the opponent's preferences based on the historical offers of the current negotiation reasonably, and can find out the optimal offer (if it is applicable) based on the predicted preferences efficiently. In the future, we would like to extend this approach from the bilateral negotiation to the multi-lateral negotiation and exam the performance in more complex negotiation environments.

References

1. Lai, G., Li, C., Sycara, K.: A General Model for Pareto Optimall Multi-Attribute Negotiations. In: Second International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS 2006), Future University, Hakodate, Japan, pp. 55–76 (May 2006)

2. Fatima, S., Wooldridge, M., Jennings, N.: An Agenda-Based Framework for Multi-Issue Negotiation. *Artificial Intelligence* 152(1), 1–45 (2004)
3. Bosse, T., Jonker, C., Treur, J.: Experiments in Human Multi-Issue Negotiation: Analysis and Support. In: Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004), pp. 671–678. IEEE Computer Society, Los Alamitos (2004)
4. Fatima, S., Wooldridge, M., Jennings, N.: Optimal Negotiation of Multiple Issues in Incomplete Information Settings. In: International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), pp. 1080–1087. IEEE Computer Society, Los Alamitos (2004)
5. Fatima, S., Wooldridge, M., Jennings, N.: Multi-Issue Negotiation with Deadlines. *Journal of Artificial Intelligence Research (JAIR)* 27, 381–417 (2006)
6. Lai, G., Sycara, K., Li, C.: A Pareto Optimal Model for Automated Multi-attribute Negotiations. In: Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2007), Honolulu, hawaii, IFAAMAS, pp. 1040–1042 (May 2007)
7. Zeng, D., Sycara, K.: Bayesian Learning in Negotiation. *International Journal of Human-Computer Studies* 48(1), 125–141 (1998)
8. Gal, Y., Pfeffer, A.: Predicting Peoples Bidding Behavior in Negotiation. In: 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, pp. 370–376. ACM, New York (2006)
9. Coehoorn, R., Jennings, N.: Learning on Opponent's Preferences to Make Effective Multi-issue Negotiation Trade-offs. In: Proceedings of the 6th International Conference on Electronic Commerce, ICEC 2004, Delft, Netherlands, pp. 59–68. ACM Press, New York (2004)
10. Chajewska, U., Koller, D., Ormoneit, D.: Learning An Agent's Utility Function by Observing Behavior. In: Proc. 18th International Conf. on Machine Learning, pp. 35–42. Morgan Kaufmann, San Francisco (2001)
11. Ren, F., Zhang, M.: Prediction of Partners Behaviors in Agent Negotiation under Open and Dynamic Environment. In: The 3rd International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS 2007). Springer, Heidelberg (2007)

The Density-Based Agglomerative Information Bottleneck^{*}

Yongli Ren¹, Yangdong Ye¹, and Gang Li²

¹ School of Information Engineering, Zhengzhou University, Zhengzhou, China
yonglitom@gmail.com, yeyd@zzu.edu.cn

² School of Engineering and Information Technology, Deakin University,
221 Burwood Highway, Vic 3125, Australia
gang.li@deakin.edu.au

Abstract. The *Information Bottleneck* method aims to extract a compact representation which preserves the maximum relevant information. The sub-optimality in agglomerative Information Bottleneck (aIB) algorithm restricts the applications of *Information Bottleneck* method. In this paper, the concept of *density-based chains* is adopted to evaluate the information loss among the neighbors of an element, rather than the information loss between pairs of elements. The DaIB algorithm is then presented to alleviate the sub-optimality problem in aIB while simultaneously keeping the useful hierarchical clustering tree-structure. The experiment results on the benchmark data sets show that the DaIB algorithm can get more relevant information and higher precision than aIB algorithm, and the paired t-test indicates that these improvements are statistically significant.

Keywords: Information Bottleneck, density, hierarchical tree-structure.

1 Introduction

Extracting relevant features from a complex data is a fundamental task in machine learning. The problem is often that the data contains many features, and it is difficult to define which of them are relevant in an unsupervised manner. The *Information Bottleneck* (IB) method [1] is one of the methods which principally address this problem, and the idea of IB method is to model features extraction as data compression and to quantify the relevance of the extracted feature by how much information it preserves about a specified relevance feature.

Given a joint distribution $p(x, y)$, the *Information Bottleneck* method construct a new representation variable T that defines partitions over the elements of X that are informative about Y . The compactness of the representation is then determined by the mutual information $I(T; X)$, while the accuracy of the representation is measured by the mutual information $I(T; Y)$. The *Information*

* This research was supported by the National Science Foundation of China under grant No. 60773048., and Deakin CRGS Grant 2008.

Bottleneck method proposes to find a stochastic mapping, parameterized by probability $p(t|x)$ that maximizes:

$$F_{max} = I(T; Y) - \beta I(T; X) \quad (1)$$

where the β is a positive number which balances the trade-off between compression and accuracy. As $\beta \rightarrow \infty$, the IB objective function becomes the maximization of $I(T; Y)$, the mutual information between T and Y . In this situation, the $p(t|x)$ will approach zero or one almost everywhere.

Without any assumption about the origin of the joint distribution $p(x, y)$, Tishby et al. show that the IB problem has an exact optimal solution [1]. However, how to construct the optimal or approximated solutions remains a problem. Several algorithms have been developed for the IB problem (see [2] for detailed review and comparison), such as, the iterative IB algorithm [1], the agglomerative IB (aIB) algorithm [3], and the sequential IB (sIB) algorithm [4].

Among those algorithms, the hierarchical tree-structure output of the aIB algorithm makes itself unique. The aIB algorithm arranges the elements into a tree-structure, which could be further used to process databases for effective browsing and speed up search-by query. It starts with the trivial partition in which each element $x \in X$ represents a singleton cluster or component $t \in T$. To minimize the loss of mutual information $I(T; Y)$, the aIB algorithm merges “the most possible merging pair” which locally minimizes the loss of $I(T; Y)$ at each step. Let t_i and t_j be two elements of T , the information loss due to the merging of t_i and t_j is defined as [2]:

$$d(t_i, t_j) = (p(t_i) + p(t_j)) \cdot \bar{d}(t_i, t_j), \quad (2)$$

where $\bar{d} \equiv JS_{\Pi}[p(y|t_i), p(y|t_j)] - \beta^{-1}JS_{\Pi}[p(x|t_i), p(x|t_j)]$, and $JS_{\Pi}[p, q]$ is the Jensen-Shannon divergence between two distributions $p(\cdot)$ and $q(\cdot)$, here $\Pi = \left\{ \frac{p(t_i)}{p(t_i)+p(t_j)}, \frac{p(t_j)}{p(t_i)+p(t_j)} \right\}$.

J. Goldberger et al. apply the aIB to unsupervised image clustering as a processing step for efficient image retrieval [5]. However, as a greedy process, the aIB only merges the element pair which minimizes the information loss, and there is no guarantee to preserve the relevant information as much as possible. In addition, when there are more than one pair of elements with the same minimal information loss, the algorithm will randomly choose one pair to merge. These usually lead to a sub-optimal clustering result. S. Gordon et al. use the sIB and K-means to improve accuracy of the aIB results, but their method could not generate a hierarchical clustering structure which is important for many applications.

In this paper, we introduce the concept of *density-based chains*, through which both the information between two elements, and the information among the neighbors of an element can be considered. Based on this idea, a new algorithm DaIB is presented to alleviate the sub-optimality problem in aIB while simultaneously keeping the useful hierarchical clustering tree-structure.

The rest of the paper is organized as follows. In section 2, we define the density-based chain, and propose the DaIB algorithm. In section 3, we present

the experiment results that evaluate the performance of DaIB compared to aIB under the document clustering scenario. Finally, in section 4, conclusions and future work are presented.

2 The DaIB Algorithm

The aIB algorithm suffers from the sub-optimality problem which makes the aIB algorithm fail to preserve as much mutual information as possible. To alleviate this problem, we introduce the density-based agglomerative Information Bottleneck (DaIB) algorithm. Based on the density-based chain defined in section 2.1, in each step, we will find the “best mergers” from a series of chains in which each of them contains at least two components of the current partition T_i .

2.1 Density-Based Chains

As one of the important methods in data mining, clustering analysis can be used to reveal the hidden structure in a given data set [2]. Considering the sample data set in Fig. 1, we can easily discover these two clusters. The main reason why these two clusters are obvious is that they have different element distribution densities. To discover the clusters of arbitrary shape, M. Ester et al. propose the DBSCAN algorithm [6], which defines an ϵ -neighborhood of an object to decide whether the neighborhood of an object is dense enough, and uses the density connectivity to discover the clusters. Inspired by this, the DaIB algorithm will adopt the density-based chains to discover the hidden IB structure of a data set, and use it with the IB-based information measurement rather than a distance as in the DBSCAN algorithm.

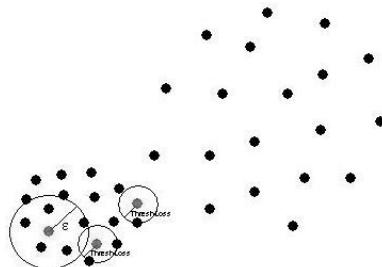


Fig. 1. The sample data set

Let $MinLoss$ denote the minimal information loss of element pairs in current partition. The threshold for information loss, $ThreshLoss$, is defined as:

$$ThreshLoss = MinLoss * r, \quad (3)$$

where r is a predefined parameter.

For a component t_i and one of its neighbors t_j , if $d(t_i, t_j) < \text{ThreshLoss}$, t_i and t_j are considered to be dense enough and we denote them as $\{t_i, t_j\}$. Let t_k denote another neighbor of t_i . If $d(t_i, t_k) < \text{ThreshLoss}$, t_i and t_k are dense enough and we denote it as $\{t_i, t_k\}$. This, together with $\{t_i, t_j\}$, makes t_j, t_i and t_k dense enough to be merged together, and we denote it as $\{t_j, t_i, t_k\}$. This procedure continues until all neighbors which satisfy that its information loss is less than ThreshLoss have been added. The resulted groups of components are called *Density-based Chains*.

By this procedure, the DaIB algorithm considers both the information loss between two elements, and the information loss among the neighbors of an element. This is different from the aIB algorithm which considers only the information loss between two elements. The aIB merges two elements into one at each step, while the DaIB typically will merge all elements in the same density-based chain components at each step. Considering that it may exists several density-based chains at each step, the DaIB merges the elements in each component of the chain into a new element, accordingly, in general more than one new elements will be generated in every partition.

To identify the density-based chains, firstly we have to find out all those merging pairs $d(t_i, t_j)$ with information loss less than the ThreshLoss . Let S be the set of these merging pairs. If we use nodes to represent components, and use the undirected link to connect two components if their merging cost is less than the ThreshLoss , then the S can be represented as an undirected graph. Fig. 2 gives one example which contains three density-based chains in S : $\{t_1, t_2, t_3\}$, $\{t_4, t_5, t_6, t_7\}$ and $\{t_8, t_9\}$ with 3, 4 and 2 components respectively.

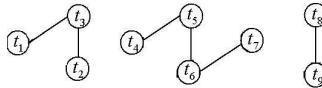


Fig. 2. The resulting graph of S

For every chain in S , we merge all the components in this chain as a new component \bar{t} , so the chains in Fig. 2 will be merged into 3 new components. The probability distribution $p(\bar{t}), p(y|\bar{t})$ and $p(\bar{t}|x)$ of the new component is calculated as:

$$\begin{aligned} p(\bar{t}) &= \sum_{i=1}^k p(t_i) \\ p(y|\bar{t}) &= \frac{1}{p(\bar{t})} \sum_{i=1}^k p(t_i, y) \quad \forall y \in Y \\ p(\bar{t}|x) &= \begin{cases} 1 & \text{if } x \in t_i, \text{ for some } 1 \leq i \leq k \\ 0 & \text{otherwise} \end{cases} \quad \forall x \in X \end{aligned} \tag{4}$$

where t_i and k denote the component and the number of these components in this chain respectively.

2.2 The DaIB Algorithm

The DaIB algorithm proceeds mainly in two phases, the initialization phase and the iterative search phase. In the initialization phase, all the potential information losses are calculated. In the iterative search phase, the density-based chains are found and each component is merged to produce a clustering result.

The Initialization phase: Initiate every element $x \in X$ as a singleton cluster or component, and calculate the information losses between each potential merging pairs by equation (2).

The Iterative Search Phase: Find the set S of the merging pairs (t_i, t_j) that their information lossss satisfy $d(t_i, t_j) < ThreshLoss$. Use the graph-based method described in section 2.1 to find all the density-based chains in S . For each chain in S , we merge all the components in this chain into a new component \bar{t} . The probability distribution $p(\bar{t})$, $p(y|\bar{t})$ and $p(\bar{t}|x)$ of the new component are calculated by equation (4). This procedure is repeated until T degenerates into a single value.

2.3 The Analysis of the DaIB Algorithm

For the current partition T_i , let m denote the number of density-based chains in T_i and let n denote the number of components in these chains. Obviously, the partition T_j after merging these chains satisfies $|T_j| = |T_i| - n + m$. So the cardinality of T in the DaIB algorithm does not degenerate one by one, and it will generate the hierarchical clustering structure in less steps. Through the density-based chain, the DaIB algorithm consider both the information loss between two components, and the information losses among a density chain. At each merging step of DaIB, the merger decision is unique and there is no arbitrary choice as in the aIB algorithm, where at each step there might have several minimas and the aIB chooses one of them arbitrarily to merge.

3 Experiment Design and Results Analysis

In this section, we compare the performance of the DaIB algorithm and the aIB algorithm under the document clustering scenario, as in the papers [7,8,2,4]. The nine data sets used in our experiment are subsets of the 20-Newsgroup corpus [9]. Each of the nine data sets consists of 500 documents randomly chosen from the themes in the 20-Newsgroup corpus. All of the chosen documents have been preprocessed by:

- Removing file headers, and leaving only the subject line and the body.
- Lowering all upper-case characters.
- Uniting all digits into a single “digit” symbol.
- Ignoring the non-alpha-numeric characters.
- Removing the stop-words and these words that appeared only once.
- Keeping only top 2000 words according to their contribution.

Algorithm 1. The DaIB algorithm

Input:

The joint distribution $p(x, y)$.

The parameter r .

Output:

A pruned hierarchical clustering tree-structure.

Algorithm Process:

```

1: Initialization:
2: for every  $x_i \in X$ ,  $1 \leq i \leq |X|$ 
3:    $t_i = \{x_i\}$ ;
4:    $p(t_i) = p(x_i)$ ;
5:    $p(y|t_i) = p(y|x_i)$ ;
6:    $p(t_i|x_i) = 1$ ;
7: end for
8: for every pair  $(t_i, t_j)$ ,  $i < j$ , calculate
9:    $d(t_i, t_j) = (p(t_i) + p(t_j)) \cdot \bar{d}(t_i, t_j)$ ;
10: end for
11: Iterative Search:
12: while  $|T| > 1$ 
13:   find  $MinLoss = \min\{d(t_i, t_j)\}$ ;
14:    $ThreshLoss = MinLoss * r$ ;
15:   for every pair  $(t_i, t_j)$ ,  $i < j$ .
16:     if  $d(t_i, t_j) < ThreshLoss$ 
17:       insert  $(t_i, t_j)$  into  $S$ ;
18:     end if
19:   end for
20:   find the density-based chains in  $S$ ;
21:   for every density-based chain:  $dcc$ , in  $S$ 
22:     merge the components  $(t_i, t_j, \dots)$  in  $dcc \Rightarrow \bar{t} :$ 
23:      $p(\bar{t}) = \sum_{i=1}^k p(t_i)$ 
24:      $p(y|\bar{t}) = \frac{1}{p(\bar{t})} \sum_{i=1}^k p(t_i, y) \quad \forall y \in Y$ 
25:      $p(\bar{t}|x) = 1$  if  $x \in t_i, 1 \leq i \leq k$ ;
26:     update  $T = \{T - \{t_i, t_j, \dots\}\} \cup \{\bar{t}\}$ ;
27:     update the probability distributions w.r.t.  $\bar{t}$ ;
28:   end for
29: end while

```

In each of these nine data sets, there is a sparse document-word count matrix: each row of the matrix denotes a document $x \in X$; each column denotes a word $y \in Y$; the matrix value $m(x, y)$ denotes the occurrence times of the word y in the document x . The detailed information about these nine data set is summarized in Table 1.

3.1 The Evaluation Method

In this paper, we use the *mutual information*, *micro-averaged precision* and *recall* as the quantitative measures.

Table 1. Data Sets

Data Sets	Associated Themes	Category	Size
Binary_1	talk.politics.mideast, talk.politics.misc	2	500
Binary_2	talk.politics.mideast, talk.politics.misc	2	500
Binary_3	talk.politics.mideast, talk.politics.misc	2	500
Multi5_1	comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, talk.politics.mideast	5	500
Multi5_2	comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, talk.politics.mideast	5	500
Multi5_3	comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, talk.politics.mideast	5	500
Multi10_1	alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.guns	10	500
Multi10_2	alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.guns	10	500
Multi10_3	alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.guns	10	500

Mutual Information. The main idea of the IB method is to extract a compact representation T , which preserves the maximal mutual information $I(T; Y)$, so we use the mutual information $I(T; Y)$ to compare clusterings. The higher the mutual information, the better the clustering. As our approach produces a pruned tree-structure, we can compare the mutual information on the same cardinalities $|T|$ with the aIB algorithm.

Micro-averaged Precision and Recall. Following [4, 2], the micro-averaged precision and recall are also used as a quantitative measure. Firstly, we define the labels of all documents in some cluster $t \in T$ as the most dominate label in that cluster. Then, for each category $c \in C$ we use: $A_1(c, T)$ to denote the number of the documents which are assigned to c correctly, and use $A_2(c, T)$ to denote the number of the documents which are assigned to c incorrectly, and use $A_3(c, T)$ to denote the number of the documents which are not assigned to c incorrectly.

The micro-averaged precision is defined as:

$$P(T) = \frac{\sum_c A_1(c, T)}{\sum_c A_1(c, T) + A_2(c, T)}.$$

The micro-averaged recall is defined as:

$$R(T) = \frac{\sum_c A_1(c, T)}{\sum_c A_1(c, T) + A_3(c, T)}.$$

When the data sets and the algorithm are both uni-labeled, we will have $P(T) = R(T)$. For this reason, we will just need to use the $P(T)$ only in this paper.

3.2 The Parameter Involved in DaIB

The parameter r is an important parameter in the DaIB algorithm, and it determines all the density-based chains in the current partition. A larger r could increase the number of density-based chains and will emphasize on the information among the neighbors; while a smaller value will decrease the number of the density-based chains and emphasize on the information between two components. Fig. 3 shows the preserved mutual information for the different values of r on all the nine experiment data sets, and indicates that the value $r = 1.015$ could get general good results on all the nine data sets. Especially, it is a good one on the data sets Binary_2, Multi5_1, Multi5_2, Multi10_2 and Multi10_3. So we recommend that r takes the value of 1.015.

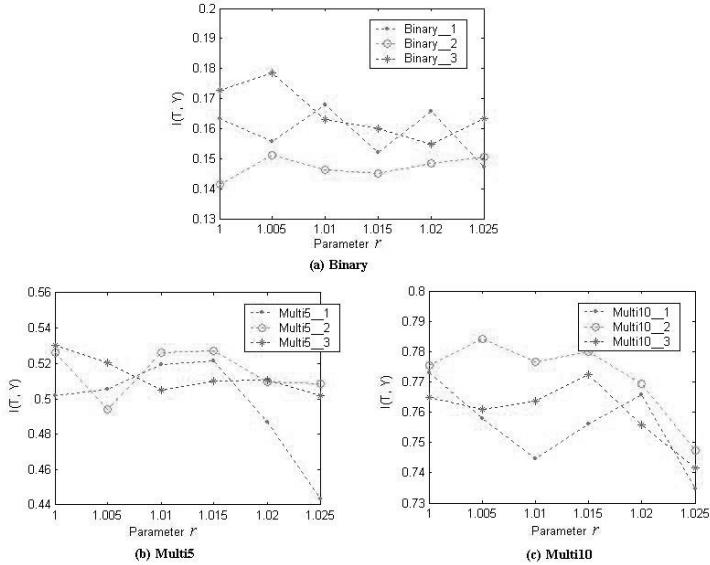


Fig. 3. The effect of the parameter r

3.3 The Comparison of the Mutual Information

The hierarchical clustering tree-structure generated by the DaIB algorithm is pruned to some cardinalities $|T|$, so that the mutual information on the same cardinalities can be compared. We run the aIB algorithm and the DaIB algorithm on the data set, respectively. On the generated hierarchical structures, for each equal cardinality $|T|$, we compare these two algorithm on the *ratio* defined as:

$$ratio = \frac{I_{DaIB}(T; Y)}{I_{aIB}(T; Y)},$$

where $I_{DaIB}(T; Y)$ and $I_{aIB}(T; Y)$ denote the mutual information preserved by DaIB and aIB respectively. The higher the *ratio*, the better the DaIB algorithm. Fig. 4(a)-(c) give the *ratio* on the last 100 cardinalities got from DaIB algorithm on each data sets. The DaIB algorithm successfully preserves more mutual information in 774 out of all 900 same cardinalities. Fig. 4(d)-(f) show the *ratio* on the last 10 cardinalities got from the DaIB algorithm on each data set. It is evident that the less the cardinality of T is, the higher the *ratio* is. The detailed mutual information when $2 \leq |T| \leq 10^1$ is presented in table 2, in which the bold values are corresponding to the known cardinality of the data set.

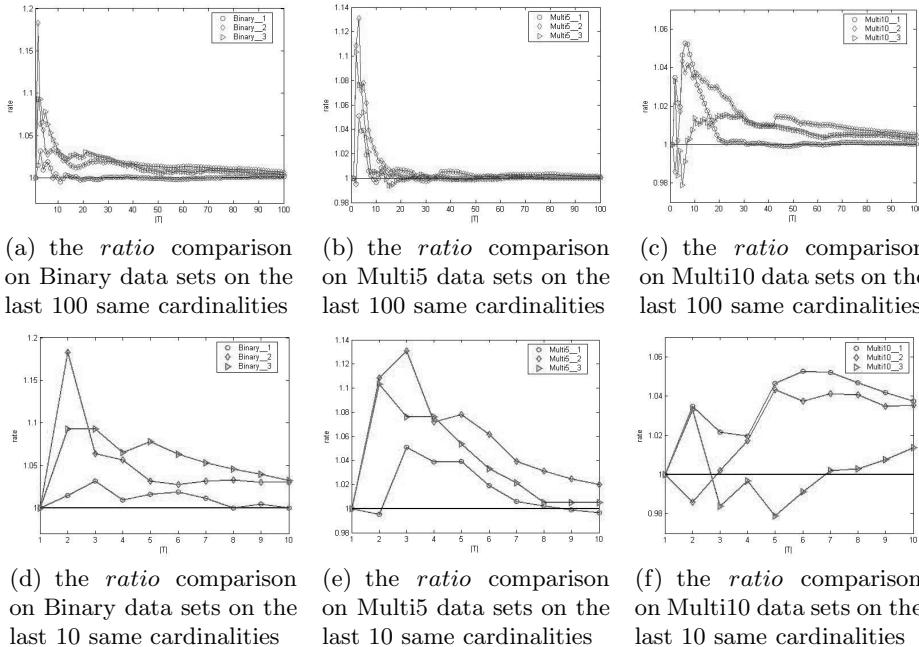


Fig. 4. The comparison of the mutual information on the nine data sets

3.4 The Comparison of the Micro-averaged Precision

The DaIB algorithm can achieve higher micro-averaged precision than aIB on all the nine data sets. The detailed micro-averaged precision results are presented in Table 3. The most notable improvement achieves 28.4% on the data set Binary_2,

¹ In DaIB algorithm, $ThreshLoss = MinLoss * r$, the parameter r decides the number of the density-based chains. In each step, the DaIB algorithm may merge more than two elements and this will lead to some cardinalities to be pruned. For each reason, the *ratio* is compared on the last 100 cardinalities produced by the DaIB algorithm.

Table 2. The detailed mutual information on the nine data sets when $2 \leq |T| \leq 10$

Data sets	Alg.	$ T = 2$	$ T = 3$	$ T = 4$	$ T = 5$	$ T = 6$	$ T = 7$	$ T = 8$	$ T = 9$	$ T = 10$
Binary_1	aIB	0.1688	0.2739	0.3723	0.4486	0.5219	0.5826	0.6424	0.6915	0.7394
	DaIB	0.1713	0.2825	0.3758	0.4559	0.5316	0.5893	0.6425	0.6947	0.7392
Binary_2	aIB	0.1445	0.2564	0.3498	0.4314	0.5006	0.5579	0.6122	0.6651	0.7138
	DaIB	0.1709	0.2727	0.3696	0.4451	0.5143	0.5755	0.6323	0.6850	0.7355
Binary_3	aIB	0.1729	0.2741	0.3671	0.4418	0.5105	0.5721	0.6240	0.6733	0.7211
	DaIB	0.1889	0.2996	0.3910	0.4763	0.5429	0.6025	0.6525	0.6998	0.7445
Multi5_1	aIB	0.1750	0.3075	0.4125	0.5018	0.5867	0.6558	0.7167	0.7751	0.8295
	DaIB	0.1742	0.3232	0.4289	0.5215	0.5979	0.6597	0.7183	0.7742	0.8267
Multi5_2	aIB	0.1772	0.3068	0.4264	0.5130	0.5955	0.6685	0.7276	0.7854	0.8404
	DaIB	0.1964	0.3469	0.4571	0.5530	0.6322	0.6946	0.7503	0.8047	0.8573
Multi5_3	aIB	0.1827	0.3262	0.4391	0.5303	0.6066	0.6723	0.7342	0.7840	0.8304
	DaIB	0.2016	0.3510	0.4726	0.5587	0.6267	0.6868	0.7377	0.7881	0.8347
Multi10_1	aIB	0.1732	0.2862	0.3808	0.4519	0.5223	0.5877	0.6514	0.7067	0.7578
	DaIB	0.1792	0.2924	0.3882	0.4729	0.5498	0.6182	0.6819	0.7361	0.7861
Multi10_2	aIB	0.1718	0.2835	0.3761	0.4569	0.5320	0.6019	0.6633	0.7236	0.7794
	DaIB	0.1694	0.2841	0.3826	0.4766	0.5519	0.6267	0.6902	0.7488	0.8069
Multi10_3	aIB	0.1664	0.2884	0.3852	0.4714	0.5390	0.6005	0.6619	0.7187	0.7696
	DaIB	0.1719	0.2837	0.3840	0.4615	0.5342	0.6017	0.6638	0.7242	0.7801

Table 3. The Micro-averaged precision on the nine data sets

	$P(T)$	DaIB	aIB	Improvement
Binary_1	0.880	0.840		0.04
Binary_2	0.882	0.598		0.284
Binary_3	0.932	0.850		0.082
Multi5_1	0.732	0.566		0.166
Multi5_2	0.760	0.638		0.122
Multi5_3	0.818	0.768		0.05
Multi10_1	0.514	0.424		0.09
Multi10_2	0.488	0.340		0.148
Multi10_3	0.474	0.388		0.086
Average	0.72	0.6013		0.1187

Table 4. Paired t-test for mutual information(MI) and Micro-averaged precision($P(T)$) of DaIB and aIB

Comparison	Paired-t statistics	MI	$P(T)$
DaIB vs. aIB	df	8	8
	t	5.9188	4.7688
	p-value	0.0004	0.0014

where the *ratio* also reaches the largest. The two-tailed, paired t-test with 95% confidence level shows significant statistical differences in mutual information and micro-averaged precision between DaIB and aIB in Table 4.

3.5 Experiment Results Analysis

By taking into account the information loss among the neighbors of an element, the DaIB algorithm outperforms the aIB algorithm on the following aspects:

1. The DaIB algorithm can preserve more mutual information than the aIB algorithm at almost all of the same cardinalities on the nine data sets. This is more evident as the value of $|T|$ decreases. On the Binary_2 data set, when $|T| = 2$, the mutual information in DaIB is 18% more than that in aIB.
2. The DaIB algorithm yields higher micro-averaged precision than the aIB algorithm on all the nine data sets. The biggest improvement 28.4% was achieved on the Binary_2 data set. The improvements on Multi5_1 and Multi10_2 data sets are 16.6% and 14.8%, respectively. The average of the micro-averaged precision of the DaIB algorithm on the nine data sets is 72%, which is higher than that of the aIB algorithm 60.13% by 11.87%.
3. The micro-averaged precisions of the aIB algorithm on Binary_1, Binary_2 and Binary_3 are 84%, 59.8% and 85% respectively. It is clear that the micro-averaged precision on the Binary_2 data set is much less than the ones on Binary_1 and Binary_3. The micro-averaged precisions of the DaIB algorithm on Binary_1, Binary_2 and Binary_3 are 88%, 88.2% and 93.2% respectively, which is more stable than the aIB algorithm. Moreover, the results also demonstrate more stable micro-averaged precision on the other six data sets.

4 Conclusion

The proposed DaIB algorithm alleviates the sub-optimality problem in the aIB algorithm by using the concept of density-based chains. Compared with the aIB algorithm, the proposed algorithm preserves more mutual information and achieves higher micro-averaged precision. Meanwhile, it outputs a pruned hierarchical clustering tree-structure, and produces a more stable result. The main contributions of this paper are as follows:

1. We introduce the density-based chain into the IB method, and propose the DaIB algorithm which considers not only the information loss when merging two neighboring components, but also the information losses when merging all components which are closely to each other.
2. The proposed DaIB algorithm successfully alleviate the sub-optimality problem in the aIB algorithm, and it can preserve more mutual information and achieve higher micro-averaged precision than the aIB algorithm. The paired t-test indicated that there are significant improvements on mutual information and micro-averaged precision statistically.

In our future work we plan to address the problems like how to automatically determine the best cardinality for aIB/DaIB algorithms, we optimistically expect that these further work will promise to be of assistance to scientists wishing to analysis high dimensional data sets.

References

1. Naftali Tishby, F.C.P., Bialek, W.: The information bottleneck method. on Communication and Computation. In: Proc. 37th Allerton Conference, pp. 368–377 (1999)
2. Slonim, N.: The Information Bottleneck: Theory and Applications. Ph.D thesis, the Senate of the Hebrew University (2002)
3. Slonim, N., Tishby, N.: Agglomerative information bottleneck. In: Advances in Neural Information Processing Systems (NIPS), vol. 12, pp. 617–623 (1999)
4. Noam Slonim, N.F., Tishby, N.: Unsupervised document classification using sequential information maximization on Research and Development in Information Retrieval. In: Proc. of the 25th Ann. Int. ACM SIGIR Conf., 129–136 (2002)
5. Jacob Goldberger, S.G., Greenspan, H.: Unsupervised image set clustering using an information theoretic framework. IEEE Transactions on Image Processing 15(2), 449–458 (2006)
6. Ester, M., Hans-Peter Kriegel, J.S., Xu., X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of 2nd international conference on knowledge discovery and data mining (KDD), pp. 226–231 (1996)
7. Slonim, N., Tishby, N.: Document clustering using word clusters via the information bottleneck method on Research and Development in Information Retrieval. In: Proc. of the 23rd Ann. Int. ACM SIGIR Conf., pp. 208–215 (2000)
8. Slonim, N., Tishby, N.: The power of word clusters for text classification. In: 23rd European Colloquium on Information Retrieval Research (ECIR) (2001)
9. Lang, K.: Learning to filter netnews. In: Proc. of the 12th International Conf. on Machine Learning, pp. 331–339 (1995)

State-Based Regression with Sensing and Knowledge

Richard Scherl¹, Cao Son Tran², and Chitta Baral³

¹ CS Department, Monmouth University, West Long Branch, NJ

rscherl@monmouth.edu

² CS Department, New Mexico State U., Las Cruses, NM

tson@cs.nmsu.edu

³ CS and Engineering, Arizona State U., Tempe, AZ

chitta@asu.edu

Abstract. This paper develops a state-based regression method for planning domains with sensing operators and a representation of the knowledge of the planning agent. The language includes primitive actions, sensing actions, and conditional plans. We prove the soundness and completeness of the regression formulation with respect to the definition of progression and the semantics of a propositional modal logic of knowledge. It is our expectation that this work will serve as the foundation for the extension of recently successful work on state-based regression planning to include sensing and knowledge as well.

Keywords: Regression, Plans, Knowledge, Sensing.

1 Introduction

Progression and regression are two important reasoning methods in reasoning about actions and change. Progression is defined for computing the possible states resulting from the execution of a plan from a given state. Formally, the progression function could be defined as a mapping

$$F : Actions \times States \longrightarrow 2^{States} \quad (1)$$

where *States* denotes the set of possible states of the world and *Actions* denotes the set of actions. This function is then extended to compute the result of the execution of a plan from a given state. Regression, on the other hand, is used to determine the possible states of the world, from which the execution of a given plan results in some states satisfying a predefined formula. Given the progression function F , mathematically, the regression function R should be defined by the inverse of F ; i.e., given a formula φ and an action a , the regression function R is defined by a mapping

$$R : Formulas \times Actions \longrightarrow Formulas \quad (2)$$

where *Formulas* is the set of formulas. Each formula (from the domain and the range of R) characterizes a set of states, such that

$$R(\phi, a) = \psi \text{ if and only if } \forall s_1, s_2. (s_1 \models \psi \wedge s_2 \in F(a, s_1) \rightarrow s_2 \models \phi)$$

where \models denotes the usual satisfiability relation between states and formulas. For action domains with sensing actions and incomplete information, this is the formula for regression adopted in [12,6].

By defining the regression function as the inverse of the progression function, we obtain a generic formalism that is ready for use in action domains in which the progression function is known. However, this definition is not *constructive*, i.e., to compute the result of regression on a formula, one might have to guess the answer and then verify it using the progression function. Given that the complexity of the problem of computing the result of regression is NP-complete [6], which holds even if actions are deterministic, this is not a surprise.

Despite the computational complexity of this problem, several regression formalisms for action theories with sensing actions and incomplete information have been investigated [12,5,6,7,8,10] and some have been successfully implemented [6,10]. Regression formalisms can be classified by their use of Equation (2) in their definition of regression. In some proposals, [12,6], the regression function is defined by Equation (2). We call these proposals *indirect definitions* of the regression function. In other formalisms, the regression function is defined *directly*, i.e., no reference to the progression function is used in its definition and Equation (2) serves as a means for the verification of its soundness and completeness [5,7,8,10].

Works that define a direct regression function follow two patterns: one defines the function on formulas and another defines the function on sets of states or formulas in conjunctive normal forms. The later, also referred to as *state-based regression*, has been implemented in a conditional planner [10], which is competitive against several other conditional planners [9]. This result is also in line with the success of regression planners in classical planning [2,3].

In this paper, we develop a direct regression function for domains that include sensing actions, knowledge, and conditional actions. The work developed here can be seen as an extension of the work of [10] to be sound and complete with respect to the full semantics of knowledge and actions with conditional effects. Needless to say, this completeness is obtained at the cost of greater complexity (as discussed from a theoretical perspective in [1]), but the greater expressivity is needed for many problems such as the illustrative example used in this paper.

It is our expectation that this work will serve as the foundation for the extension of the promising state-based regression planning methods [3,2] to domains that include sensing and a representation of the knowledge of the planning agent. The development of conditional planning algorithms based on the regression operator presented here is not discussed in this paper, but forms an important part of our future work in this area.

2 Language

A planning domain $\mathbf{D} = \langle \mathbf{F}, \mathbf{O}_{\text{ns}}, \mathbf{O}_{\text{se}}, \mathbf{A}, \mathbf{I}, \mathbf{G} \rangle$ consists of a finite set of propositional fluent symbols \mathbf{F} , a finite set of ordinary (nonsensing) action operators \mathbf{O}_{ns} , a finite set of sensing action operators \mathbf{O}_{se} , a representation \mathbf{A} of the preconditions and effects of these action operators, a specification of the initial state of the world \mathbf{I} , and a

specification of the goal state **G**. The propositional fluent symbols include the symbol \top that is true in all interpretations.

The representation of the initial state **I** consists of propositions of the form **initially** φ , where φ is an arbitrary propositional formula formed from **F**. For example: **initially** $\neg P_3$ and **initially** $P_1 \vee P_2$. The planner is given knowledge of this initial state of the world. Note that in the example to follow, the use of implication in something of the form $P_1 \rightarrow P_2$ is merely an abbreviation for $\neg P_1 \vee P_2$.

A goal **G** (e.g., $P_1 \wedge \neg P_2$) is always a conjunction of literals. The goal of the planner is to achieve knowledge of these literals and also know that they are achieved.

The specification of actions (non-sensing actions) **A**, indicates the effects (with conditions) and also executability preconditions of all actions in **O**. Both the effects, conditions, and executability conditions are restricted to be conjunctions of literals which we represent as sets of literals. Consider an arbitrary action ACT_1 :

Effect: $\{\{P_1^1, \dots, P_{n^1}^1\} \Rightarrow \{Q_1^1, \dots, Q_{m^1}^1\}, \dots \{P_1^j, \dots, P_{n^j}^j\} \Rightarrow \{Q_1^j, \dots, Q_{m^j}^j\}\}$
ExCond: $\{R_1, \dots, R_o\}$

The effect is a set of condition-effect pairs. The action ACT_1 is executable in a state as long as the conjunction of $\{R_1, \dots, R_o\}$ holds. For each i , the conjunction of the literals $\{Q_1^i, \dots, Q_{m^i}^i\}$ must hold in the successor state if the conjunction of $\{P_1^i, \dots, P_{n^i}^i\}$ holds in the state in which the action begins. It is required that the conditions of the various condition-action pairs be mutually exclusive. Therefore no more than one condition can hold in any single state. It is assumed that the language includes an action NOOP that has one condition-effect with the condition as \top , an empty consequent, and **ExCond** as \top .

Some notation is useful to talk about the specifications of actions. The function **excond**(a) returns the ExCond of action a . The function **effects**(a) returns a list (set) of pairs consisting of the antecedent and consequent of the conditional effect. Given such a pair e , the function **condition**(e) yields a list of literals that constitutes the antecedent or condition of effect e and the function **head**(e) yields the list of literals that constitutes the consequent of e . For a literal l , \bar{l} denotes its complementary literal; for a set of literals S , $\bar{S} = \{\bar{l} \mid l \in S\}$.

There are also sensing actions that determine the truth of a fluent:

$SENSE_1 : \text{Effect: } \{\} \text{ Determines: } P_1 \text{ ExCond: } \{R_1, \dots, R_o\}$

The sensing action $SENSE_1$ determines the truth of proposition P_1 and is executable if the conjunction $\{R_1, \dots, R_o\}$ is satisfied. Given a sensing action a , **determines**(a) returns the determines fluent of action a . The restriction to a single fluent is not more restrictive than a set of fluents since a sequence of sensing actions can be equivalent to a sensing action that determines a set (i.e., a conjunction) of literals. Note that since we require that the **Effect** component be empty, it is ensured that sensing actions have no effect on the world.

Plans are constructed out of a sequence of actions and the if/then constructs are called conditional plans. For simplicity of the presentation of the definitions in this paper, we do impose some restrictions on the form of conditional plans as indicated in the following definition:

Definition 1 (Conditional Plan). Let a be an action.

1. $[]$ is a conditional plan.
2. $a; c$ is a conditional plan if c is a conditional plan.
3. $[\text{if } f \text{ then } c_1 \text{ else } c_2]$ is a conditional plan if f is a fluent and c_1 and c_2 are conditional plans.

3 State Semantics

A *state* is a complete (i.e., for each fluent f either f or $\neg f$ is included) and consistent set of fluent literals (i.e., for each fluent f both f and $\neg f$ are not included). It is a propositional representation of the truth (falsity) of propositions in a particular possible world. A *knowledge set* (or k -set) is a set of states. A *combined structure* (or c -structure) is a pair $\langle s, \Sigma \rangle$ where Σ is a k -set and s is a state belonging to Σ . A *partial state* (or p -state) is a consistent set of fluent literals. A *partial structure* (or p -structure) is a pair $\langle \delta, \Delta \rangle$ where Δ is a set of p -states and δ is a p -state belonging to Δ . A p -structure $\gamma = \langle \delta, \Delta \rangle$ extends a p -structure $\gamma' = \langle \delta', \Delta' \rangle$, denoted by $\gamma' \sqsubseteq \gamma$, if (i) $\delta' \subseteq \delta$; (ii) for each $\lambda \in \Delta$ there exists some $\lambda' \in \Delta'$ such that $\lambda' \subseteq \lambda$; and (iii) for each $\lambda' \in \Delta'$ there exists some $\lambda \in \Delta$ such that $\lambda' \subseteq \lambda$.

For example, the following are states if we consider only the atoms F and G: $s_1 = \{F, G\}$, $s_2 = \{\neg F, G\}$, $s_3 = \{F, \neg G\}$, $s_4 = \{\neg F, \neg G\}$. If s is a state (or more generally a p -state), then $s \models l_1$, where l_1 is a literal, means that $l_1 \in s$. The definition of \models can be inductively generalized in the obvious way to $s \models \varphi$ where φ is an arbitrary formula or a set of literals representing a conjunction of literals. Knowledge sets are a representation of the knowledge (ignorance) of the planner. For example, if we consider only the atoms F and G, some possible knowledge sets are: $b_1 = \{s_1, s_2, s_3, s_4\}$, $b_2 = \{s_1, s_4\}$, and $b_3 = \{s_2, s_3\}$. If Σ is a knowledge set, i.e., a set of sets of literals, then $\Sigma \models \text{Knows}(l_1)$ means that $\forall s \in \Sigma, s \models l_1$. Otherwise $\Sigma \models \neg \text{Knows}(l_1)$. This definition of \models can be inductively generalized in the obvious way to define $\Sigma \models \text{Knows}(\varphi)$, where φ is an arbitrary formula. Given a c -structure $st = \langle s, \Sigma \rangle$, $st \models l_1$, if $l_1 \in s$. Additionally, $st \models \text{Knows}(l_1)$, if $\Sigma \models \text{Knows}(l_1)$. For the definition of the transition function (to be presented next), we need to introduce a structure \perp . For any expression Ψ , $\perp \not\models \Psi$. Finally, it is required that $s \in \Sigma$. We are therefore using the semantics S5 as the basis of our modal logic of knowledge and there is no need to allow nesting of modal operators.

Given a planning domain, we can build the initial states and c -structures.

Definition 2 (Initial state). A state s is an initial state of a planning domain \mathbf{D} if s satisfies I .

Definition 3 (Initial c -structure). A c -structure $\langle s, \Sigma_0 \rangle$ is an initial c -structure if every $u \in \Sigma_0$ is an initial state.

It is straight forward to develop an algorithm that converts the conjunction of the φ s from each **initially** φ statements into the set of possible initial states.

Some notation will be needed in the machinery to be developed. Two p -states δ and δ' are *compatible* with respect to a set of fluent literals S , denoted by $\delta \sim_S \delta'$ if $S \cap \delta = S \cap \delta'$.

4 Progression

In progression, a plan is executed starting on an initial structure. We need to specify a transition function $\hat{\Phi}$ from plans and structures into structures. This is based on the specification of a transition function Φ from actions and structures into structures.

Some notation needs to be initially defined so that the transition function can be specified. For a p-state δ and action a , a is executable in δ if $\text{excond}(a) \subseteq \delta$. The effect of a in δ is defined by

$$e_a(\delta) = \begin{cases} \text{head}(p) & p \in \text{effects}(a) \text{ and } \delta \models \text{condition}(p) \\ \emptyset & \neg \exists p \in \text{effects}(a) \text{ s.t. } \delta \models \text{condition}(p) \end{cases}$$

This holds because we assume that $\text{condition}(p)$ and $\text{condition}(p')$ are mutual exclusive for $p \neq p'$. If $\delta \models \text{condition}(p)$, we say that p is *applicable* in δ .

Definition 4 (Result). *The result of executing a non-sensing action a in δ is defined by*

$$Res(a, \delta) = \begin{cases} (\delta \setminus \overline{e_a(\delta)}) \cup e_a(\delta) & \text{if } \delta \models \text{excond}(a) \\ \perp & \text{otherwise} \end{cases}$$

The notation \perp is used to indicate that the execution of a in δ fails. The transition function over p-structures and actions is defined as follows.

Definition 5 (Transition Function). *For an action a and a p-structure $\langle \delta, \Delta \rangle$,*

- if $\delta \not\models \text{excond}(a)$ then $\Phi(a, \langle \delta, \Delta \rangle) = \perp$;
- if $\delta \models \text{excond}(a)$; and a is a non-sensing action then

$$\Phi(a, \langle \delta, \Delta \rangle) = \langle Res(a, \delta), \{Res(a, \delta') \mid \delta' \in \Delta, \delta' \models \text{excond}(a)\} \rangle$$

- if $\delta \models \text{excond}(a)$ and a is a sensing action which senses f (i.e. $\text{determines}(a) = \{f\}$)

$$\Phi(a, \langle \delta, \Delta \rangle) = \langle \delta, \{\delta' \mid \delta' \in \Delta, \delta' \sim_{\{f, \neg f\}} \delta, \text{ and } \delta' \models \text{excond}(a)\} \rangle$$

Following [8], the function Φ needs to be extended to progress conditional plans.

Definition 6 (Extended Transition Function). *Let c be a conditional plan and $\sigma = \langle \delta, \Delta \rangle$ be a p-structure.*

1. if $c = []$ then $\hat{\Phi}(c, \sigma) = \sigma$;
2. if $c = a; c_1$ where a is an action and c_1 is a conditional plan then $\hat{\Phi}(c, \sigma) = \hat{\Phi}(c_1, \Phi(a, \sigma))$;
3. if $c = [\text{if } f \text{ then } c_1 \text{ else } c_2]$ where c_1 and c_2 are conditional plans then

$$\hat{\Phi}(c, \sigma) = \begin{cases} \hat{\Phi}(c_1, \Phi(\sigma)) & \text{if } \Phi(\sigma) \models \text{Knows}(f) \\ \hat{\Phi}(c_2, \Phi(\sigma)) & \text{if } \Phi(\sigma) \models \text{Knows}(\neg f) \end{cases}$$

4. Otherwise \perp .

Given a planning domain $\mathbf{D} = \langle \mathbf{F}, \mathbf{O}_{ns}, \mathbf{O}_{se}, \mathbf{A}, \mathbf{I}, \mathbf{G} \rangle$ a progression solution is defined as follows:

Definition 7 (Progression Solution). A plan c is a progression solution for a planning domain \mathbf{D} if for every initial c -structure σ_0 of \mathbf{D} , $\hat{\Phi}(c, \sigma_0) \neq \perp$ and $\hat{\Phi}(c, \sigma_0) \models \text{Knows } (G)$.

Intuitively, the planner begins with some knowledge of the possible ways the world can be. The actual world could be any one of the possibilities. A plan is a solution, if no matter which of the possibilities is in fact the actual world, an execution of the plan in that world with the knowledge of the possible ways the world could be, yields knowledge of the goal.

5 Medical Example

A running medical example, similar to that used in other work in this area [11] is used to illustrate our approach. A patient is ill, but alive. We know that if he is infected with disease 123, then he is also hydrated. But we do not know whether he is infected. We do have a stain, which can be used to test for infection with disease 123. If the result of the staining is blue, then the patient is infected. We have a sensing action to determine whether or not the result of the stain action is blue or not. We can treat disease 123 with medication, but the problem is that if the patient is not hydrated, he will die from the medication. So, it is important to construct the appropriate plan so that the patient is guaranteed not to die.

In this domain (\mathbf{D}_1), \mathbf{O}_{ns} contains the following actions: MEDICATE, and STAIN. Additionally, \mathbf{O}_{se} contains INSPECT. The set \mathbf{F} contains DEAD, BLUE, INFECTED, and HYDRATED.

We have the following axiomatization of the initial state:

$$\text{initially } \neg\text{DEAD} \wedge \neg\text{BLUE} \quad \text{initially INFECTED} \rightarrow \text{HYDRATED}$$

and the goal is to have the patient not dead and not infected:

$$\text{Goal :} \neg\text{DEAD} \wedge \neg\text{INFECTED}$$

There are 16 possible states for \mathbf{D}_1 of which the following are initial states:

$$\begin{aligned} s_1 &= \{\neg\text{DEAD}, \neg\text{INFECTED}, \neg\text{BLUE}, \neg\text{HYDRATED}\} \\ s_2 &= \{\neg\text{DEAD}, \neg\text{INFECTED}, \neg\text{BLUE}, \text{HYDRATED}\} \\ s_3 &= \{\neg\text{DEAD}, \text{INFECTED}, \neg\text{BLUE}, \text{HYDRATED}\} \end{aligned}$$

Therefore, the knowledge set that we need to consider is $\{s_1, s_2, s_3\}$. The patient is not dead ($\neg\text{DEAD}$) and the stain is not blue ($\neg\text{BLUE}$) in all three states. In one state the patient is infected (INFECTED) and hydrated (HYDRATED), in another he is not infected and not hydrated, and in the other he is hydrated and not infected. So, we have the following three initial c -structures:

$$\langle s_1, \{s_1, s_2, s_3\} \rangle \quad \langle s_2, \{s_1, s_2, s_3\} \rangle \quad \langle s_3, \{s_1, s_2, s_3\} \rangle$$

Two of the initial states (s_1 and s_2) are already satisfying the goal. The third one (s_3) does not. Thus, we must perform actions that do not undo the goal requirements in the first two and changes the third one to a state that satisfies the goal. Additionally, in the resulting structures, the planning agent must know that the goal holds.

The axiomatization **A** of the actions are as follows:

- STAIN: **Effect** : $\{\{\text{INFECTED}\} \Rightarrow \{\text{BLUE}\}\}$
ExCond : $\neg\text{BLUE}$
- MEDICATE: **Effect** : $\{\{\text{HYDRATED}\} \Rightarrow \{\neg\text{INFECTED}\},$
 $\{\neg\text{HYDRATED}\} \Rightarrow \{\text{DEAD}\}\}$
ExCond : \top
- INSPECT: **Effect**: \emptyset **Determines**: BLUE **ExCond**: \top

A plan to accomplish the goal is as follows:

$$c = [\text{STAIN}; \text{INSPECT}; [\text{if } \text{BLUE} \text{ then } \text{MEDICATE} \text{ else } \text{NOOP}]]$$

The plan ensures that medicate is only applied in the correct state. We have that

$$\hat{\Phi}(c, \langle s_1, \{s_3, s_1, s_2\} \rangle) = \langle s_1, \{s_1, s_2\} \rangle \quad \hat{\Phi}(c, \langle s_2, \{s_3, s_1, s_2\} \rangle) = \langle s_2, \{s_1, s_2\} \rangle \\ \hat{\Phi}(c, \langle s_3, \{s_1, s_2, s_3\} \rangle) = \langle s_4, \{s_4\} \rangle$$

where $s_4 = \{\neg\text{DEAD}, \neg\text{INFECTED}, \text{BLUE}, \text{HYDRATED}\}$. Since the goal is satisfied by s_1 , s_2 , and s_4 , we conclude that c is a progression solution (plan) for the domain \mathbf{D}_1 .

6 Regression

Various formalisms on regression for reasoning about actions have been developed (e.g. [4,5,7,8]). Regression has also formed the basis for a number of regression-based planners (e.g. [2,10]). In these works, regression was designed to ignore actions that do not directly contribute to the current goal. In other words, they would consider only “useful” actions in the regression. But with these restricted forms of regression, there are plans found by progression based planners that can not be found by a regression based planner. This can be seen in the following example:

Example 1. Consider $\mathbf{D}_2 = \langle \{f, h\}, \{a, b\}, \emptyset, \mathbf{A}_2, \emptyset, \{h\} \rangle$, where \mathbf{A}_2 is defined by **effects**(a) = $\{f\} \Rightarrow \{h\}$ and **effects**(b) = $\{\neg f\} \Rightarrow \{h\}$. Also, **excond**(a) = **excond**(b) = \top . It is easy to see that both $[a; b]$ and $[b; a]$ are progression solutions for \mathbf{D}_2 .

Now let us try to find the plan $[b; a]$ by regression, following an adaptation of the formalism in [2] for \mathbf{D}_2 . Intuitively, we should start with the regression of a in $\{h\}$. This will result in $\{f\}$, i.e., to achieve h by means of a , we need to achieve f . As $\{f\}$ is not satisfied by the initial condition, another regression step is needed. Because f is not present in the head of any effect of a or b , neither a nor b will be considered as “useful” for the goal of achieving f . This implies that $[b; a]$ cannot be found by a regression formalism considering only “useful” actions in its reasoning. The same argument can be made for $[a; b]$. As such, *regression formalisms under the restriction of using only “useful” actions are generally incomplete* with respect to the complete semantics. \square

In this work, we define a regression formalism that is a truly reversal of the progression. Even though our later goal is to use this work in planners, our current purpose is

to establish the equivalence between progression and regression. We will present our formulation in a series of definitions. We start with the definition of the regression of a non-sensing action in a p-state, a set of p-states, and a p-structure. This is followed by the definition of the regression of a sensing action in a set of p-structures. Finally, we define the extended regression function, which allows for the regression of conditional plans and can be seen as the counter part to the extended transition function.

In defining the regression function, the first question we need to answer is “when can an action a be regressed in a p-state δ ?” Assume that a is a non-sensing action. Intuitively, the regression of a in δ should result in δ' such that $\delta \subseteq Res(a, \delta')$. From the definition of the function Res , we know that there are two possibilities: (i) there exists an effect p of a which is applicable in δ' ; or (ii) otherwise none is applicable. The first case implies that (a) there exists no literal in $\text{head}(p)$ such that its negation belongs to δ ; and (b) if a literal l belongs to $\text{excond}(a) \cup \text{condition}(p)$ (l is true in δ') and its negation \bar{l} belongs to δ (l is false in δ) then \bar{l} should belong to $\text{head}(p)$. In the second case, we have that $\text{excond}(a)$ must not be false in δ and for every effect p of a , its precondition, $\text{condition}(p)$, must be false in δ' . The following definition reflects these conditions (the cases (i) and (ii) correspond to Items 1 and 2, respectively.)

Definition 8 (Regressable). A non-sensing action a is regressable in a p-state δ if either 1 or 2 holds:

1. there exists some $p \in \text{effects}(a)$ such that

- $\text{head}(p) \cap \delta = \emptyset$,
- $\overline{\text{condition}(p)} \cap \delta \subseteq \text{head}(p)$, and
- $\text{excond}(a) \cap \delta \subseteq \text{head}(p)$.

We say that a is regressable via p in δ in this case.

2. $\text{excond}(a) \cap \delta = \emptyset$ and there exists a p-state δ' such that $\delta \cup \text{excond}(a) \subseteq \delta'$ and for every $p \in \text{effects}(a)$, $\text{condition}(p) \cap \delta' \neq \emptyset$.

A non-sensing action a is regressable in a p-structure $\langle \delta, \Delta \rangle$ if it is regressable in every p-state belonging to Δ .

The next step is to define the result of the regression of an action a in a p-state δ . Let us denote it with δ' . Clearly, we must have that a is executable in δ' . Furthermore, if an effect p of a is applicable in δ' then $\text{condition}(p)$ must be satisfied in δ' . In this case, $\text{head}(p)$ need not be present in δ' since it will be added to $Res(a, \delta')$ (Item 1, Def. 8). On the other hand, if none of the effects of a is applicable in δ' then for every $p \in \text{effects}(a)$, δ' should contain at least some elements in $\text{condition}(p)$ (Item 2, Def. 8). This leads to the following definition.

Definition 9 (Regression (non-sensing) in a p-State). Let a be a non-sensing action regressable in the p-state δ . Let

$$r_a^1(\delta) = \{Reg(a, p, \delta) \mid p \in \text{effects}(a) \text{ s.t. } a \text{ is regressable via } p \text{ in } \delta\}$$

where $Reg(a, p, \delta) = (\delta \setminus \text{head}(p)) \cup \text{condition}(p) \cup \text{excond}(a)$; and

$$r_a^2(\delta) = \left\{ \delta' \middle| \begin{array}{l} \exists \gamma. [\gamma \subseteq \bigcup_{p \in \text{effects}(a)} \overline{\text{condition}(p)}, \\ \delta' = \delta \cup \text{excond}(a) \cup \gamma, \\ \delta' \text{ is consistent,} \\ \forall p \in \text{effects}(a). [\delta' \cap \overline{\text{condition}(p)} \neq \emptyset]] \end{array} \right\}$$

We say that $r_a(\delta) = r_a^1(\delta) \cup r_a^2(\delta)$ is the set of p-states resulting from the regression of a in δ .

It is easy to see that a is regressable in δ if and only if $r_a(\delta) \neq \emptyset$.

Example 2. For domain \mathbf{D}_2 in Example 1 and $\delta = \{h\}$, we have that $r_a^1(\delta) = \{\{f\}\}$ and $r_a^2(\delta) = \{\{h, \neg f\}\}$. \square

It should be mentioned that $r_a(\delta)$ might contain some p-states which are superset of other elements in $r_a(\delta)$. Due to the fact that if an action a is regressable in a p-state δ then it is regressable in a p-state $\delta' \subseteq \delta$, the presence of such elements do not have any impact on the theoretical results presented in this paper. Eliminating this redundancy will be important in the development of a regression-based planner and will therefore be one of our future concerns. We now extend regression to a set of p-states.

Definition 10 (Regression (non-sensing) in a Set of p-States). Let Δ be a set of p-states and a be a non-sensing action regressable in every $\delta \in \Delta$. The regression of a in Δ , denoted by $r_a(\Delta)$, is given by

$$r_a(\Delta) = \{\Delta' \mid (\forall \delta' \in \Delta'. \exists \delta \in \Delta \text{ s.t. } \delta' \in r_a(\delta)) \text{ and } (\forall \delta \in \Delta. \exists \delta' \in \Delta' \text{ s.t. } \delta' \in r_a(\delta))\}$$

The first condition implies that each element in Δ' must be the result of the regression of some element in Δ and the second condition makes sure that nothing extra is introduced into Δ' . If a is not regressable in Δ , we write $r_a(\Delta) = \emptyset$. We are now ready to define the result of the regression of an action in a p-structure.

Definition 11 (Regression of Non-Sensing Actions). Let a be a non-sensing action regressable in the p-structure $\sigma = \langle \delta, \Delta \rangle$. We define

$$\mathcal{R}(a, \sigma) = \{\langle \delta', \Delta' \rangle \mid \delta' \in r_a(\delta), \Delta' \in r_a(\Delta), \delta' \in \Delta'\}.$$

Now it is necessary to define both regressability and regression for sensing actions. The main difference between non-sensing actions and sensing actions is that sensing actions do not change the world while non-sensing actions do. This leads to the following definition.

Definition 12 (Regressable for Sensing Actions). Let a be a sensing action which determines f . We say

- a is regressable in a p-state δ if $\overline{\text{excond}}(a) \cap \delta = \emptyset$ and $\{f, \neg f\} \cap \delta \neq \emptyset$.
- a is regressable in a set of p-states Δ if it is regressable in every $\delta \in \Delta$ and $\delta \sim_{\{f, \neg f\}} \delta'$ for every pair δ, δ' in Δ .
- a is regressable in a p-structure $\sigma = \langle \delta, \Delta \rangle$ if it is regressable in Δ .
- a is regressable in a set of p-structures Ω if it is regressable in every $\sigma \in \Omega$.

The first condition guarantees that δ can be extended to a p-state in which a is executable and the fluent that senses by a is known after its execution. The second condition ensures that if a is regressable in Δ then $\Delta \models \text{Knows}(f)$ or $\Delta \models \text{Knows}(\neg f)$. To continue we need the following notation:

Definition 13. A p-structure $\sigma = \langle \delta, \Delta \rangle$ agrees with a literal l (written as $\sigma \gg l$) if for every $\delta' \in \Delta$, either $l \in \delta'$ or $\bar{l} \notin \delta'$.

If $\sigma \gg l$, $\sigma + l$ denotes the p-structure $\langle \delta \cup \{l\}, \{\delta' \cup \{l\} \mid \delta' \in \Delta\} \rangle$; for a set of p-structures Ω , $\Omega + l = \{\sigma + l \mid \sigma \in \Omega\}$.

The above definition is extended to a set of p-structures Ω and a set of fluent literals S in an obvious way. Note that l is known to be true in the p-structure $\sigma + l$. To define the regression of sensing actions, observe that given a p-structure $\sigma = \langle \delta, \Delta \rangle$ and a sensing action a , the execution of a in σ will result in $\sigma' = \langle \delta, \Delta' \rangle$ with $\Delta' \subseteq \Delta$ and every $\delta' \in \Delta'$, $\delta' \sim_{\{f, \neg f\}} \delta$ where **determines** $(a) = f$. As such, instead of defining the regression of sensing actions in a single p-structure, we define the regression of sensing actions in a set of p-structures.

Intuitively, the result of the regression of a in a set of p-structures Ω should result in a set of p-structures Ω' such that (i) a is executable in every $\sigma' \in \Omega'$; (ii) for each $\sigma \in \Omega$, there exists some $\sigma' \in \Omega'$ such that $\sigma \sqsubseteq \Phi(a, \sigma')$; and (iii) any p-state belonging to a p-structure in Ω' must be an extension of a p-state belonging to some p-structure in Ω with **excond**(a) holding. (i) can be satisfied by adding **excond**(a) to every p-structure in Ω . (ii) and (iii) can be satisfied by creating for each p-structure $\langle \delta, \Delta' \rangle$ in Ω a p-structure $\langle \delta, \Delta \rangle$ where Δ' consists of Δ and all p-states (in other structures in Ω) representing possible world states which do not agree with δ on f . Formally, we define $\mathcal{R}(a, \Omega)$ as follows.

Definition 14 (Regression of Sensing Actions). Let a be a sensing action which senses f and Ω be a set of p-structures.

1. if a is not regressable in Ω then $\mathcal{R}(a, \Omega) = \emptyset$;
2. if a is regressable in Ω then

$$\begin{aligned} \mathcal{R}(a, \Omega) = & \{ \langle \delta, \Delta \rangle + \text{excond}(a) \mid \exists \langle \delta, \Delta' \rangle \in \Omega \text{ and} \\ & \Delta = \Delta' \cup \{ \gamma \mid \exists \langle \gamma, \Gamma \rangle \in \Omega \text{ s.t. } \delta \not\sim_{\{f, \neg f\}} \gamma \} \}. \end{aligned}$$

We can show that if a is regressable in Ω then $\mathcal{R}(a, \Omega) \neq \emptyset$. We next extend \mathcal{R} to define the regression $\widehat{\mathcal{R}}$ on conditional plans.

Definition 15 (Extended Regression Function). Let Ω be a set of p-structures and c be a conditional plan c . We define $\mathcal{R}(c, \emptyset) = \emptyset$ for every c and extend the \mathcal{R} function to $\widehat{\mathcal{R}}$ as follows:

1. For $c = []$, $\widehat{\mathcal{R}}([], \Omega) = \Omega$.
2. For $c = a; c'$ where c' is a conditional plan and a is a non-sensing action, $\widehat{\mathcal{R}}(c, \Omega) = \bigcup_{\sigma \in \widehat{\mathcal{R}}(c', \Omega)} \mathcal{R}(a, \sigma)$.
3. For $c = a; c'$ where c' is a conditional plan and a is a sensing action, $\widehat{\mathcal{R}}(c, \Omega) = \mathcal{R}(a, \widehat{\mathcal{R}}(c', \Omega))$.
4. For $c = [\text{if } f \text{ then } c_1 \text{ else } c_2]$ where c_1 and c_2 are two conditional plans, $R_1 = \widehat{\mathcal{R}}(c_1, \Omega)$ and $R_2 = \widehat{\mathcal{R}}(c_2, \Omega)$,
 - (a) if $R_1 = \emptyset$, $R_2 = \emptyset$, or $\sigma \not\gg f$ for some $\sigma \in R_1$ or $\sigma \not\gg \bar{f}$ for some $\sigma \in R_2$ then $\widehat{\mathcal{R}}(c, \Omega) = \emptyset$
 - (b) otherwise, $\widehat{\mathcal{R}}(c, \Omega) = (R_1 + f) \cup (R_2 + \bar{f})$.

We are now ready to define the notion of regression solution.

Definition 16 (Regression Solution). Given a planning problem $\mathbf{D} = \langle \mathbf{F}, \mathbf{O}_{\text{ns}}, \mathbf{O}_{\text{se}}, \mathbf{A}, \mathbf{I}, \mathbf{G} \rangle$, let $\Omega = \{\langle \mathbf{G}, \{\mathbf{G}\} \rangle\}$. A conditional plan c is a regression solution for \mathbf{D} if (i) $\widehat{\mathcal{R}}(c, \Omega) \neq \emptyset$ and (ii) for every initial c-structure σ , there exists some p-structure σ' in $\widehat{\mathcal{R}}(c, \Omega)$ such that $\sigma' \sqsubseteq \sigma$.

Example 3. To illustrate the above definition, continue with Example 2, let $\sigma_G = \langle \{h\}, \{\{h\}\} \rangle$, and $\Omega = \{\sigma_G\}$. Let $\delta_1 = \{f\}$ and $\delta_2 = \{\neg f, h\}$. We have that $r_a(\{h\}) = \{\delta_1, \delta_2\}$. It is easy to verify that $r_b(\delta_1) = \{f\}$ and $r_b(\delta_2) = \{\neg f\}$. This allows us to conclude that $\widehat{\mathcal{R}}([b; a], \Omega) \neq \emptyset$ and for each initial c-structure σ_0 there exists some $\gamma \in \widehat{\mathcal{R}}([b; a], \Omega)$ such that $\gamma \sqsubseteq \sigma_0$, i.e., $[b; a]$ is indeed a regression solution for \mathbf{D}_2 . \square

The following two theorems are our main theoretical results:

Theorem 1 (Soundness). Every regression solution of a planning problem \mathbf{D} is a progression solution of \mathbf{D} .

The proof of this theorem relies on the following properties: (i) for a non-sensing action a and a p-state δ , if $r_a(\delta) \neq \emptyset$ then for each $\delta' \in r_a(\delta)$, $\delta \sqsubseteq \text{Res}(a, \delta')$; (ii) this result can be extended to a regression solution as follows: if c is a conditional plan and Ω is a set of p-structures such that $\widehat{\mathcal{R}}(c, \Omega) \neq \emptyset$ then for every $\sigma \in \widehat{\mathcal{R}}(c, \Omega)$, $\widehat{\Phi}(c, \sigma) \neq \perp$ and there exists a $\sigma' \in \Omega$ such that $\sigma' \sqsubseteq \widehat{\Phi}(c, \sigma)$.

Theorem 2 (Completeness). Every progression solution of a planning problem \mathbf{D} is a regression solution of \mathbf{D} .

Given the planning domain \mathbf{D} and a progression solution c of \mathbf{D} , and $\Omega = \{\langle \mathbf{G}, \{\mathbf{G}\} \rangle\}$, the proof of this theorem is divided into two steps: (i) $\widehat{\mathcal{R}}(c, \Omega) \neq \emptyset$; and (ii) for each initial c-structure σ there exists $\gamma \in \widehat{\mathcal{R}}(c, \Omega)$ such that $\gamma \sqsubseteq \sigma$.

The full version of this paper contains proofs of both theorems. We conclude with the continuation of our running example.

Example 4. Consider the medical domain \mathbf{D}_1 , let

$$\begin{aligned} \sigma_G &= \langle \{\neg \text{DEAD}, \neg \text{INFECTED}\}, \{\{\neg \text{DEAD}, \neg \text{INFECTED}\}\} \rangle, \\ c &= \text{STAIN}; \text{INSPECT}; [\text{if BLUE then MEDICATE else NOOP}]. \end{aligned}$$

For $\Omega = \{\sigma_G, \{\sigma_G\}\}$, we want to compute: $\widehat{\mathcal{R}}(c, \Omega)$ (3)

Let $\delta_1 = \{\neg \text{DEAD}, \text{HYDRATED}, \text{BLUE}\}$, $\delta_2 = \{\neg \text{DEAD}, \neg \text{INFECTED}, \neg \text{BLUE}\}$, and

$$\delta_3 = \{\neg \text{DEAD}, \text{INFECTED}, \text{HYDRATED}, \neg \text{BLUE}\}.$$

The first step creates the following formula to be regressed further:

$$\widehat{\mathcal{R}}([\text{STAIN}; \text{INSPECT}], \{\langle \delta_1, \{\delta_1\} \rangle, \langle \delta_2, \{\delta_2\} \rangle\}) \quad (4)$$

Sentence (4) regresses to (5).

$$\widehat{\mathcal{R}}([\text{STAIN}], \{\langle \delta_1, \{\delta_1, \delta_2\} \rangle, \langle \delta_2, \{\delta_2, \delta_1\} \rangle\}) \quad (5)$$

Finally, sentences (5) regresses to:

$$\widehat{\mathcal{R}}(\[], \{\langle \delta_3, \{\delta_3, \delta_2\} \rangle, \langle \delta_2, \{\delta_3, \delta_2\} \rangle\}) \quad (6)$$

We now have the following two structures for testing whether we have a regression solution:

$$\omega_1 = \langle \delta_3, \{\delta_3, \delta_2\} \rangle \text{ and } \omega_2 = \langle \delta_2, \{\delta_2, \delta_3\} \rangle.$$

Since $\omega_1 \sqsubseteq \langle s_3, \{s_3, s_1, s_2\} \rangle$, $\omega_2 \sqsubseteq \langle s_2, \{s_3, s_1, s_2\} \rangle$, and $\omega_2 \sqsubseteq \langle s_1, \{s_3, s_1, s_2\} \rangle$, c is a regression solution for \mathbf{D} . \square

7 Summary and Future Work

This paper develops a constructive state-based regression method for planning domains with sensing operators and a full representation of the knowledge of the planning agent. The language includes primitive actions, sensing actions, and conditional plans. We prove the soundness and completeness of the regression formulation with respect to the definition of progression and the semantics of a modal logic of knowledge. Space does not permit detailed comparison with the various approaches found in the literature in this area. This work can serve as a basis for future work on regression based planning. It is in this context where comparisons with related approaches will be most instructive.

Acknowledgements

Both Chitta Baral and Richard Scherl acknowledge support from DTO's AQUAINT program under award number N61339-06-C-0143. Richard acknowledges additional support from the Knowledge Fusion Center of the Army Research Laboratory under contract number DAAD-03-2-0034. Chitta acknowledges additional support from NSF under grant number 0412000 and from ONR-MURI number N00014-07-1-1049. Son Tran acknowledges support from NSF grants EIA-0220590 and CNS-0454066.

References

1. Baral, C., Kreinovich, V., Trejo, R.: Computational complexity of planning and approximate planning in the presence of incompleteness. *Artificial Intelligence* 122(1-2), 241–267 (2000)
2. Bonet, B., Geffner, H.: Planning as heuristic search. *Artificial Intelligence* 129, 5–33 (2001)
3. Nguyen, X.L., Kambhampati, S., Nigenda, R.S.: Planning graph as the basis for deriving heuristics for plan synthesis by state space and CSP search. *Artificial Intelligence* 135, 73–123 (2002)
4. Pednault, E.: Toward a Mathematical Theory of Plan Synthesis. Ph.D thesis, Stanford University (1986)
5. Reiter, R.: *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press, Cambridge (2001)
6. Rintanen, J.: Conditional Planning in the discrete belief space. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence, pp. 1260–1265 (2005)
7. Scherl, R., Levesque, H.: Knowledge, action, and the frame problem. *Artificial Intelligence* 144, 1–39 (2003)

8. Son, T.C., Baral, C.: Formalizing sensing actions – a transition function based approach. *Artificial Intelligence* 125, 19–91 (2001)
9. Tuan, L.: Regression in the presence of incomplete information and sensing actions, and its application to conditional planning. Ph.D thesis, Arizona State University (2004)
10. Tuan, L.C., Baral, C., Son, T.C.: A state-based regression formulation for domains with sensing actions and incomplete information. *Logical methods in Computer Science* 2(4) (2006)
11. Weld, D., Anderson, C., Smith, D.: Extending graphplan to handle uncertainty & sensing actions. In: *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (1998)
12. Herzig, A., Lang, J., Marquis, P.: Action representation and partially observable planning in epistemic logic. In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence* (2003)

Some Results on the Completeness of Approximation Based Reasoning

Cao Son Tran and Enrico Pontelli

Knowledge representation, Logic, and Advanced Programming Laboratory
Computer Science Department, New Mexico State University
MSC CS, P.O.Box 30001, Las Cruces, NM 88003, USA
`{tson, epontell}@cs.nmsu.edu`

Abstract. We present two results relating the completeness condition of the 0-approximation for two formalisms: the action description language \mathcal{A} and the situation calculus. The first result suggests that the condition for the situation calculus formalism implies the condition for the action language formalism. The second result indicates that an action theory in \mathcal{A} can sometimes be simplified to an equivalent action theory whose completeness condition is weaker than the original theory for certain queries.

1 Introduction

Intelligent agents need to be able to reason about the effects of their actions—i.e., reason about actions and change (RAC)—and to make decisions based on this reasoning. Several agent architectures have been developed building on this perspective, e.g., the architecture proposed in [2] and further developed in [1]. One of the most important problems in reasoning about actions and change is to determine whether a property, typically described by a fluent formula φ , is true after the execution of an action sequence α from the initial state δ . We typically denote this using the query φ **after** α (a.k.a. *hypothetical reasoning*). The majority of the formalisms for RAC solve this problem by defining an entailment relation, denoted by \models , between action theories and queries (see, e.g., [4]). We write $(\mathcal{D}, \delta) \models \varphi$ **after** α to denote the fact that the action theory (\mathcal{D}, δ) entails φ **after** α , i.e., φ is true after the execution of α from the initial state, where \mathcal{D} represents the action domain and δ represents the initial state. Observe that the majority of the approaches to RAC originally define \models assuming that δ is a *complete* description about the initial state.

The *possible world semantics* can be employed to reason about effects of actions in presence of incomplete information [11]. In this approach, a fluent formula φ is true after the execution of an action sequence α iff it is true after the execution of α in *every* possible initial state of the world. Roughly, an entailment relation \models^P is defined by adapting \models to deal with incomplete information. It states that $(\mathcal{D}, \delta) \models^P \varphi$ **after** α iff, for every possible completion δ' of δ , $(\mathcal{D}, \delta') \models \varphi$ **after** α —where δ' is considered a completion of δ if it contains δ and it is a complete description of the initial state.

One main disadvantage of the possible world semantics is its high complexity. For example, [3] showed that, even for deterministic action theories, determining whether a

fluent is true or false after the execution of a single action is co-NP complete. Moreover, the presence of incomplete information makes the planning problem—another important problem in RAC—computationally harder (see, e.g., [3]).

An alternative to the possible world semantics is the reasoning based on approximations [13]. Instead of considering all possible states, approximations define what will *definitely* be true or false after the execution of an action. This approach reduces the complexity of the hypothetical reasoning but is in general incomplete. This stipulates the research in [7,14,15] to search for conditions under which the approximation is complete. While the approach in [7] addresses the question “When does the reasoning based on approximation coincide with the possible world semantics?”, the approach in [14] focuses on answering the question “When does the reasoning based on approximation for a particular fluent formula φ coincide with the possible world semantics?”.

In this paper, we investigate the relationship between these two completeness conditions. We will start by reviewing, in the next section, some definitions relevant to the completeness conditions in [7,14]. We will then discuss the relationship between these conditions and develop a transformation for simplification of action theories. We will finally present a result that directly relates the two conditions.

2 Approximation Based Reasoning

In this section, we review the basic definitions of the situation calculus language, the action language \mathcal{A} , and the 0-approximation in both formalisms.

2.1 Situation Calculus

Situation calculus has been introduced by McCarthy [8] and further developed in [9], and it is probably the oldest formalism for representing and reasoning about actions. In situation calculus, actions and their effects are encoded directly into a first order theory. The basic components of the situation calculus language, in the notation of Reiter [12], include a special constant S_0 denoting the initial situation, a binary function symbol Do , where $Do(a, s)$ denotes the successor situation to s resulting from executing the action a , fluent relations of the form $F(s)$ (or $F(\mathbf{x}, s)$), denoting that the fluent F (resp. $F(\mathbf{x})$) is true in the situation s , and a special predicate $Poss(a, s)$ (resp. $Poss(a(\mathbf{x}), s)$) denoting that action a (resp. $a(\mathbf{x})$) is executable in situation s .¹

A dynamic domain can be represented by a theory \mathcal{D} comprising of (i) axioms describing the initial situation S_0 ; (ii) action precondition axioms (one for each primitive action A), characterizing $Poss(A, s)$; (iii) successor state axioms (one for each fluent F), stating under what condition $F(Do(a, s))$ holds, as a function of what holds in s ; (iv) unique names axioms for the primitive actions; and some foundational, domain independent axioms. In particular, each domain \mathcal{D} is given by a set of axioms

$$\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{una}$$

where \mathcal{D}_0 , \mathcal{D}_{ap} , \mathcal{D}_{ss} , and \mathcal{D}_{una} encode the axioms about initial situation, the action preconditions, the successor state axioms, and the unique name axioms, respectively.

¹ For simplicity, we omit the parameters of actions and fluents.

Each axiom in \mathcal{D}_{ap} is in the form $Poss(a, s) \equiv \Pi_a[s]$ and each axiom in \mathcal{D}_{ss} is of the form $F(Do(a, s)) \equiv \gamma_F^+(a, s) \vee (F(s) \wedge \neg\gamma_F^-(a, s))$. We illustrate this in the next example.

Example 1. Let us consider the well-known bomb in the toilet example in [10] assuming that we do not have any knowledge about the initial situation. In this domain, we have two actions *Dunk* and *Flush* and two fluents *Clogged* and *Armed*. Dunking a packet into the toilet disarms the bomb but causes the toilet to be clogged. Flushing the toilet makes it unclogged.

The basic action theory for this domain is given next²

$$\mathcal{D}^b = \mathcal{D}_0^b \cup \mathcal{D}_{ap}^b \cup \mathcal{D}_{ss}^b \cup \mathcal{D}_{una}^b$$

- The action precondition axioms (\mathcal{D}_{ap}^b) are
 - $Poss(Flush, s) \equiv \top$ (i.e., $\Pi_{Flush} = \top$).
 - $Poss(Dunk, s) \equiv \neg Clogged(s)$ (i.e., $\Pi_{Dunk} = \neg Clogged$).
- Note that Π_A is a formula in the language \mathcal{L} , whose propositions are the fluents in \mathcal{D} , where A is either *Flush* or *Dunk*.
- The successor state axioms (\mathcal{D}_{ss}^b) for the fluents are:
 - $Clogged(Do(a, s)) \equiv (a = Dunk) \vee (Clogged(s) \wedge \neg(a = Flush))$
Here, $\gamma_{Clogged}^+(a) = (a = Dunk)$ and $\gamma_{Clogged}^-(a) = (a = Flush)$.
 - $Armed(Do(a, s)) \equiv \perp \vee (Armed(s) \wedge \neg(a = Dunk))$
Here, $\gamma_{Armed}^+(a) = \perp$ and $\gamma_{Armed}^-(a) = (a = Dunk)$.
- \mathcal{D}_{una}^b contains the following axiom: $Dunk \neq Flush$. This is because we only have 0-ary actions.
- \mathcal{D}_0^b is empty. □

In the situation calculus, to determine whether φ is true after the execution of the action sequence α , we determine whether $\mathcal{D} \models \varphi(Do(\alpha, S_0)) \wedge Poss(\alpha, S_0)$ where, for an action a and action sequence α , $Do([a, \alpha], s)$ stands for $Do(\alpha, Do(a, s))$, $Poss([a, \alpha], s)$ is the shorthand for $Poss(a, s) \wedge Poss(\alpha, Do(a, s))$, and \models is the logical entailment relation in first order logic.³ In this way, the possible world semantics is naturally employed as S_0 can be incomplete. In fact, we can easily check that

$$\mathcal{D}^b \models Poss([Flush, Dunk], S_0) \wedge \neg Armed(Do([Flush, Dunk], S_0)). \quad (1)$$

As we have mentioned, the progression problem becomes computationally harder (assuming that $NP \neq P$) in the presence of incomplete information about the initial situation. This has motivated Liu and Levesque [7] to explore the use of *approximations* for the progression task, and develop conditions under which the reasoning based on the approximation is complete. Their formulation is inspired by the reasoning algorithm developed for proper knowledge bases [6] and is restricted to local effect theories. Formally, this construction proceeds as follows.⁴

² We use \top and \perp to denote true and false respectively.

³ Strictly speaking, we need to add foundational axioms to \mathcal{D} .

⁴ For simplicity, we assume a propositional language.

- Situations are characterized in terms of *proper* knowledge bases (*proper KBs*), i.e., theories Σ which are consistent (w.r.t. the axioms of equality), and where all formulae can be expressed in the form $\forall(e \supset \ell)$, where e is a quantifier free formula containing only equalities and ℓ is a literal. We denote with ξ_ℓ the maximal disjunction of ground instances of the literal ℓ such that $\Sigma \models \xi_\ell$.
- A proper KB Σ is *complete* w.r.t. a fluent F if either $\Sigma \models F$ or $\Sigma \models \neg F$; Σ is *context-complete* w.r.t. \mathcal{D} if it is complete w.r.t. each F appearing in any γ_G^+ or γ_F^- .
- The theory \mathcal{D} is assumed to be *local effect*, i.e., γ_F^+ and γ_F^- are finite disjunctions of formulae of the form $(a = A \wedge \varphi)$, where A is a ground action, and the various φ are ground formulae. Given a ground action A and a ground fluent F , we will denote with F_A^+ (resp. F_A^-) the formula $\bigvee\{\varphi \mid (a = A \wedge \varphi) \text{ appears in } \gamma_F^+\}$ (resp. $\bigvee\{\varphi \mid (a = A \wedge \varphi) \text{ appears in } \gamma_F^-\}$).

Evaluation of formulae φ w.r.t. a proper KB Σ is based on a 3-value interpretation function $V(\Sigma, \varphi)$, which is sound, and can be proved complete when the formula meets certain criteria (\mathcal{NF} normal form).

Liu and Levesque provide a definition of progression which preserves the *proper* property of the encoding of situations. In presence of a finite domain of constants, progression of a proper KB Σ w.r.t. a ground action A ($\mathcal{P}_A(\Sigma)$) is defined as the sentences (for each ground fluent F):

$$\text{def_true}_F \vee (\xi_F \wedge \neg \text{poss_false}_F) \supset F \quad \text{def_false}_F \vee (\xi_{\neg F} \wedge \neg \text{poss_true}_F) \supset \neg F$$

where

$$\begin{aligned} \text{def_true}_F &= \begin{cases} \top & V(\Sigma, F_A^+) = 1 \\ \perp & \text{o.w.} \end{cases} & \text{poss_true}_F &= \begin{cases} \top & V(\Sigma, F_A^+) \neq 0 \\ \perp & \text{o.w.} \end{cases} \\ \text{def_false}_F &= \begin{cases} \top & V(\Sigma, F_A^-) = 1 \wedge V(\Sigma, F_A^+) = 0 \\ \perp & \text{o.w.} \end{cases} & \text{poss_false}_F &= \begin{cases} \top & V(\Sigma, F_A^-) \neq 0 \\ \perp & \text{o.w.} \end{cases} \end{aligned}$$

The notion can be generalized to sequences of actions $\alpha = [A_1, \dots, A_n]$, as $\mathcal{P}_\alpha = \mathcal{P}_{A_n} \circ \dots \circ \mathcal{P}_{A_1}$.⁵ If Σ_0 is context complete and \mathcal{D} is local effect then $\mathcal{P}_A(\Sigma_0)$ can be shown to be a classical progression.

Example 2. For the theory \mathcal{D}^b , the language of the knowledge base Σ_0 consists of two predicates *Armed* and *Clogged* and there is no constant or function symbol in this language. The formulae expressing progression for the theory \mathcal{D}^b are given below:

- for the action *Dunk* we have

$$\begin{array}{ll} \perp \vee (\xi_{\text{Armed}} \wedge \neg \top) \supset \text{Armed} & \top \vee (\xi_{\neg \text{Armed}} \wedge \neg \perp) \supset \neg \text{Armed} \\ \top \vee (\xi_{\text{Clogged}} \wedge \neg \perp) \supset \text{Clogged} & \perp \vee (\xi_{\neg \text{Clogged}} \wedge \neg \top) \supset \neg \text{Clogged} \end{array}$$

This simplifies to $\top \supset \neg \text{Armed}$ and $\top \supset \neg \text{Clogged}$.

- for the action *Flush* we have

$$\begin{array}{ll} \perp \vee (\xi_{\text{Armed}} \wedge \neg \perp) \supset \text{Armed} & \perp \vee (\xi_{\neg \text{Armed}} \wedge \neg \perp) \supset \neg \text{Armed} \\ \perp \vee (\xi_{\text{Clogged}} \wedge \neg \top) \supset \text{Clogged} & \top \vee (\xi_{\neg \text{Clogged}} \wedge \neg \perp) \supset \neg \text{Clogged} \end{array}$$

So, we have $\xi_{\text{Armed}} \supset \neg \text{Armed}$ and $\top \supset \neg \text{Clogged}$.

⁵ \circ denotes function composition, where $(f \circ g)(x) = f(g(x))$.

This computation allows us to conclude that the approximation based reasoning in \mathcal{D}^b will yield the same conclusion as in Eq. 1. Observe that this can also be concluded thanks to the fact that \mathcal{D}^b is *context-complete*. \square

2.2 Language \mathcal{A} and the 0-Approximation Semantics

Gelfond and Lifschitz [4] introduced the action language \mathcal{A} for representing actions and reasoning about their effects. In this approach, dynamic domains are represented by action descriptions, whose semantics is defined by a transition function that maps an action and a state to a new state.

The alphabet of a domain consists of a set \mathbf{A} of action names and a set \mathbf{F} of fluent names. A (fluent) literal l is either a fluent $f \in \mathbf{F}$ or its negation $\neg f$. Fluent literals of the forms f and $\neg f$ are said to be complementary. With \mathbf{L} we denote the set of all fluent literals, i.e., $\mathbf{L} = \{f, \neg f \mid f \in \mathbf{F}\}$. A *fluent formula* is a formula constructed from fluent literals using the connectives \wedge , \vee , and \neg . A *domain description* \mathcal{D} is a set of statements of the following forms:

$$a \text{ causes } l \text{ if } \psi \tag{2}$$

$$\text{executable } a \text{ if } \psi \tag{3}$$

where $a \in \mathbf{A}$ is an action, l is a fluent literal, and ψ is a set of fluent literals. (2) is called a *dynamic law*, describing the effect of action a . It says that if a is performed in a state where ψ holds, then l will hold in the successor state. (3) is an *executability condition* for a , stating that a is executable in any state in which ψ holds.

Given a domain description \mathcal{D} , for a fluent literal l , we denote with $\neg l$ its complementary literal. For a set of fluent literals σ , we denote with $\neg\sigma$ the set $\{\neg l \mid l \in \sigma\}$. A set of fluent literals σ is consistent if for every fluent f , either f or $\neg f$ does not belong to σ . We will use the two terms *consistent set of fluent literals* and *partial state* interchangeably. A set of fluent literals σ is complete if for every fluent f , either f or $\neg f$ belongs to σ . When σ is consistent and complete, it is called a *state*. A state s containing a partial state δ is called a *completion* of δ . For a partial state δ , we denote by $\text{ext}(\delta)$ the set of all completions of δ . For a set of partial states Δ , we denote by $\text{ext}(\Delta)$ the set of states $\cup_{\delta \in \Delta} \text{ext}(\delta)$.

A fluent literal l (resp. set of fluent literals γ) *holds* in a consistent set of fluent literals σ if $l \in \sigma$ (resp. $\gamma \subseteq \sigma$); l (resp. γ) *possibly holds* in σ if $\neg l \notin \sigma$ (resp. $\neg\gamma \cap \sigma = \emptyset$). The value of a formula φ in σ may be either true, false, or unknown, and it is defined as usual. It is easy to see that if σ is a state then for every formula φ , the value of φ is known (to be either true or false) in σ . From now on, to avoid confusion, we will use letters (possibly indexed) σ , δ , and s to denote a set of fluent literals, a partial state, and a state respectively.

An \mathcal{A} action theory is a pair (\mathcal{D}, Δ) where \mathcal{D} is a domain description and Δ is a set of partial states.

Example 3. The bomb in the toilet example is represented by $\Delta_1 = \emptyset$ and the action theory

$$\mathcal{D}_1 = \left\{ \begin{array}{ll} \text{Dunk causes } \neg \text{Armed} & \text{Dunk causes Clogged} \\ \text{Flush causes } \neg \text{Clogged} & \text{executable Flush if } \top \\ \text{executable Dunk if } \neg \text{Clogged} & \end{array} \right. \quad \square$$

The 0-approximation semantics has been originally introduced in [13]. Let \mathcal{D} be a domain description. An action a is *executable* in a partial state δ if there exists an executability condition

$$\text{executable } a \text{ if } \psi$$

in \mathcal{D} such that ψ holds in δ .

For an action a and a partial state δ s.t. a is executable in δ , the set of *effects* of a in δ , denoted by $e(a, \delta)$, and the set of *possible effects* of a in δ , denoted by $pe(a, \delta)$, are:

$$\begin{aligned} e(a, \delta) &= \{l \mid \text{there exists } [a \text{ causes } l \text{ if } \psi] \text{ in } \mathcal{D} \text{ s.t. } \psi \text{ holds in } \delta\} \\ pe(a, \delta) &= \{l \mid \text{there exists } [a \text{ causes } l \text{ if } \psi] \text{ in } \mathcal{D} \text{ s.t. } \psi \text{ possibly holds in } \delta\} \end{aligned}$$

Intuitively, $e(a, \delta)$ and $pe(a, \delta)$ are the sets of literals that *certainly hold* and *may hold*, respectively, in the successor state of every state $s \in ext(\delta)$. The *successor partial state* of a state s after the execution of an action a , denoted by $\Phi^0(a, \delta)$, is defined as follows.

Definition 1. For any action a and partial state δ ,

1. if a is not executable in δ then $\Phi^0(a, \delta) = \perp$;
2. otherwise, $\Phi^0(a, \delta) = e(a, \delta) \cup (\delta \setminus \neg pe(a, \delta))$.

The *final partial state* of a partial state δ after the execution of a sequence of actions α , denoted by $\widehat{\Phi}^0(\alpha, \delta)$, is defined as follows.

Definition 2. For any sequence of actions $\alpha = [a, \beta]$ and partial state δ ,

1. $\widehat{\Phi}^0(\[], \delta) = \delta$ and if $\beta = []$ then $\widehat{\Phi}^0(\alpha, \delta) = \Phi^0(a, \delta)$;
2. otherwise, $\widehat{\Phi}^0(\alpha, \delta) = \widehat{\Phi}^0(\beta, \Phi^0(a, \delta))$.

Given the extended transition function $\widehat{\Phi}^0$, the entailment relation between an action theory and a query with respect to the 0-approximation semantics, denoted by \models^0 , is defined as follows (recall that Δ is a set of partial states).

Definition 3. An action theory (\mathcal{D}, Δ) entails a query $[\varphi \text{ after } \alpha]$ with respect to the 0-approximation semantics, denoted by $(\mathcal{D}, \Delta) \models^0 \varphi \text{ after } \alpha$, if for every $\delta \in \Delta$, $\widehat{\Phi}^0(\alpha, \delta) \neq \perp$ and φ is true in $\widehat{\Phi}^0(\alpha, \delta)$.

It should be noted that the transition function $\Phi_0(a, \delta)$ coincides with the transition function defined for complete action theories in [4] if δ is complete. As such, the possible world semantics for an action theory (\mathcal{D}, Δ) can be characterized by $\widehat{\Phi}^0$ of the theory $(\mathcal{D}, ext(\Delta))$. For convenience of our discussion, we say $(\mathcal{D}, \Delta) \models^P \varphi \text{ after } \alpha$ iff $(\mathcal{D}, ext(\Delta)) \models^0 \varphi \text{ after } \alpha$. The following example demonstrates the use of the 0-entailment in reasoning about the effects of actions.

Example 4. Consider the action theory $(\mathcal{D}_1, \Delta_1)$ from Example 3. We have $e(Flush, \emptyset) = pe(Flush, \emptyset) = \{\neg Clogged\}$. Hence,

$$\Phi^0(Flush, \emptyset) = e(Flush, \emptyset) \cup (\emptyset \setminus \neg pe(Flush, \emptyset)) = \{\neg Clogged\} = \delta_1.$$

Furthermore, we have $e(Dunk, \delta_1) = \{Clogged, \neg Armed\}$ and $pe(Dunk, \delta_1) = \{Clogged, \neg Armed\}$. Thus,

$$\Phi^0(Dunk, \delta_1) = e(Dunk, \delta_1) \cup (\delta_1 \setminus \neg pe(Dunk, \delta_1)) = \{Clogged, \neg Armed\}.$$

This also implies that $(\mathcal{D}_1, \Delta_1) \models^0 \neg Armed \text{ after } [Flush; Dunk]$. □

The 0-approximation is sound [13], but it is, in general, incomplete, as shown next.

Example 5. Let \mathcal{D}_2 be the domain obtained from \mathcal{D}_1 by replacing *Dunk causes* $\neg\text{Armed}$ with *Dunk causes* $\neg\text{Armed}$ if *Armed*. We can check that $\neg\text{Armed}$ is true after the execution of $[\text{Flush}; \text{Dunk}]$ if the possible world semantics is used; however, $(\mathcal{D}_2, \Delta_1) \not\models^0 \neg\text{Armed}$ **after** $[\text{Flush}; \text{Dunk}]$. This shows that the 0-approximation is incomplete. \square

The incompleteness of the 0-approximation motivated Son and Tu [14] to find conditions for its completeness. In particular, given a fluent formula φ and (\mathcal{D}, Δ) , they investigate the conditions under which $(\mathcal{D}, \Delta) \models^0 \varphi$ **after** α iff $(\mathcal{D}, \Delta) \models^P \varphi$ **after** α holds. Their conditions are based on the notion of dependency between fluent literals and the reducibility of a set of states to a partial state.

Definition 4. Let \mathcal{D} be a domain description. A fluent literal l depends on a fluent literal g , written as $l \triangleleft g$, iff one of the following conditions holds.

1. $l = g$.
2. \mathcal{D} contains a law $[a \text{ causes } l \text{ if } \psi]$ such that $g \in \psi$.
3. There exists a fluent literal h such that $l \triangleleft h$ and $h \triangleleft g$.
4. The complement of l depends on the complement of g , i.e., $\neg l \triangleleft \neg g$.

Note that the dependency relation between fluent literals is reflexive, transitive but not symmetric. We next define the dependency between actions and fluent literals.

Definition 5. Let \mathcal{D} be a domain description. An action a depends on a fluent literal l , written as $a \triangleleft l$, iff one of the following conditions is satisfied.

1. \mathcal{D} contains an executability condition $[\text{executable } a \text{ if } \psi]$ such that $l \in \psi$.
2. There exists a fluent literal g such that $a \triangleleft g$ and $g \triangleleft l$.

For a fluent literal l (resp. action a), we will denote by $\Omega(l)$ (resp. $\Omega(a)$) the set of fluent literals that l (resp. a) depends on. A fluent literal l (resp. an action a) depends on a set of fluent literals σ , denoted by $l \triangleleft \sigma$ (resp. $a \triangleleft \sigma$), iff $l \triangleleft g$ (resp. $a \triangleleft g$) for some $g \in \sigma$.

A disjunction of fluent literals $\gamma = l_1 \vee \dots \vee l_k$ depends on a set of fluent literals σ , denoted by $\gamma \triangleleft \sigma$ if there exists $1 \leq j \leq k$ such that $l_j \triangleleft \sigma$; otherwise, γ does not depend on σ , denoted by $\gamma \not\triangleleft \sigma$.

Intuitively, $l_1 \triangleleft l_2$ means that knowledge about l_2 might be needed in reasoning about the truth value of l_1 after the execution of some plan; $a \triangleleft l$ means that l might have influence on determining the executability of action a .

Example 6. For the domain description \mathcal{D}_2 , we have

$$\begin{array}{ll} \Omega(\text{Clogged}) = \{\text{Clogged}\} & \Omega(\neg\text{Clogged}) = \{\neg\text{Clogged}\} \\ \Omega(\text{Armed}) = \{\text{Armed}, \neg\text{Armed}\} & \Omega(\neg\text{Armed}) = \{\text{Armed}, \neg\text{Armed}\} \\ \Omega(\text{Dunk}) = \{\neg\text{Clogged}\} & \Omega(\text{Flush}) = \emptyset \end{array}$$

This says that knowledge about *Clogged* is needed in reasoning about *Clogged* and the executability of the action *Dunk*; knowledge about *Armed* is required for reasoning about *Armed*; etc. \square

Definition 6. Let \mathcal{D} be a domain description. Let S be a belief state (i.e., a set of states), δ be a partial state, and $\varphi = \gamma_1 \wedge \dots \wedge \gamma_n$ be a fluent formula where each γ_i is a disjunction of fluent literals. We say that S is reducible to δ with respect to φ , denoted by $S \gg_{\varphi} \delta$ if

1. δ is a subset of every state s in S ,
2. for every $1 \leq i \leq n$, there exists a state $s \in S$ such that $\gamma_i \not\in (s \setminus \delta)$, and
3. for every action a , there exists a state $s \in S$ such that $a \not\in (s \setminus \delta)$.

Intuitively, the above definition specifies a condition under which reasoning using belief states (represented by S) can be done by using the 0-approximation using the approximation state δ . For a set of partial states Δ , we say that $\text{ext}(\Delta) \gg_{\varphi} \Delta$ if for every $\delta \in \Delta$, $\text{ext}(\delta) \gg_{\varphi} \delta$. Indeed, it has been shown in [15] that

Theorem 1. Let (\mathcal{D}, Δ) be an action theory and $\text{ext}(\Delta) \gg_{\varphi} \Delta$. Then, for every sequence of actions α , $(\mathcal{D}, \Delta) \models^P \varphi \text{ after } \alpha$ iff $(\mathcal{D}, \Delta) \models^0 \varphi \text{ after } \alpha$.

Example 7. Consider \mathcal{D}_2 (Example 5). Let $\delta_1 = \emptyset$. Then, $S_1 = \text{ext}(\delta_1)$ is not reducible to δ_1 with respect to $\varphi = \neg \text{Armed}$, i.e., $\text{ext}(\emptyset) \not\gg_{\neg \text{Armed}} \emptyset$, because for each $s \in S_1$, either fluent literal Armed or $\neg \text{Armed}$ belongs to $s \setminus \delta_1$ and the fluent literal $\neg \text{Armed}$ depends on both Armed and $\neg \text{Armed}$ (Condition 2 in Def. 6 is not satisfied). Similarly,

$$\begin{aligned} \text{ext}(\{\text{Clogged}\}) &\not\gg_{\neg \text{Armed}} \{\text{Clogged}\} \\ \text{ext}(\{\neg \text{Clogged}\}) &\not\gg_{\neg \text{Armed}} \{\neg \text{Clogged}\} \end{aligned}$$

Let $\delta_2 = \{\text{Armed}\}$. Then we have $\text{ext}(\delta_2) = \{s_1, s_2\}$, where $s_1 = \{\text{Armed}, \text{Clogged}\}$ and $s_2 = \{\text{Armed}, \neg \text{Clogged}\}$. We will easily check that $\text{ext}(\delta_2) \gg_{\neg \text{Armed}} \delta_2$. Hence, we have $\text{ext}(\{\text{Armed}\}) \gg_{\neg \text{Armed}} \{\text{Armed}\}$. Similarly, we can check that $\text{ext}(\{\neg \text{Armed}\}) \gg_{\neg \text{Armed}} \{\neg \text{Armed}\}$. \square

3 Relating Two Completeness Conditions

The equivalence between situation calculus and the action language \mathcal{A} has been proved in [5] for the case where the initial situation is completely described, where a theory in situation calculus \mathcal{D} is complete if $\mathcal{D}_{S_0} \models F(S_0)$ or $\mathcal{D}_{S_0} \models \neg F(S_0)$ for each fluent F in \mathcal{D} . A natural question is to explore what is the relationship between the two completeness conditions described in the previous sections.

3.1 A Simplification of \mathcal{A} Theories

Let us first discuss some differences between the two formulations through some examples.

Example 8. Consider the theory \mathcal{D} with one action A and three fluents F , P , and Q where $\gamma_F^+(a) = (a = A \wedge P) \vee (a = A \wedge Q)$, $\gamma_F^-(a) = \perp$, and, for $X \in \{P, Q\}$, $\gamma_X^+(a) = \gamma_X^-(a) = \perp$. Furthermore, let $\mathcal{D}_{S_0} = \{P(S_0)\}$. For \mathcal{D} to be context-complete, we have that P and Q should be known in S_0 . That is, S_0 should also contain Q or $\neg Q$ to guarantee that $V(F, \Sigma_0)$ is either 0 or 1.

A reasonable description of \mathcal{D} in the language \mathcal{A} , say $\mathcal{D}_{\mathcal{A}}$, consists of two dynamic laws A **causes** F **if** P and A **causes** F **if** Q and $\Delta = \{\delta_0\}$ where $\delta_0 = \{P\}$. It is easy to check that $ext(\delta_0) \gg_F \delta_0$ and $(\mathcal{D}_{\mathcal{A}}, \Delta) \models^0 F$ **after** A . So, knowledge about Q is not necessary in determining whether $V(F, \Sigma_0)$ is 1 or 0. \square

This example shows that to answer certain queries, the context-complete requirement in [7] might be too strong. The next example shows that, on the other hand, the 0-approximation is somewhat more sensitive to the specification of actions' effects in \mathcal{A} .

Example 9. Consider the theories $(\mathcal{D}_1, \Delta_1)$ and $(\mathcal{D}_2, \Delta_1)$. Recall that the difference between \mathcal{D}_1 and \mathcal{D}_2 lies in that [*Dunk causes* $\neg Armed$] in \mathcal{D}_1 is replaced by

$$\text{Dunk causes } \neg Armed \text{ if Armed}$$

in \mathcal{D}_2 . Intuitively, these two representations are equivalent in the sense that for each complete state of the world s , the execution of *Dunk* in s results in the same state in which the bomb is disarmed, as it is either a direct effect of *Dunk* or it is true by inertial. On the other hand, Examples 4-7 show that \models^0 is complete for the fluent formula $\neg Armed$ w.r.t. \mathcal{D}_1 but not w.r.t. \mathcal{D}_2 . \square

The second example shows a weakness of the 0-approximation in that it is more sensitive to the domain specification than the situation calculus formalism. This is reflected by the completeness condition in [14]: we have that $ext(\Delta_1) \gg_{\neg Armed} \Delta_1$ w.r.t. \mathcal{D}_1 but $ext(\Delta_1) \not\gg_{\neg Armed} \Delta_1$ w.r.t. \mathcal{D}_2 . Why does the situation calculus formulation not suffer from this problem? The main reason is in the encoding of the successor state axioms. For example, the successor state axiom for *Armed* is

$$Armed(Do(a, s)) \equiv \perp \vee (Armed(s) \wedge \neg(a = Dunk \wedge Armed(s)))$$

Here, the precondition *Armed* of the conditional effect $\neg Armed$ of *Dunk* (in the dynamic law [*Dunk causes* $\neg Armed$ **if** *Armed*]) is encoded as a part of the formula $\gamma_{\neg Armed}^-(Dunk)$. However, this can be simplified to

$$Armed(Do(a, s)) \equiv \perp \vee (Armed(s) \wedge \neg(a = Dunk)).$$

which effectively makes $\neg Armed$ to be an effect of *Dunk*. This argument can be seen as another justification for us to simplify \mathcal{D}_2 to \mathcal{D}_1 when using the action language \mathcal{A} .

At this point, one is tempted to conclude that if a complement of a fluent literal L appears in φ of a dynamic law A **causes** L **if** φ , we can remove it from φ and the resulting action theory remains faithful to its original representation. This is, however, not the case. Consider the domain with one action *Flip* that toggles a light bulb switch. The effects are to make the switch *On* and $\neg On$ if it was $\neg On$ and *On*, respectively. This is given by the two laws

$$\text{Flip causes } On \text{ if } \neg On \quad \text{and} \quad \text{Flip causes } \neg On \text{ if } On$$

Removing $\neg On$ or *On* in the **if** part from the first or second law respectively would create an inconsistent domain. Interestingly, the successor state axiom for *On* is

$$On(Do(a, S)) \equiv \gamma_{On}^+(a)[s] \vee (On(s) \wedge \neg \gamma_{On}^-(a)[s]$$

where $\gamma_{On}^+(a) \equiv a = \text{Flip} \wedge \neg \text{On}$ and $\gamma_{On}^-(a) \equiv a = \text{Flip} \wedge \text{On}$. The successor state axiom for On can be simplified to

$$\text{On}(\text{Do}(a, S)) \equiv \gamma_{On}^+(a)[s] \vee (\text{On}(s) \wedge \neg(a = \text{Flip}))$$

Observe that the second representation would yield the same set of models for the theory. Nevertheless, setting $\gamma_{On}^-(a) \equiv a = \text{Flip}$ would violate the consistency condition by Reiter [12], which states that $\exists s. [\text{Poss}(a, s) \supset \gamma_F^+(a, s) \wedge \gamma^- F(a, s)]$. This means that γ_{On}^- cannot be simplified in this case. On the other hand, the simplification for $\gamma_{\text{Armed}}^-(a)$ to $a = \text{Dunk}$ is acceptable since the consistency condition is satisfied by the new formula. This discussion suggests a simplification of \mathcal{A} action theories.

Definition 7. For a fluent literal L , a dynamic law A **causes** L if φ is (A, L) -cyclic if $\neg L \in \varphi$.

L and $\neg L$ are A -relevant in \mathcal{D} if \mathcal{D} contains at least one (A, L) -cyclic and one $(A, \neg L)$ -cyclic dynamic law.

L and $\neg L$ are A -irrelevant if they are not A -relevant.

Let \mathcal{D} be a domain description and \mathcal{D}_R be the domain description obtained from \mathcal{D} by replacing each (A, L) -cyclic dynamic law A **causes** L if φ with A **causes** L if $\varphi \setminus \{\neg L\}$ if L and $\neg L$ are A -irrelevant.

We can see that Armed and $\neg \text{Armed}$ are Dunk -irrelevant in both theories \mathcal{D}_1 and \mathcal{D}_2 . On the other hand, Examples 4-7 show that \mathcal{D}_1 is preferable to \mathcal{D}_2 for dealing with incomplete information. We can show that:

Theorem 2. For every set of partial states Δ , (\mathcal{D}, Δ) is equivalent to (\mathcal{D}_R, Δ) w.r.t. the possible world semantics.

Proof. (Sketch) It is easy to check that $\Phi_D(a, s) = \Phi_{D_R}(a, s)$ for every action a and state s . This implies that the two theories are equivalent w.r.t. the possible world semantics. \square

3.2 From a Situation Calculus Theory to an Action Theory

We will now show a result relating the context-completeness condition in [7] and the reducibility condition in [14]. We begin with a translation of local effect situation calculus theories into \mathcal{A} action theories and conclude with a theorem relating these two conditions.

Our translation from a situation calculus theory \mathcal{D} into $(\mathcal{D}_{\mathcal{A}}, \Delta_{\mathcal{A}})$ is inspired by the translation in [5] (we will only deal with propositional theories). Assume that

$$\mathcal{D} = \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{una} \cup \mathcal{D}_0$$

Let \mathbf{F} and \mathbf{A} denote the set of fluents and actions of $(\mathcal{D}_{\mathcal{A}}, \Delta_{\mathcal{A}})$ respectively. The translation is done as follows.

- for each action precondition axiom

$$\text{Poss}(A, s) \equiv \Pi_A[s] \text{ in } \mathcal{D}_{ap}$$

A belongs to \mathbf{A} and $\mathcal{D}_{\mathcal{A}}$ contains **executable** A if Π_A .

- for each successor state axiom

$$F(do(a, s)) \equiv \gamma_F^+(a)[s] \vee (F(s) \wedge \neg\gamma_F^-(a)[s]) \text{ in } \mathcal{D}_{ss},$$

F belongs to \mathbf{F} and

- for each disjunct $[a = A \wedge \phi]$ in $\gamma_F^+(a)$, \mathcal{D}_A contains

A causes F if ϕ .

- for each disjunct $[a = A \wedge \phi]$ in $\gamma_F^-(a)$, \mathcal{D}_A contains

A causes $\neg F$ if ϕ

- Let $\Delta_A = \{\delta_0 \mid \delta_0 \text{ is a minimal set of fluent literals satisfying } \mathcal{D}_{S_0}\}$.

It is easy to check that the action theory (D_1, Δ_1) (Example 3) is the result of the above translation from the theory \mathcal{D}^b in Example 1. Similarly to [5], we can prove:

Theorem 3. *For every fluent formula φ and action sequence $\alpha = [a_1, \dots, a_n]$ in \mathcal{D} , $\mathcal{D} \models \text{Poss}(\alpha, S_0) \wedge \varphi(Do(\alpha, S_0))$ iff $(\mathcal{D}_A, \Delta_A) \models^P \varphi$ after α .*

Proof. To prove the theorem, we show that

- For every model M of \mathcal{D} , the state $s_M = \{F \mid F \text{ is a fluent literal, } M \models F(S_0)\}$ belongs to $\text{ext}(\Delta_A)$ and for every action sequence α and fluent literal F , $M \models \text{Poss}(\alpha, S_0) \wedge F(Do(\alpha, S_0))$ iff $\widehat{\Phi}^0(\alpha, s_M) \neq \perp$ and F is true in $\widehat{\Phi}^0(\alpha, s_M)$.
- For every $s_M \in \text{ext}(\Delta_A)$ there exists a model M of \mathcal{D} such that for every action sequence α and fluent F : $\widehat{\Phi}^0(\alpha, s_M) \neq \perp$ and F is true in $\widehat{\Phi}^0(\alpha, s_M)$ iff $M \models \text{Poss}(\alpha, S_0) \wedge F(Do(\alpha, S_0))$. \square

The next theorem relates the context-complete condition on \mathcal{D} and the reducibility condition on $(\mathcal{D}_A, \Delta_A)$.

Theorem 4. *Let \mathcal{D} be a context-complete situation calculus theory and $(\mathcal{D}_A, \Delta_A)$ be its \mathcal{A} -translation. It holds that $\text{ext}(\Delta_A) \gg_\varphi \Delta_A$ for each fluent formula φ .*

Proof. The proof of the theorem relies on the following results:

- For every $\delta \in \Delta_A$, every law $[A \text{ causes } F \text{ if } \varphi]$ in \mathcal{D}_A , and fluent literal l appearing in φ , δ contains either l or $\neg l$;
- For every fluent literal g , $s \in \text{ext}(\delta)$, $\delta \in \Delta_A$, and $g \in (s \setminus \delta)$, there does not exist a fluent literal l , $l \neq g$, such that $l \triangleleft g$;
- If ψ is a fluent formula in normal form then for every δ in Δ_A there exists a state s in $\text{ext}(\delta)$ such that ψ contains no fluent literals in $(s \setminus \delta)$. \square

This indicates that the context-complete condition is actually more restrictive than reducibility. Observe that, in order to obtain the same form of action theory described earlier, it is necessary to convert the formulae in the conditions of the dynamic causal laws to disjunctive normal form and distribute the disjuncts in separate laws. We are working on extending the 0-approximation to allow the use of arbitrary propositional formulae in the dynamic causal laws.

4 Conclusions

In this paper, we studied the relationship between two completeness conditions for the 0-approximation. The insights gained through the study allowed the development of a simplification procedure of a \mathcal{A} action theory to an equivalent theory whose completeness condition can be weakened for several classes of queries.

We also showed that the context-complete condition on local effect action theories proposed in [7] implies the reducibility condition for action theories in the language \mathcal{A} developed in [14].

Acknowledgments: The authors are partially supported by NSF grants CNS-0220590, IIS 0812267, and HRD-0420407.

References

1. Balduccini, M., Gelfond, M.: Diagnostic Reasoning with A-Prolog. *Theory and Practice of Logic Programming* 3(4,5), 425–461 (2003)
2. Baral, C., Gelfond, M.: Reasoning agents in dynamic domains, pp. 257–279. Kluwer Academic Publishers, Dordrecht (2000)
3. Baral, C., Kreinovich, V., Trejo, R.: Computational complexity of planning and approximate planning in the presence of incompleteness. *Artificial Intelligence* 122, 241–267 (2000)
4. Gelfond, M., Lifschitz, V.: Representing actions and change by logic programs. *Journal of Logic Programming* 17(2,3,4), 301–323 (1993)
5. Kartha, G.: Soundness and completeness theorems for three formalizations of action. In: *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 724–729. Morgan Kaufmann Publishers, San Mateo (1993)
6. Levesque, H.J.: A completeness result for reasoning with incomplete first-order knowledge bases. In: KR, pp. 14–23 (1998)
7. Liu, Y., Levesque, H.: Tractable reasoning with incomplete first-order knowledge in dynamic systems with context-dependent actions. In: *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI, Edinburgh, Scotland (2005)
8. McCarthy, J.: Programs with common sense. In: *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, London, pp. 75–91. Her Majesty's Stationery Office (1959)
9. McCarthy, J., Hayes, P.: Some philosophical problems from the standpoint of artificial intelligence. In: Meltzer, B., Michie, D. (eds.) *Machine Intelligence*, vol. 4, pp. 463–502. Edinburgh University Press, Edinburgh (1969)
10. McDermott, D.: A critique of pure reason. *Computational Intelligence* 3, 151–160 (1987)
11. Moore, R.: A formal theory of knowledge and action. In: Hobbs, J., Moore, R. (eds.) *Formal theories of the commonsense world*. Ablex, Norwood (1985)
12. Reiter, R.: The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In: *Artificial Intelligence and Mathematical Theory of Computation*, pp. 359–380. Academic Press, London (1991)
13. Son, T.C., Baral, C.: Formalizing sensing actions - a transition function based approach. *Artificial Intelligence* 125(1-2), 19–91 (2001)
14. Son, T.C., Tu, P.H.: On the Completeness of Approximation Based Reasoning and Planning in Action Theories with Incomplete Information. In: *Int. Conf. on Principles of Knowledge Representation and Reasoning*, pp. 481–491 (2006)
15. Tu, P.H.: Reasoning and Planning With Incomplete Information in the Presence of Static Causal Laws. Ph.D thesis, New Mexico State University (2007)

KT and S4 Satisfiability in a Constraint Logic Environment

Lynn Stevenson^{1,2}, Katarina Britz^{1,2}, and Tertia Hörne²

¹ Meraka Institute, CSIR, South Africa

² School of Computing, University of South Africa

mrlstevenson@gmail.com,

arina.britz@meraka.org.za,

hornet@unisa.ac.za

Abstract. The modal satisfiability problem is solved either by using a specifically designed algorithm, or by translating the modal logic formula into an instance of a different class of problem, such as a first-order logic problem, a propositional satisfiability problem, or, more recently, a constraint satisfaction problem. In the latter approach, the modal formula is translated into layered propositional formulae. Each layer is translated into a constraint satisfaction problem which is solved using a constraint solver. We extend this approach to the modal logics *KT* and *S4* and introduce a range of optimizations of the basic prototype. The results compare favorably with those of other solvers, and support the adoption of constraint programming as implementation platform for modal and other related satisfiability solvers.

1 Introduction

One of the reasoning tasks associated with modal logic is the modal satisfiability problem. This is a decision problem that returns ‘yes’ if an algorithm can generate some model in which a given formula is satisfiable. This problem has been researched extensively using different automated approaches. These approaches fall into two distinct categories. Either a special, purpose-build algorithm is used, or the modal formula is translated into an instance of a different class of problem, and solved with the highly optimized solvers available for that class. A widely used class of purpose-built algorithms are based on tableau methods, and include the tableau solvers FaCT [1] and DLP [2]. A well-known translation method is the translation of modal formulae into first-order logic [3]. Other translation-based approaches include translation into a propositional satisfiability problem (SAT) [4], or a constraint satisfaction problem (CSP) [5,6].

In this paper, we further investigate the feasibility of the constraint-based approach proposed by Brand et al. [5,6]. A modal formula is stratified into layers, each of which is solved using the constraint logic programming (CLP) language, *ECLⁱPS^e* [7]. The benefit of this approach is the ability to make use of an existing, well-developed constraint programming language, with its mature solvers and predefined libraries of functions. A key contribution of this approach is that

the domains of variables are assigned values over and above the Booleans 0 and 1: a value of u can be assigned to a variable, thereby allowing partial assignments. This has the effect of considerably speeding up finding a solution. As in many other application domains, the advantage of a constraint-based approach to the satisfiability problem is its support for sophisticated conceptual modelling by means of constraints.

The solver developed by Brand et al. is, however, limited to the modal logic K . It only deals with formulae that are in conjunctive normal form (CNF), and have not been optimized using any of the standard techniques such as caching. In this paper, we discuss the extension of this solver to the modal logics KT and $S4$. A number of enhancements to the original prototype are discussed, the most significant of which are the following: We relax the requirement that formulae be in CNF; instead, formulae are kept in a negation normal form (NNF). Extensive simplification is applied to NNF clauses using propositional unit literals (l and $\neg l$) and unit modal literals ($\Box l$ and $\neg \Box l$). Where a propositional variable occurs only positively or only negatively in a layered formula, all clauses in which it occurs are excluded from the translation into the CSP. This significantly prunes the search space. Caching of formulae and their status is introduced, which reduces reprocessing.

These enhancements return favorable results that compare well with the results of the solvers FaCT, DLP and KSAT [8]. These solvers are highly optimized, whereas our results have been obtained without optimized data structures. This means that there is a strong case for incorporating constraint methods into tableau solvers, and to develop tableau solvers using constraint programming. Our findings therefore support the use of constraint programming as a feasible environment for the implementation of modal satisfiability solvers. This approach is also applicable to related areas such as description logic reasoners.

The remainder of this paper is organized as follows: Section 2 provides the theoretical background of the KCSP solver developed by Brand et al. [6]. Sections 3 and 4 provide details of the KT_CSP and S4_CSP solvers for KT and $S4$ respectively, and discuss the results obtained using the Heuerding / Schwendimann test data sets. In Section 5, the exponential nature of the results is discussed and they are compared with those of the TANCS '98 conference. The final section (Section 6) includes details of possible further areas of research.

2 Background to the KCSP Solver

We introduce some of the terminology used in the work that follows. The reader is referred to texts such as Blackburn et al. [9] for in-depth details of modal logic.

Definition 1. *The basic modal language K is defined using a set Φ of atomic propositions, the elements of which are denoted $p, p_1, \dots, q, q_1, \dots$, the propositional connectives \neg and \wedge , and the unary modality \Box . The set of well-formed formulae generated from Φ , denoted $Fma(\Phi)$, is generated by the rule*

$$\varphi ::= p \mid \perp \mid \neg\varphi \mid \varphi \wedge \psi \mid \Box\varphi$$

where p ranges over the elements of Φ , \perp is the falsum and φ, ψ, \dots are modal formulae.

A *propositional atom* is any propositional formula that cannot be decomposed propositionally. A *propositional literal* is either a propositional atom or its negation. A *modal atom* is any modal formula that cannot be decomposed propositionally – that is, any formula whose main connective is not propositional. A *modal literal* is either a modal atom or its negation.

In the context of this paper, a modal formula can be normalized into either NNF or CNF. A CNF formula consists of the conjunction of clauses, where each CNF clause is the disjunction of propositional literals and / or modal literals. A modal formula is in NNF if negation occurs only immediately before propositional and modal atoms and the only Boolean connectives it contains are $\{\neg, \wedge, \vee\}$. A modal NNF formula consists of the conjunction of NNF clauses, where each NNF clause consists of a disjunction of NNF formulae. A *propositional unit clause* is any clause l where l is a propositional literal. A *modal unit clause* is any clause $\Box\varphi$ or $\neg\Box\varphi$ where φ is any formula. A *unit modal literal* is any clause $\Box l$ or $\neg\Box l$, where l is a propositional literal.

The satisfiability of a modal formula can be determined by translating it into layers of propositional formulae. This approach was first proposed by F. Giunchiglia and R. Sebastiani [4], and implemented in their KSAT solver. The modal atoms of a particular layer are processed as though they are propositional atoms, and truth values are assigned to them. Whenever a modal layer contains a $\neg\Box$ -modality, further processing at the next modal layer is required. The KSAT solver makes use of the well-known DPLL SAT algorithm to determine satisfiability of the propositional formula at each layer.

A related approach was proposed by Brand et al. [6], and implemented in their KCSP solver. In this case, the stratified modal formula is translated into a CSP. A CSP consists of a set of variables, a domain for each variable and a set of constraints. The variables can be assigned any value in their corresponding domain, with the limitation that the constraints on the variables need to be satisfied. The constraints therefore limit the scope of the variables.

In both KSAT and KCSP, the modal formula needs to be in conjunctive normal form. The propositional approximation of the modal formula is fed to the solver, which returns a set of truth assignments to the propositional and modal atoms, termed the *top-level* atoms, at the current modal level. This is defined formally as follows:

Definition 2. A total truth assignment μ for a modal K formula φ is a set of literals

$$\mu = \{\Box\alpha_1, \dots, \Box\alpha_n, \neg\Box\beta_1, \dots, \neg\Box\beta_m, p_1, \dots, p_r, \neg q_1, \dots, \neg q_s\}$$

such that every top-level atom of φ occurs either positively or negatively in μ .

Theorem 1. [4] A modal formula φ is K -satisfiable if and only if there exists a K -satisfiable truth assignment μ such that $\mu \models_p \varphi$.

This means that the K -satisfiability of a formula φ can be reduced to determining the K -satisfiability of its truth assignments. Such an assignment is termed total when a truth value is assigned to each top-level atom; it is termed a partial assignment when the truth values ensure the satisfiability of φ . The satisfiability of the modal portion of φ is then determined as follows.

Definition 3. *The restricted truth assignment μ^r for a modal K formula φ is defined as*

$$\mu^r = \bigwedge_i \square\alpha_i \wedge \bigwedge_j \neg\square\beta_j$$

Theorem 2. [4] *The restricted truth assignment μ^r is satisfiable if and only if the formula*

$$\varphi_j = \bigwedge_i \alpha_i \wedge \neg\beta_j$$

is K -satisfiable for every $\neg\square\beta_j$ occurring in μ^r .

Constraints are defined on clauses in the modal formula. A clause $(\neg p_1 \vee p_2)$ has the constraints that $p_1 = 0$ and / or $p_2 = 1$. A clause $\square p_4$ has the constraint that $\square p_4 = 1$.

A constraint solver always returns a *total assignment* to its variables. Because less computational effort is required to return a partial assignment, Brand et al. [6] followed an approach of setting the domain of each variable to $\{0, 1, u\}$, where u indicates that a truth value has not been assigned to the associated variable.

A modal formula is negated and converted into CNF before being passed to KCSP.

Algorithm 1. [6] The KCSP algorithm schema:

```

function KCSP( $\varphi$ ) // succeeds if  $\varphi$  is satisfiable
     $\varphi_{csp} = to\_csp(\varphi);$ 
     $\mu := csp(\varphi_{csp});$ 
     $\Theta = \bigwedge \{\alpha : \square\alpha = 1 \text{ is in } \mu\};$ 
    for each  $\square\beta = 0$  in  $\mu$  do
        KCSP( $\Theta \wedge \neg\beta$ ); // backtrack if this fails
    end;

```

The procedure *to_csp* identifies the top-level atoms of φ , sets their domains to $\{0, 1, u\}$ and defines the constraints on each clause. The resulting formula is then passed to the *ECLⁱPS^e* constraint solver with the call to *csp* which returns a truth assignment. If it contains negative literals, that is, literals to which a value of 0 has been assigned, further processing is required. Each of the modal literals $\neg\square\beta_j$ effectively generates a new branch of the modal tree. The conjunction of each β_j and the α_i variables having $\square\alpha_i = 1$, form the modal formula that will be processed at the next modal layer. If there are no negative modal literals, the formula is satisfiable and no further processing is required. If there are negative modal literals, the algorithm backtracks.

Using a domain of $\{0, 1, u\}$ reduces the processing requirement. If all the $\neg\Box\beta_j$ variables at a particular modal layer can be assigned a value of u , no further processing will be required.

Various optimizations have been applied to the solver to reduce the search space. These included simplification of the initial formula by applying *unit subsumption* and *unit resolution*. When unit subsumption is applied to a modal formula containing a unit clause l , every clause containing l is removed. When unit resolution is applied to a modal formula, $\neg l$ is removed from every clause in which it occurs. Further details of these optimizations are provided in [6,5].

3 The KT-CSP Solver

The KCSP prototype described in the previous section yielded some promising results, but did not yet establish the constraint-based approach to modal satisfiability as viable alternative to existing solvers. This is due to two factors: Firstly, few of the standard optimisations to tableau solvers were implemented, so the modelling benefits could not be appreciated fully, and secondly, the prototype only addressed the modal logic K. We address these issues below.

To extend KCSP to the modal logic *KT*, the solver was modified to allow for a reflexive accessibility relation. This introduces two challenges – the number of clauses in a formula is significantly increased, and the formula is no longer in CNF. To address the first challenge, we propose the following lemma, which has the effect of reducing the number of clauses generated. Full details including further examples and proofs of lemmas are available in [10].

Lemma 1. *Applying the axiom $\Box^n\varphi \rightarrow \varphi$ at each modal layer, to each occurrence of $\Box^n\varphi$, is a sound and complete strategy to enforce the reflexivity of R in the KT-CSP algorithm.*

However, applying the lemma yields a formula which is no longer in CNF. We find that, when Lemma 1 is applied to a modal clause with n positive modal literals, and the resulting formula is converted to CNF, the original modal clause is replaced by 2^n modal clauses. There is thus an exponential increase in the number of clauses when a modal formula is converted to CNF.

Two prototypes were developed to deal with reflexivity. In the one, the formulae were retained in CNF and in the other, they were not. We discuss only the second prototype since it produced better results. The approach followed when dealing with clauses that are not in CNF is illustrated by the following example.

Example 1. Consider the modal formula

$$\varphi = \Box\psi \wedge (\neg\Box\psi_1 \vee (\neg\Box\psi_2 \wedge \neg\Box\psi_3)).$$

When we convert it to CNF and apply Lemma 1, we get

$$\varphi' = \Box\psi \wedge \psi \wedge (\neg\Box\psi_1 \vee \neg\Box\psi_2) \wedge (\neg\Box\psi_1 \vee \neg\Box\psi_3).$$

There are several possible truth assignments that the constraint solver can return: $\mu_1 = \{\Box\psi, \psi, \neg\Box\psi_1\}$, $\mu_2 = \{\Box\psi, \psi, \neg\Box\psi_1, \neg\Box\psi_2\}$, $\mu_3 = \{\Box\psi, \psi, \neg\Box\psi_1, \neg\Box\psi_3\}$ and $\mu_4 = \{\Box\psi, \psi, \neg\Box\psi_2, \neg\Box\psi_3\}$.

If we do not convert the formula to CNF, and apply Lemma 1, we can verify the satisfiability of $\varphi_1 = \Box\psi \wedge \psi \wedge \neg\Box\psi_1$ and $\varphi_2 = \Box\psi \wedge \psi \wedge \neg\Box\psi_2 \wedge \neg\Box\psi_3$, with the proviso that φ_2 is processed only if φ_1 is not satisfiable. The possible truth assignments the constraint solver will return, are $\mu_5 = \{\Box\psi, \psi, \neg\Box\psi_1\}$ and $\mu_6 = \{\Box\psi, \psi, \neg\Box\psi_2, \neg\Box\psi_3\}$.

Now suppose $\neg\psi_1$ is unsatisfiable. For a complex formula, this could take considerable resources to establish. When the formula is converted to CNF, $\neg\psi_1$ is processed three times, causing backtracking each time; if it is not converted, it is processed only once. \dashv

We therefore propose that, instead of converting the formula into CNF, it is retained in NNF. We first apply Lemma 1, and then *selectively construct* the formula to convert to a CSP.

Definition 4. An NNF clause ψ in a modal formula φ is represented as

$$\psi = \psi' \vee \theta_1 \vee \dots \vee \theta_n, \text{ where}$$

$$\psi' = \bigvee \{l : l \in P\} \vee \bigvee \{\Box\alpha : \Box\alpha \in B^+\} \vee \bigvee \{\neg\Box\beta : \Box\beta \in B^-\},$$

P is a set of propositional literals, B^+ and B^- are sets of modal atoms, and $\theta_1, \dots, \theta_n$ are NNF formulae.

After the modal formula has been negated and converted to NNF, the following algorithm is applied.

Algorithm 2. The KT-CSP algorithm schema:

```

function KT_CSP( $\varphi$ )                                // succeeds if  $\varphi$  is satisfiable
     $\varphi_{kt} = \text{apply\_reflexivity}(\varphi);$ 
     $\varphi_{formula} = \text{construct\_formula}(\varphi_{kt});$ 
     $\varphi_{csp} = \text{to\_csp}(\varphi_{formula});$ 
     $\mu := \text{csp}(\varphi_{csp});$                       // backtrack if this fails
     $\Theta = \bigwedge \{\alpha : \Box\alpha = 1 \text{ is in } \mu\};$ 
    for each  $\Box\beta = 0$  in  $\mu$  do
        KT_CSP( $\Theta \wedge \neg\beta$ );                  // backtrack if this fails
    end;

```

Lemma 1 is applied to the input formula, after which the function *construct_formula* proceeds as follows: Each clause in φ_{kt} is grouped as per Definition 4. The formula $\varphi_{formula}$ is constructed as the conjunction of the ψ' components of each NNF clause. If there is no ψ' component in an NNF clause, the θ_1 component is used instead. This formula is then converted into a CSP and fed to the constraint solver. If the constraint solver cannot find a solution, it backtracks to *construct_formula* and a new formula is built using the remaining θ_i clauses.

This prototype was tested using the Heuerding / Schwendimann data sets [11] which consist of nine classes of 21 provable formulae (the ‘p’ data sets) and 21 unprovable formulae (the ‘n’ datasets). Data sets are available for each of the logics K , KT and $S4$ and can be downloaded off the Internet [12]. The formulae in each class get progressively more complex. The capability of a solver is measured in terms of the number of ‘p’ and ‘n’ data sets it can solve in a particular class in *less than* 100 CPU seconds. Thus, if a result of 3 is recorded for the `k_branch_n` data sets, this means that only the first 3 of the 21 data sets could be solved in under 100 CPU seconds. Whenever all 21 data sets are solved, the symbol ‘>’ is used. The initial prototype did not return particularly good results. Table 1 lists the results per class, for each category. A series of enhancements was therefore applied.

Table 1. Initial Results of the KT-CSP Prototype

	kt_branch	kt_45	kt_dum	kt_grz	kt_md	kt_path	kt_ph	kt_poly	kt_t4p
n	3	8	14	>	5	10	7	2	3
p	2	8	5	9	4	2	4	>	3

Propositional and Modal Simplification: In the KCSP solver, both subsumption and unit resolution are applied to the initial modal formula. In the KT-CSP prototype, the modal formula is no longer in CNF, making this approach more complex. In addition, because the application of Lemma 1 generates further propositional variables, simplification is now required at each modal layer. We therefore reconsider the unit subsumption and unit resolution rules applied in the DPLL SAT procedure and extend them to include NNF clauses and unit modal literals. The resulting reduction in the number of choice points leads to a significant improvement in the results obtained.

Enhancement 1. *We apply the following rules after the application of Lemma 1 to the modal formula at each modal layer:*

1. *For every clause ψ that is either a propositional unit clause or a unit modal literal, unit subsumption is applied to every other NNF clause containing ψ and such clauses are removed, provided that ψ does not occur in a modal formula within the NNF clause.*
2. *For every clause ψ that is either a propositional unit clause or a unit modal literal, unit resolution is applied and $\neg\psi$ is removed from every other NNF clause in which it occurs, provided that $\neg\psi$ does not occur in a modal formula within the NNF clause.*

This process is repeated until no further simplification is possible.

Early Pruning: Lemma 1 and then simplification are applied to each modal formula. An analysis of some of the data sets showed that we can have a scenario at the next modal layer where, after processing a number of the $\varphi_j = \bigwedge_i \alpha_i \wedge \neg\beta_j$

formulae, a φ_j can be encountered which has p in some α_i and $\neg p$ in $\neg\beta_j$. Such a formula is unsatisfiable. Its early detection prevents unnecessary processing. This observation led to the following enhancement:

Enhancement 2. Let $\varphi' = \psi_1 \wedge \bigwedge_j \neg\beta_j$ where $\psi_1 = \bigwedge_i \alpha_i$. If a propositional unit clause l in $\bigwedge_j \neg\beta_j$ also occurs in ψ_1 , force a backtrack to the previous modal layer.

Grouping of Clauses: By grouping disjoint clauses and processing each group separately, the number of choice points can be reduced.

Enhancement 3. Suppose we have a modal formula φ . Let $\gamma = \{p_1, \dots, p_n\}$ be the set of propositional atoms p_i occurring in φ , where p_i can occur at any modal layer in φ . Partition the clauses in φ as follows:

$$\varphi = \psi_1 \wedge \dots \wedge \psi_m,$$

where each ψ_i is a conjunction of NNF clauses and for each $p_k \in \gamma$, if p_k occurs in ψ_i , then p_k does not occur in any other ψ_j , where $j \neq i$. By determining the satisfiability of each ψ_i , we determine the satisfiability of φ .

Value Assignments: Any top-level propositional literal that *only* occurs positively or that *only* occurs negatively in the modal formula may, without loss of generality, be assigned a value of 1 or 0 respectively. Therefore, when the CSP for these literals is constructed, their domains can be limited to {1} and {0} respectively, instead of to {0, 1, u}. If a clause contains a positive propositional literal p to which a value of 1 has been assigned, this clause is *True* since the clause is the disjunction of variables. We therefore do not need to pass this clause to the constraint solver. Again, by reducing the number of clauses in the CSP, we reduce the number of choice points. Note that this case differs from unit subsumption in that we are now dealing with propositional literals that are not propositional unit clauses.

Enhancement 4. Suppose we have a modal formula φ . Let $\gamma = \{p_1, \dots, p_{n1}, \neg q_1, \dots, \neg q_{n2}\}$, where if $p_i \in \gamma$, then p_i occurs only positively in φ and if $q_j \in \gamma$, then q_j occurs only negatively in φ . We apply the following rule to the clauses in φ : For each clause ψ in φ , if ψ contains a single propositional literal p and $p \in \gamma$, then this clause is removed from φ .

Caching: Some formulae repeat at various nodes – in some cases, there is a prolific propagation of the same formula. A solution to this problem is to be found in the introduction of a cache.

Enhancement 5. Formulae, together with their satisfiability status, are stored in a cache. Before a new formula φ is processed, the store is checked to see if it has already been cached. If it has, and has been marked as satisfiable, no further processing is required. If it has been marked as unsatisfiable, a backtrack is forced.

If φ has not yet been processed, check to see if it is a subformula of any modal formula φ' that has been cached. If this is the case and φ' has been marked as satisfiable, no further processing is required. Otherwise, if a subformula of φ has been cached and marked as unsatisfiable, a backtrack is forced.

If no information is available in the cache for φ , it is added to the cache with a status of unsatisfiable. It is then processed, and only if it is found to be satisfiable is its status in the store updated.

Enhancements 1–5 significantly improved the results of Table 1. The final results are provided in Table 2.

Table 2. Final results of the KT_CSP prototype

	kt_branch	kt_45	kt_dum	kt_grz	kt_md	kt_path	kt_ph	kt_poly	kt_t4p
n	10	>	>	>	6	>	7	10	>
p	>	>	>	>	5	>	4	>	>

4 The S4_CSP Solver

Applying Lemma 1 to enforce reflexivity, and the axiom $\square\varphi \rightarrow \square\square\varphi$ to enforce transitivity of the accessibility relation in $S4$, leads to the same looping behavior experienced by other solvers. In addition to looping, we note that the modal depth of a formula can remain unchanged at each successive modal layer. To deal with modal $S4$ formulae, we introduce the following two lemmas, the latter of which defines the *stopping condition* which prevents the algorithm from looping when enforcing transitivity.

Lemma 2. Applying the axiom $\square^n\varphi \rightarrow (\square\square\varphi \wedge \square\varphi \wedge \varphi)$ at each modal layer to each occurrence of $\square^n\varphi$ is a sound and complete strategy to enforce reflexivity and transitivity of R in the $S4$ -CSP algorithm.

Lemma 3. Suppose we have a modal formula φ at modal layer n and suppose φ is also the modal formula generated at modal layer $n+1$. No further processing of this branch is required, as it is satisfiable.

The S4_CSP algorithm differs from the KT_CSP algorithm in two ways - Lemma 2 is applied to the modal formula at each modal layer, instead of Lemma 1 and loop checking is implemented, as per Lemma 3.

The results obtained by the initial S4_CSP prototype are listed in Table 3. The results of the $s4_45$ data sets are particularly bad. Note that the p -data sets generally returned better results than the n -data sets.

Simplification and Early Pruning Revisited: Consider the two clauses $(p_4 \vee (p_2 \wedge (\square p_1 \vee \square p_0))) \wedge \square p_0$. When Lemma 2 is applied, they become $(p_4 \vee$

Table 3. Initial results of the S4_CSP prototype

	s4_branch	s4_45	s4_grz	s4_ipc	s4_md	s4_path	s4_ph	s4_s5	s4_t4p
n	8	2	>	8	8	9	7	5	10
p	>	1	>	>	10	10	4	5	13

$(p_2 \wedge ((\Box \Box p_1 \wedge \Box p_1 \wedge p_1) \vee (\Box \Box p_0 \wedge \Box p_0 \wedge p_0)) \wedge (\Box \Box p_0 \wedge \Box p_0 \wedge p_0)$. We now have a set of clauses which is difficult to simplify. However, if simplification was applied *before* Lemma 2, we would have the formula $(p_4 \vee p_2) \wedge \Box p_0$ which is much easier to deal with. This leads to the following enhancements:

Enhancement 6. *For each unit modal literal ψ in a modal formula φ , we apply the rules of Enhancement 1 to every other clause containing ψ before Lemma 2 is applied to φ .*

Enhancement 7. *We apply the following two simplification rules to every propositional unit clause and unit modal literal ψ in a modal formula φ :*

1. *NNF-unit subsumption is applied to every other clause containing ψ : A formula $\varphi_1 \wedge (\psi_1 \vee (\psi \wedge \psi_2)) \wedge \psi$, in which φ_1 consists of the conjunction of any number of NNF clauses and ψ_1 and ψ_2 are NNF clauses, is replaced with $\varphi_1 \wedge (\psi_1 \vee \psi_2) \wedge \psi$.*
2. *NNF-unit resolution is applied to every other NNF clause containing $\neg\psi$: A formula $\varphi_1 \wedge (\psi_1 \vee (\neg\psi \wedge \psi_2)) \wedge \psi$, whose variables are defined as in 1. above, is replaced with $\varphi_1 \wedge \psi_1 \wedge \psi$.*

Enhancements 6–7, together with a re-implementation of enhancement 2, led to a considerable improvement in results, particularly for the *s4_45* datasets. The final results of the S4_CSP solver are listed in Table 4.

Table 4. Final results of the S4_CSP prototype

	s4_branch	s4_45	s4_grz	s4_ipc	s4_md	s4_path	s4_ph	s4_s5	s4_t4p
n	8	14	>	8	9	20	8	6	12
p	>	12	>	>	12	19	4	5	13

5 Comparison of Results

In order to gain further insights into these results, it is necessary to compare them with the results of existing state-of-the-art solvers. Few solvers can effectively deal with the modal logics *KT* and *S4* – our research showed that only FaCT and DLP have been optimized for *S4*. See [10] for further details.

As already discussed, the formulae in each class of data sets get progressively more complex. This causes results such as:

Data set number	9	10	11	12	13
CPU secs	28.17	95.14	360	1320	6605

in the case of *kt_branch_n*. These results clearly demonstrate exponential behavior, which justifies the comparison of our results with those of the TANCS-1998 competition, which was based on the Heuerding / Schwendimann data sets. The results obtained for the *KT* data sets by the KSAT, DLP, FaCT solvers [8],

together with the results of the KT_CSP solver, are listed in Table 5. In Table 6, we list the results obtained for the *S4* data sets by the DLP, FaCT and S4_CSP solvers (the KSAT solver does not support *S4*). As can be seen, the results of the KT_CSP and S4_CSP prototypes compare favorably.

If we were to rerun the TANCS-1998 results on the hardware used for our benchmark results, the results would obviously be much better. For example, the DLP solver would inevitably solve all 21 *kt_ph_n* data sets. However, in the case of data sets such as *kt_branch_n*, the improvement would be in the order of 1 to at most 4 data sets. If one considers the results for *kt_branch_n* listed above, the hardware would need to be considerably faster to solve the 12th data set in under 100 CPU seconds.

Table 5. Results of the Heuerding / Schwendimann KT data sets

	FaCT	DLP	KSAT	KT_CSP	FaCT	DLP	KSAT	KT_CSP
	n	n	n	n	p	p	p	p
kt_45	>	>	5	>	>	>	5	>
kt_branch	4	11	7	10	6	16	8	>
kt_dum	>	>	12	>	11	>	7	>
kt_grz	>	>	>	>	>	>	9	>
kt_md	5	>	4	6	4	3	2	5
kt_path	3	>	5	>	5	6	2	>
kt_ph	7	18	5	7	6	7	4	4
kt_poly	7	6	2	10	>	6	1	>
kt_t4p	2	>	1	>	4	3	1	>

Table 6. Results of the Heuerding / Schwendimann S4 data sets

	FaCT	DLP	S4_CSP	FaCT	DLP	S4_CSP
	n	n	n	p	p	p
s4_branch	4	8	8	4	10	>
s4_45	>	>	14	>	>	12
s4_grz	>	>	>	2	9	>
s4_ipc	4	>	8	5	10	>
s4_md	4	>	9	8	3	12
s4_path	1	>	20	2	3	19
s4_ph	4	18	8	5	7	4
s4_s5	2	>	6	>	3	5
s4_t4p	3	>	10	5	>	13

6 Conclusion and Future Work

A strength of translating each modal layer into a constraint satisfaction problem lies in the ability to limit the domain of a propositional variable which occurs only positively or only negatively to a single value. This allows us to reduce the

number of conjunctive clauses in the modal layer, thereby significantly reducing the search space. Because the modal problem has been stratified into layers, this is easily implemented.

Although we obtained good results, we have identified areas for further improvement. Firstly, the selection criteria applied when constructing the CSP from the NNF formula can be further enhanced. Secondly, further improvement is possible by optimizing the data structures to exploit normal forms for modal logics. Finally, our results support the adoption of constraint programming as underlying formalism for description logic reasoners. This should lead to improved scope for taking advantage of the improved modelling opportunities provided by the constraint-approach, especially when dealing with more expressive description logics. We are currently investigating this further.

References

1. Horrocks, I.: The FaCT System. In: de Swart, H. (ed.) TABLEAUX 1998. LNCS (LNAI), vol. 1397, pp. 307–312. Springer, Heidelberg (1998)
2. Patel-Schneider, P.F.: DLP system description. In: Proceedings of the 1998 International Workshop on Description Logics Workshop (DL 1998), vol. 11, pp. 87–89 (1998), CEUR-WS.org
3. Ohlbach, H., Nonnengart, A., de Rijke, M., Gabbay, D.: Encoding two-valued non-classical logics in classical logic. In: Handbook of Automated Reasoning, pp. 1403–1486. Elsevier Science, Amsterdam (2001)
4. Giunchiglia, F., Sebastiani, R.: Building Decision Procedures for Modal Logics from Propositional Decision Procedure. In: McRobbie, M.A., Slaney, J.K. (eds.) CADE 1996. LNCS, vol. 1104, pp. 583–597. Springer, Heidelberg (1996)
5. Brand, S., Gennari, R., de Rijke, M.: Constraint programming for modelling and solving modal satisfiability. In: Rossi, F. (ed.) CP 2003. LNCS, vol. 2833, pp. 795–800. Springer, Heidelberg (2003)
6. Brand, S., Gennari, R., de Rijke, M.: Constraint methods for modal satisfiability. In: Apt, K.R., Fages, F., Rossi, F., Szteredi, P., Váncza, J. (eds.) CSCLP 2003. LNCS (LNAI), vol. 3010, pp. 66–86. Springer, Heidelberg (2004)
7. Wallace, M.G., Novello, S., Schimpf, J.: ECLiPSe: A platform for constraint logic programming. ICL Systems Journal 12(1), 159–200 (1997)
8. Horrocks, I., Patel-Schneider, P.: FaCT and DLP. In: de Swart, H. (ed.) TABLEAUX 1998. LNCS (LNAI), vol. 1397, pp. 27–30. Springer, Heidelberg (1998)
9. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press, Cambridge (2001)
10. Stevenson, L.: Modal Satisfiability in a Constraint Logic Environment. University of South Africa, M.Sc dissertation (2008)
11. Balsiger, P., Heuerding, A., Schwendimann, S.: A benchmark method for the propositional modal logics K, KT, S4. Journal of Automated Reasoning 24(3), 297–317 (2000)
12. Jaeger, G., Balsiger, P., Heuerding, A., Schwendimann, S.: K, KT, S4 test data sets (retrieved, September 2007),
<http://www.iam.unibe.ch/~lwb/benchmarks/benchmarks.html>

Clustering with Feature Order Preferences

Jun Sun^{1,4}, Wenbo Zhao², Jiangwei Xue³, Zhiyong Shen^{1,4}, and Yidong Shen¹

¹ State Key Laboratory of Computer Science,

Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

² Department of Computer Science and Engineering,

University of California, San Diego, La Jolla, CA 92093, USA

³ Department of Mathematics, The Pennsylvania State University, USA

⁴ Graduate University, Chinese Academy of Sciences, Beijing 100049, China

{junsun,zyshen,ydshen}@ios.ac.cn, w3zhao@ucsd.edu, xue_j@math.psu.edu

Abstract. We propose a clustering algorithm that effectively utilizes feature order preferences, which have the form that feature s is more important than feature t . Our clustering formulation aims to incorporate feature order preferences into prototype-based clustering. The derived algorithm automatically learns distortion measures parameterized by feature weights which will respect the feature order preferences as much as possible. Our method allows the use of a broad range of distortion measures such as Bregman divergences. Moreover, even when generalized entropy is used in the regularization term, the subproblem of learning the feature weights is still a convex programming problem. Empirical results demonstrate the effectiveness and potential of our method.

1 Introduction

Data clustering is a fundamental technique of unsupervised learning that has been extensively studied for several decades [9] and yet is still an active area in machine learning. It aims to group objects, usually represented as data points in \mathbb{R}^d , into several clusters in a meaningful way. Many clustering techniques have emerged over the years and have been widely applied to many tasks, such as image segmentation [14], unsupervised document organization [7], grouping genes and proteins with similar functionality, and so on.

In supervised learning, labeled data are used to guide the learning procedure to obtain the most accurate model. Since no labels exist in unsupervised learning, it is very difficult to define precisely which clustering is the best one [8]. Consequently, heavy assumptions have to be made to measure the goodness of a clustering. However, when some extra domain knowledge is available, it becomes much easier to find a reasonable clustering for the task at hand. The problem of how to effectively incorporate domain knowledge into a clustering system has been an active topic in machine learning and data mining research. For example, the problem of clustering with instance-level knowledge in the form of pairwise constraints, namely, *must-link* and *cannot-link* constraints, has received a significant amount of attention in recent years [3,10].

In this paper, we propose a novel clustering algorithm which is able to take into account a new form of feature-level domain knowledge: *feature order preferences*. Feature order preferences have the form that feature s is more important than feature t . Obviously, feature order preferences are much easier to obtain than precise relative weights of the features. This work is inspired by the research on how to utilize instance-level order preferences in ranking and regression problems [6,17]. Our proposed clustering formulation aims to incorporate distance learning into prototype-based clustering, where the distortion measure is parameterized by the feature weights which will respect the feature order preferences as much as possible. Our clustering objective function allows the use of any Bregman divergence [2], which is a large family of distortion measures including squared Euclidean distance and Kullback-Leibler divergence. An important component in our algorithm is the subproblem of feature weight learning. Even when generalized entropy is used in the regularization term, this problem is still a convex programming problem, so efficient and effective algorithms exist [5]. Experimental results on several datasets demonstrate the effectiveness and potential of our proposed clustering algorithm with feature order preferences.

The rest of the paper is organized as follows. In Section 2, we formulate the clustering objective function in detail. The algorithm is derived in Section 3 and several extensions are discussed in Section 4. In Section 5, we evaluate the proposed method using several datasets. We conclude the paper in Section 6.

2 Model Formulation

Let $\mathcal{F} \subseteq \mathbb{R}^d$ denote the input space from which n data points, $\mathbf{x}_1, \dots, \mathbf{x}_n$, where $\mathbf{x}_i = [x_{i1}, \dots, x_{id}]^T$, are sampled. Given k and $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{F}$, the goal of clustering is to find a disjoint partitioning $\{\pi_c\}_{c=1}^k$ of the data where π_c is the c -th cluster. $n_c = |\pi_c|$ is the number of points in the c -th cluster.

We introduce some notational conventions first. Boldface lowercase letters, such as \mathbf{x} and \mathbf{y} , denote column vectors. The superscript T is used to denote the transpose of a vector. $\mathbf{1}_d$ denotes the d -dimensional column vector whose entries are all 1's. We use $\log(\cdot)$ to denote natural logarithm. \mathbb{R}_+ and \mathbb{R}_{++} denote the set of nonnegative and positive real numbers respectively. $\Delta_d = \{\mathbf{w} \in \mathbb{R}_+^d \mid \mathbf{w}^T \mathbf{1}_d = 1\}$ is the probability simplex. For any $\mathbf{w} = [w_1, \dots, w_d]^T \in \Delta_d$, the elements $\{w_j\}_{j=1}^d$ of \mathbf{w} are nonnegative and sum to one.

2.1 Clustering Objective with Feature Order Preferences

In practice, each feature of the data points may be of different importance with respect to the current clustering task. We use $\mathbf{w} = [w_1, \dots, w_d]^T \in \Delta_d$ to denote a feature weight vector so that w_j ($1 \leq j \leq d$) indicates the relative importance of feature j . For example, $\mathbf{w} = [\frac{1}{d}, \dots, \frac{1}{d}]^T$ is the uniform feature weighting where all the features are of equal importance.

In our clustering formulation, we assume that some domain knowledge in the form of *feature order preferences* will be given. A feature order preference

is defined by a tuple (s, t, δ) , which is interpreted as $w_s - w_t \geq \delta$. So $\delta > 0$ means that feature s is more important than feature t . “Features s and t are of approximately equal importance” can be encoded by a combination of $(s, t, -\epsilon)$ and $(t, s, -\epsilon)$ where ϵ is a small positive constant. In practice, optimal feature weights are often very difficult to obtain, but feature order preferences sometimes can be easy to acquire even when a domain expert has just some vague idea about the relative importance of the features. We assume that a set of m feature order preferences, denoted by \mathcal{P} where $|\mathcal{P}| = m$, will be given.

A key component in a typical clustering formulation is the dissimilarity (or similarity) between two points measured by a distortion function. Traditionally, without any domain knowledge, each dimension of the data points are often assumed to contribute equally to the distortion measure. Now that a set of feature order preferences is given by domain experts, we want to learn a distortion measure parameterized by the feature weight vector $\mathbf{w} \in \Delta_d$. Furthermore, we want to incorporate the process of distortion measure learning into prototype-based clustering to produce more accurate clusterings. Our clustering objective function consists of three terms which will be explained in detail as follows.

First, we want to minimize the *intra-cluster distortion* of the clusters $\{\pi_c\}_{c=1}^k$. Assume that there’s a cluster representative $\boldsymbol{\mu}_c \in \mathcal{F}$ for each cluster π_c . The distortion of cluster π_c is measured by $\sum_{\mathbf{x}_i \in \pi_c} D_{\mathbf{w}}(\mathbf{x}_i, \boldsymbol{\mu}_c)$ where $D_{\mathbf{w}}(\cdot, \cdot)$ is the distortion measure between two data points parameterized by the feature weight vector $\mathbf{w} \in \Delta_d$. The quality of the entire clustering $\{\pi_c\}_{c=1}^k$ is measured by the average distortion of all the k clusters, namely, $\frac{1}{n} \sum_{c=1}^k \sum_{\mathbf{x}_i \in \pi_c} D_{\mathbf{w}}(\mathbf{x}_i, \boldsymbol{\mu}_c)$.

Second, we want the weight vector \mathbf{w} to respect the feature order preferences in \mathcal{P} . Note that we treat the preferences as soft constraints rather than hard ones. A *penalty term* will be added to the objective function so that more violations of the preferences will lead to larger penalty. Besides, if the weights are consistent with all the preferences, then the penalty term will be zero. Therefore, we use a shifted hinge function [17] in the penalty term: for $p = (s, t, \delta) \in \mathcal{P}$, the penalty term for p is $\max(\delta - (w_s - w_t), 0)$.

Third, aside from the feature order preferences, we don’t want to make further unwarranted assumptions about the values of the weights. Therefore, another *regularization term*, $-\hat{H}(\mathbf{w})$, is added to the objective function to ensure that the weights are as uniform as possible. $\hat{H}(\cdot)$ is the generalized entropy which will be defined and discussed in Section 4.2. A key property of $\hat{H}(\cdot)$ is that the more uniform the weights w_j ($1 \leq j \leq d$) are, the larger the value of $\hat{H}(\mathbf{w})$ becomes. For the algorithm derived in the next section, we’ll use $\hat{H}(\mathbf{w}) = 1 - \mathbf{w}^T \mathbf{w}$ which will be referred to as ℓ_2 -*entropy*. Extensions will be discussed in Section 4.2.

By combining the three terms discussed above, we have the overall clustering goal, which is to minimize the following clustering objective function:

$$\frac{1}{n} \sum_{c=1}^k \sum_{\mathbf{x}_i \in \pi_c} D_{\mathbf{w}}(\mathbf{x}_i, \boldsymbol{\mu}_c) + \lambda_1 \sum_{(s, t, \delta) \in \mathcal{P}} \max(\delta - (w_s - w_t), 0) - \lambda_2 \hat{H}(\mathbf{w}) \quad (1)$$

where $\lambda_1, \lambda_2 \geq 0$ are pre-specified parameters.

After simple transformations using m auxiliary variables $\boldsymbol{\xi} = [\xi_p]$ where $p \in \mathcal{P}$, our overall clustering objective can be written as

$$\begin{aligned} \min_{\{\mathbf{w}, \boldsymbol{\xi}\}, \{\pi_c\}_{c=1}^k, \{\boldsymbol{\mu}_c\}_{c=1}^k} \quad & \frac{1}{n} \sum_{c=1}^k \sum_{\mathbf{x}_i \in \pi_c} D_{\mathbf{w}}(\mathbf{x}_i, \boldsymbol{\mu}_c) + \lambda_1 \sum_{p \in \mathcal{P}} \xi_p - \lambda_2 \hat{H}(\mathbf{w}) \\ \text{subject to} \quad & \mathbf{w} \in \Delta_d \\ & w_s - w_t \geq \delta - \xi_p \quad \text{for all } p = (s, t, \delta) \in \mathcal{P} \\ & \xi_p \geq 0 \quad \text{for all } p \in \mathcal{P} \end{aligned} \quad (2)$$

2.2 Parameterized Distortion Measures

Various distortion measures can be chosen for the clustering objective. Different distortion measures imply different assumptions about the underlying distribution of the data under consideration. Many useful distortion measures, such as squared Euclidean distance and KL divergence, belong to a broad class of distortion functions known as Bregman divergences [2] which is defined as follows.

Definition 1. Suppose $\phi : \mathcal{S} \mapsto \mathbb{R}$ is a strictly convex function defined on a convex set $\mathcal{S} \subseteq \mathbb{R}^b$ such that $\phi(\cdot)$ is differentiable on $\text{ri}(\mathcal{S})$, namely, the relative interior of \mathcal{S} . The Bregman divergence $d_\phi : \mathcal{S} \times \text{ri}(\mathcal{S}) \mapsto \mathbb{R}_+$ is defined as

$$d_\phi(\mathbf{z}_1, \mathbf{z}_2) = \phi(\mathbf{z}_1) - \phi(\mathbf{z}_2) - \langle \mathbf{z}_1 - \mathbf{z}_2, \nabla \phi(\mathbf{z}_2) \rangle \quad (3)$$

where $\nabla \phi$ is the gradient of ϕ .

Different $\phi(\cdot)$ will lead to diverse divergences. For example, if $\phi(z) = z^2$, then $d_\phi(z_1, z_2) = (z_1 - z_2)^2$ is the squared loss. If $\phi(z) = z \log(z) - z$, then $d_\phi(z_1, z_2) = z_1 \log(z_1/z_2) - (z_1 - z_2)$ is generalized I-divergence. Other Bregman divergences include Itakura-Saito distance, logistic loss, hinge loss and Mahalanobis distance. A key property of Bregman divergence is formally stated as follows [2].

Lemma 1. Suppose $\{\mathbf{z}_i\}_{i=1}^l \subset \mathcal{S} \subseteq \mathbb{R}^b$ and $\frac{1}{l} \sum_{i=1}^l \mathbf{z}_i \in \text{ri}(\mathcal{S})$. Given a Bregman divergence $d_\phi : \mathcal{S} \times \text{ri}(\mathcal{S}) \mapsto \mathbb{R}_+$, the problem

$$\min_{\mathbf{s} \in \text{ri}(\mathcal{S})} \sum_{i=1}^l d_\phi(\mathbf{z}_i, \mathbf{s}) \quad (4)$$

has a unique minimizer given by $\mathbf{s}^\dagger = \frac{1}{l} \sum_{i=1}^l \mathbf{z}_i$.

We use a parameterized version of Bregman divergences for $D_{\mathbf{w}}(\cdot, \cdot)$, specifically,

$$D_{\mathbf{w}}(\mathbf{x}_i, \boldsymbol{\mu}_c) = \sum_{j=1}^d \frac{w_j}{v_j} d_\phi(x_{ij}, \mu_{cj}) \quad (5)$$

where $\mathbf{v} = [v_1, \dots, v_d]^T \in \mathbb{R}^d$, which is used to scale the average within-cluster distortion in each dimension to $[0, 1]$, is defined as follows.

$$v_j = \min_u \frac{1}{n} \sum_{i=1}^n d_\phi(x_{ij}, u) \quad (6)$$

According to Lemma 1, $v_j = \frac{1}{n} \sum_{i=1}^n \mathbf{d}_\phi(x_{ij}, \bar{\mu}_j)$ where $\bar{\mu} = [\bar{\mu}_1, \dots, \bar{\mu}_d]^T = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ is the global mean of the data points.

3 Algorithm Derivation

In this section, we derive an efficient algorithm to optimize the clustering problem in Eq. (2). Combined with $\hat{H}(\mathbf{w}) = 1 - \mathbf{w}^T \mathbf{w}$ and Eq. (5), the clustering objective (2) is equivalent to the following one.

$$\begin{aligned} & \min_{\{\mathbf{w}, \boldsymbol{\xi}\}, \{\pi_c\}_{c=1}^k, \{\boldsymbol{\mu}_c\}_{c=1}^k} && \frac{1}{n} \sum_{c=1}^k \sum_{\mathbf{x}_i \in \pi_c} \sum_{j=1}^d \frac{w_j}{v_j} \mathbf{d}_\phi(x_{ij}, \mu_{cj}) + \lambda_1 \sum_{p \in \mathcal{P}} \xi_p + \lambda_2 \mathbf{w}^T \mathbf{w} \\ & \text{subject to} && \mathbf{w} \in \Delta_d \\ & && w_s - w_t \geq \delta - \xi_p \quad \text{for all } p = (s, t, \delta) \in \mathcal{P} \\ & && \xi_p \geq 0 \quad \text{for all } p \in \mathcal{P} \end{aligned} \quad (7)$$

This is the clustering objective we want to optimize in this section. Extensions will be discussed in the next section. Note that the regularization term $-\hat{H}(\mathbf{w})$ is very important to our formulation. For example, in the extreme case that no preferences are available ($m = 0$), if $\lambda_2 = 0$, then the feature with the smallest intra-cluster distortion will receive weight 1 and others get zero weight, which is obviously undesirable in practice.

In Eq. (7), there're 3 sets of unknown variables, namely, $\{\mathbf{w}, \boldsymbol{\xi}\}$, $\{\pi_c\}_{c=1}^k$ and $\{\boldsymbol{\mu}_c\}_{c=1}^k$. When two of them are fixed, the subproblem of computing the optimal values for the variables in the remaining set is easy to solve. Hence, problem (7) can be solved by iteratively updating $\{\mathbf{w}, \boldsymbol{\xi}\}$, $\{\pi_c\}_{c=1}^k$ and $\{\boldsymbol{\mu}_c\}_{c=1}^k$ so that the objective value gradually decreases. This approach can be thought of as a “block coordinate descent” method [4].

3.1 The Computation of $\{\pi_c\}_{c=1}^k$ for Given $\{\boldsymbol{\mu}_c\}_{c=1}^k$ and $\{\mathbf{w}, \boldsymbol{\xi}\}$

Given an existing set of cluster representatives $\{\boldsymbol{\mu}_c\}_{c=1}^k$ and $\{\mathbf{w}, \boldsymbol{\xi}\}$, computing the optimal clustering $\{\pi_c\}_{c=1}^k$ in problem (7) is equivalent to solving the following minimization problem

$$\min_{\{\pi_c\}_{c=1}^k} \sum_{c=1}^k \sum_{\mathbf{x}_i \in \pi_c} D_{\mathbf{w}}(\mathbf{x}_i, \boldsymbol{\mu}_c) \quad (8)$$

where $D_{\mathbf{w}}(\mathbf{x}_i, \boldsymbol{\mu}_c) = \sum_{j=1}^d \frac{w_j}{v_j} \mathbf{d}_\phi(x_{ij}, \mu_{cj})$. Therefore, each data point \mathbf{x}_i should be assigned to a cluster π_c so that $D_{\mathbf{w}}(\mathbf{x}_i, \boldsymbol{\mu}_c)$ is minimized (ties are resolved arbitrarily). After the cluster assignment, we obtain

$$\pi_c = \{\mathbf{x} \in \{\mathbf{x}_i\}_{i=1}^n \mid D_{\mathbf{w}}(\mathbf{x}, \boldsymbol{\mu}_c) \leq D_{\mathbf{w}}(\mathbf{x}, \boldsymbol{\mu}_l) \text{ for all } 1 \leq l \leq k\}. \quad (9)$$

3.2 The Computation of $\{\mu_c\}_{c=1}^k$ for Given $\{\pi_c\}_{c=1}^k$ and $\{w, \xi\}$

Given an existing clustering $\{\pi_c\}_{c=1}^k$ and $\{w, \xi\}$, computing the optimal cluster representatives $\{\mu_c\}_{c=1}^k$ in problem (7) is equivalent to solving the following minimization problem

$$\min_{\{\mu_c\}_{c=1}^k} \sum_{c=1}^k \sum_{\mathbf{x}_i \in \pi_c} \sum_{j=1}^d \frac{w_j}{v_j} d_\phi(x_{ij}, \mu_{cj}) = \sum_{c=1}^k \sum_{j=1}^d \frac{w_j}{v_j} g(\mu_{cj}) \quad (10)$$

where $g(\mu_{cj}) = \sum_{\mathbf{x}_i \in \pi_c} d_\phi(x_{ij}, \mu_{cj})$. Since $w_j, v_j \geq 0$, problem (10) is equivalent to minimizing $g(\mu_{cj})$ for each μ_{cj} where $1 \leq c \leq k$ and $1 \leq j \leq d$. According to Lemma 1, $\mu_{cj} = \frac{1}{n_c} \sum_{\mathbf{x}_i \in \pi_c} x_{ij}$ is the minimizer of $g(\mu_{cj})$. Therefore, the optimal value of problem (10) is achieved when $\mu_c = \frac{1}{n_c} \sum_{\mathbf{x}_i \in \pi_c} \mathbf{x}_i$.

3.3 The Computation of $\{w, \xi\}$ for Given $\{\pi_c\}_{c=1}^k$ and $\{\mu_c\}_{c=1}^k$

Given an existing clustering $\{\pi_c\}_{c=1}^k$ and cluster representatives $\{\mu_c\}_{c=1}^k$, computing the optimal $\{w, \xi\}$ in problem (7) is equivalent to solving the following minimization problem

$$\begin{aligned} \min_{\{w, \xi\}} \quad & \sum_{j=1}^d w_j a_j + \lambda_1 \sum_{p \in \mathcal{P}} \xi_p + \lambda_2 \mathbf{w}^T \mathbf{w} \\ \text{subject to} \quad & \mathbf{w} \in \Delta_d \\ & w_s - w_t \geq \delta - \xi_p \quad \text{for all } p = (s, t, \delta) \in \mathcal{P} \\ & \xi_p \geq 0 \quad \text{for all } p \in \mathcal{P} \end{aligned} \quad (11)$$

where $a_j = \frac{1}{n} \sum_{c=1}^k \sum_{\mathbf{x}_i \in \pi_c} \frac{1}{v_j} d_\phi(x_{ij}, \mu_{cj})$. Problem (11) is a (convex) quadratic programming problem [5] with $d + m$ variables, $d + 2m$ linear inequality constraints and 1 linear equality constraint. The globally optimal solution to this problem can be determined efficiently.

In [13], the criterion for selecting the optimal feature weighting in clustering makes the feature weights difficult to determine. In fact, they calculate the weights through an exhaustive search over a coarse grid on Δ_d . In practice, their method can only determine the approximately optimal values of a few weights. Instead, our problem formulation makes the subproblem of determining feature weights much easier to solve. Furthermore, the incorporation of domain knowledge such as feature order preferences becomes natural. Even with the generalized entropy discussed in Section 4.2, this subproblem for computing \mathbf{w} is still a convex programming problem in which any locally optimal solution is also globally optimal. There're very effective algorithms that can solve convex programs reliably and efficiently [5].

3.4 The Main Algorithm

The outline of our algorithm for Clustering with Feature order Preferences (CFP) is presented in Fig. 1. The “convergence” criterion is met when the change in

Algorithm: CFP ($\{\mathbf{x}_i\}_{i=1}^n$, k , \mathcal{P} , λ_1 , λ_2)

Input: Dataset $\{\mathbf{x}_i\}_{i=1}^n$, number of output clusters k , a set of feature order preferences \mathcal{P} , parameters λ_1 and λ_2 .

Output: Output clustering $\{\pi_c\}_{c=1}^k$.

Procedure:

1. Initialize k cluster representatives $\{\boldsymbol{\mu}_c\}_{c=1}^k$ and set $\mathbf{w} = [\frac{1}{d}, \dots, \frac{1}{d}]^T$;
2. **repeat**
 - 2a. **E-step** : Given $\{\boldsymbol{\mu}_c\}_{c=1}^k$ and $\{\mathbf{w}\}$, re-assign data points to clusters as in Section 3.1, thus obtaining $\{\pi_c\}_{c=1}^k$.
 - 2b. **M-step1**: Given $\{\pi_c\}_{c=1}^k$, re-calculate cluster representatives $\{\boldsymbol{\mu}_c\}_{c=1}^k$ as in Section 3.2;
 - 2c. **M-step2**: Given $\{\pi_c\}_{c=1}^k$ and $\{\boldsymbol{\mu}_c\}_{c=1}^k$, re-compute $\{\mathbf{w}, \boldsymbol{\xi}\}$ by solving the quadratic programming problem (11);
- until convergence;
3. **return** $\{\pi_c\}_{c=1}^k$;

Fig. 1. CFP algorithm

the clustering objective value between two successive iterations is less than some pre-specified threshold. In our experiments, our algorithm typically converges within less than 50 iterations.

As discussed in the previous subsections, the iterative updating procedure of CFP decreases the objective value in problem (7) after each iteration. Besides, the objective value is bounded below by zero. Therefore, the algorithm CFP converges to a locally optimal solution in a finite number of steps.

4 Extensions

4.1 Extension to Other Distortion Measures

Our clustering framework can be extended to other types of distortion (or similarity) measures including directional similarity functions such as cosine similarity and Pearson's correlation [12,3]. We use cosine similarity as an example to explain the extension. We define parameterized cosine distortion as follows.

$$D_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{w}}}{\|\mathbf{x}\|_{\mathbf{w}} \|\mathbf{y}\|_{\mathbf{w}}} \quad (12)$$

where $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{w}} = \sum_{j=1}^d w_j x_j y_j$ and $\|\mathbf{x}\|_{\mathbf{w}} = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbf{w}}}$. The updating procedure for $\boldsymbol{\mu}_c$ in M-step1 of Fig. 1 becomes

$$\boldsymbol{\mu}_c = \frac{\sum_{\mathbf{x}_i \in \pi_c} \mathbf{x}_i}{\left\| \sum_{\mathbf{x}_i \in \pi_c} \mathbf{x}_i \right\|_{\mathbf{w}}} \quad (13)$$

Note that if other distortion measures are used, the subproblem of computing the feature weights given $\{\pi_c\}_{c=1}^k$ and $\{\boldsymbol{\mu}_c\}_{c=1}^k$ may not be a convex programming problem any more, so locally optimal solution to the subproblem of feature weight learning may not be globally optimal.

4.2 Extension to Generalized Entropy

Generalized entropy measures the degree of uncertainty or impurity within a probability distribution. Here we only consider the discrete probability distribution represented by the probability simplex Δ_d . Each vector $\mathbf{w} \in \Delta_d$ corresponds to a probability distribution on a set of d elements, with w_j interpreted as the probability of the j -th element. A recently proposed definition of generalized entropy is formulated as follows [11].

Definition 2. *We define generalized entropy as a mapping*

$$\widehat{H} : \Delta_d \mapsto \mathbb{R}_+$$

that satisfies the following two criteria (symmetry and concavity):

1. *For any $\mathbf{w}_1 \in \Delta_d$, and any $\mathbf{w}_2 \in \Delta_d$ whose elements are a permutation of the elements of \mathbf{w}_1 , $\widehat{H}(\mathbf{w}_1) = \widehat{H}(\mathbf{w}_2)$.*
2. *$\widehat{H}(\cdot)$ is a concave function.*

A key intuition in this definition is that the more uniform the elements of $\mathbf{w} \in \Delta_d$ are, the larger the value of generalized entropy becomes. Here we give some specific examples of generalized entropy as follows:

1. $\widehat{H}(\mathbf{w}) = \sum_{i=1}^d -w_i \log(w_i)$, which is the celebrated *Shannon entropy*.
2. $\widehat{H}(\mathbf{w}) = 1 - \mathbf{w}^T \mathbf{w}$, which will be referred to as ℓ_2 -*entropy*.
3. $\widehat{H}(\mathbf{w}) = 2 - \sum_{i=1}^d |w_i - \frac{1}{d}|$, which will be referred to as ℓ_1 -*entropy*.
4. $\widehat{H}(\mathbf{w}) = 1 - \max_{1 \leq i \leq d} w_i$, which will be referred to as ℓ_∞ -*entropy*.

Many other entropies which are special cases of this definition have been proposed in the literature [11]. Since this definition of entropy is very general, our framework can lead to many instantiations. In our proposed algorithm in Section 3, we use ℓ_2 -entropy and so the optimization problem in M-step2 of Fig. 1 is a quadratic programming problem. When ℓ_1 -entropy is used, the optimization problem in M-step2 can be formulated as a linear programming problem. With distortion measure (5), whichever entropy is used, the optimization problem in M-step2 will always be a convex programming problem, since $\widehat{H}(\cdot)$ is a concave function and all the constraints in the clustering problem (2) are linear.

4.3 Extension to Multiple, Heterogeneous Feature Spaces

Our framework can be directly extended to multiple, heterogeneous feature spaces [13]. Each object is represented by a tuple of d component feature vectors, specifically, $\mathbf{x} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(d)})$ where $\mathbf{x}^{(j)}$ comes from the j -th feature space, which is associated with a weight w_j . Here, the order preference (s, t, δ) with $\delta > 0$ means that “feature space s is more important than feature space t ”. The distortion measure in Eq. (5) becomes

$$D_{\mathbf{w}}(\mathbf{x}_i, \boldsymbol{\mu}_c) = \sum_{j=1}^d \frac{w_j}{v_j} d_{\phi}(\mathbf{x}_i^{(j)}, \boldsymbol{\mu}_c^{(j)}) \quad (14)$$

The corresponding clustering algorithm can be easily derived.

Table 1. Summary of datasets

Dataset	<i>iris</i>	<i>optdigits</i>	<i>pageblocks</i>	<i>pendigits</i>	<i>usps</i>	<i>vowel</i>	<i>wdbc</i>
<i>n</i>	150	5620	5473	10992	9298	990	569
<i>d</i>	4	64	10	16	256	10	30
<i>k</i>	3	10	5	10	10	11	2

5 Experiments

In this section, we present an empirical evaluation of our clustering method on a number of datasets. First, we briefly introduce the basic information of the datasets. We use six datasets from the UCI machine learning repository [1] and a dataset *usps* which comprises all the examples from the USPS dataset¹. Table 1 summarizes the basic properties of the datasets. These datasets provide a good representation of different characteristics. Note that in all the experiments, the “true” number of clusters *k* is provided to the clustering algorithms.

5.1 Experimental Setting

In practice, feature order preferences would be provided by domain experts. However, for convenience, we generate simulated feature order preferences by using the ground truth class information in our experiments. We first calculate the *within-class distortion* for each dimension $1 \leq j \leq d$, which is defined as $\Theta_j = \frac{1}{v_j} \sum_{c=1}^k \sum_{x_i \text{ in class } c} d_\phi(x_{ij}, \mu_{cj})$ where μ_c is the centroid of class *c*. Then, for each dimension $1 \leq j \leq d$, we calculate the inverse within-class distortion $\Gamma_j = \frac{\sum_{l \neq j} \Theta_l}{\Theta_j}$. After that, we estimate the optimal feature weights by $\tilde{w}_j = \frac{\Gamma_j}{\sum_{l=1}^d \Gamma_l}$. The weight vector $\tilde{\mathbf{w}}$ is just a rough estimate of the optimal feature weighting. We randomly sample without replacement *m* pairs (s, t) of features with the constraint that \tilde{w}_s is among the $\lfloor \frac{d}{2} \rfloor$ largest weights and \tilde{w}_t is among the $\lfloor \frac{d}{2} \rfloor$ smallest weights. Then our simulated feature order preference is $(s, t, \tilde{w}_s - \tilde{w}_t)$.

In our experiments, we use $\hat{H}(\mathbf{w}) = 1 - \mathbf{w}^T \mathbf{w}$. Besides, we set the parameters $\lambda_1 = \frac{d}{m}$ and $\lambda_2 = d$. The reason for this is that we want the three terms in Eq. (1) to contribute equally to the objective value. Since the first term is scaled to $[0, 1]$ due to \mathbf{v} , we want the other terms to be around 1. As the average weight of each feature is $\frac{1}{d}$, the second term is approximately less than $\frac{m}{d}$ and so λ_1 is set to $\frac{d}{m}$. The minimum value of $\mathbf{w}^T \mathbf{w}$ is $\frac{1}{d}$ corresponding to the uniform weighting. We want \mathbf{w} to be as uniform as possible, so λ_1 is set to d .

We compare the performance of our algorithm with Bregman hard clustering [2] which is referred to as BCUs. For the initialization, we randomly select *k* points as the cluster representatives and uniform weighting as the initial \mathbf{w} . We set $\phi(x) = x^2$ for BCUs and CFP. Unless otherwise stated, each performance result is the average of 20 runs of the whole algorithm. Besides, we use the MOSEK package² to solve the quadratic programming subproblem in M-Step2.

¹ <ftp://ftp.kyb.tuebingen.mpg.de/pub/bs/data/>

² <http://www.mosek.com>

Table 2. Clustering results on the *iris* dataset. “*C.M.*” denotes confusion matrices. CFP(m) is our proposed algorithm with m feature order preferences. This is the clustering result of only a single run of the algorithms.

Algo.	BClus			CFP(1)			CFP(2)			CFP(3)			CFP(4)		
<i>C.M.</i>	50	0	0												
	0	39	14	0	42	10	0	49	6	0	48	4	0	48	4
	0	11	36	0	8	40	0	1	44	0	2	46	0	2	46
<i>NMI</i>	0.6595			↗0.7145			↗0.8572			↗0.8642			→0.8642		
<i>Acc</i>	0.8333			↗0.8800			↗0.9533			↗0.9600			→0.9600		

5.2 Evaluation Criteria

Given class labels, we adopt two external validity measures, Normalized Mutual Information (*NMI*) [15] and Clustering Accuracy (*Acc*) [16], as our criteria.

Given a clustering \mathcal{C} and the “true” partitioning \mathcal{B} (class labels). The number of clusters in \mathcal{C} and classes in \mathcal{B} are both k . Suppose n_i is the number of objects in the i -th cluster, n'_j is the number of objects in the j -th class and n_{ij} is the number of objects which are in both the i -th cluster and j -th class. *NMI* between \mathcal{C} and \mathcal{B} is calculated as follows [15]:

$$NMI(\mathcal{C}, \mathcal{B}) = \frac{\sum_{i=1}^k \sum_{j=1}^k n_{ij} \log \frac{n \cdot n_{ij}}{n_i \cdot n_j}}{\sqrt{\sum_{i=1}^k n_i \log \frac{n_i}{n} \sum_{j=1}^k n'_j \log \frac{n'_j}{n}}}. \quad (15)$$

Clustering Accuracy (*Acc*) builds a one-to-one correspondence between the clusters and the classes. Suppose the permutation function $\text{Map}(\cdot) : \{i\}_{i=1}^k \mapsto \{j\}_{j=1}^k$ maps each cluster index to a class index, i.e., $\text{Map}(i)$ is the class index that corresponds to the i -th cluster. *Acc* between \mathcal{C} and \mathcal{B} is calculated as follows:

$$Acc(\mathcal{C}, \mathcal{B}) = \frac{\max \left(\sum_{i=1}^k n_{i, \text{Map}(i)} \right)}{n} \quad (16)$$

Larger values of *NMI* and *Acc* indicate better clustering performance.

5.3 Comparison of Clustering Performance

In this section, we compare the performance of the algorithms. We test our proposed clustering algorithm with various numbers of feature order preferences.

First, we consider a small dataset *iris*. Table 2 shows the confusion matrices, *NMI* and *Acc* values obtained by BClus and CFP on the *iris* dataset. The arrows at the left side of *NMI* and *Acc* values indicate whether the value increases (\nearrow) or remains unchanged (\rightarrow), compared with the algorithm in the previous column. Note that this is the clustering result of only a single run of the algorithms. It can be observed from Table 2 that CFP produces better clustering results as the number of feature order preferences increases.

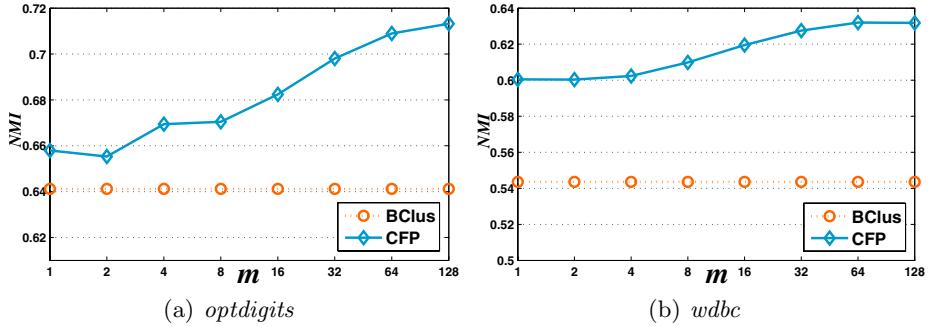


Fig. 2. The clustering performance (NMI) on datasets *optdigits* and *wdbc*

Table 3. Experimental results on all the datasets. Both NMI and Acc results are provided here. CFP(m) is our proposed algorithm with m feature order preferences. The results produced by CFP when $m = \lfloor \frac{d}{4} \rfloor, \lfloor \frac{d}{2} \rfloor, d$ are shown.

	NMI				Acc			
	BClus	CFP($\lfloor \frac{d}{4} \rfloor$)	CFP($\lfloor \frac{d}{2} \rfloor$)	CFP(d)	BClus	CFP($\lfloor \frac{d}{4} \rfloor$)	CFP($\lfloor \frac{d}{2} \rfloor$)	CFP(d)
<i>iris</i>	0.6547	0.7559	0.8290	0.8642	0.8280	0.8987	0.9353	0.9600
<i>optdigits</i>	0.6412	0.6824	0.6980	0.7090	0.6390	0.6919	0.7137	0.7252
<i>pageblocks</i>	0.1103	0.1422	0.1554	0.1865	0.4599	0.5720	0.6177	0.6868
<i>pendigits</i>	0.6957	0.6973	0.6983	0.7043	0.6868	0.6962	0.6896	0.7179
<i>usps</i>	0.5796	0.5846	0.5872	0.5898	0.5948	0.6013	0.6063	0.6120
<i>vowel</i>	0.3636	0.3960	0.4123	0.4254	0.3145	0.3316	0.3563	0.3627
<i>wdbc</i>	0.5436	0.6111	0.6172	0.6256	0.9077	0.9221	0.9236	0.9251

Then we compare the clustering results of the algorithms on datasets *optdigits* and *wdbc*. The results (NMI) with various values of m are shown in Fig. 2. It can be observed that our algorithm CFP consistently outperforms BClus, even with a small number of feature order preferences. With an increasing m , CFP generally produces increasing NMI values. Therefore, larger gains in clustering performance can be obtained with more feature order preferences.

The clustering results on all the datasets are shown in Table 3. As for CFP, results with $m = \lfloor \frac{d}{4} \rfloor, \lfloor \frac{d}{2} \rfloor, d$ are shown. As can be seen from Table 3, CFP generally produces better clustering results than BClus, and more feature order preferences often lead to better performance. For each dataset, the best result is often achieved by CFP(d). The results demonstrate that CFP effectively improves clustering quality when some feature order preferences are available.

6 Conclusions

In this paper, we propose a clustering algorithm that takes into account feature order preferences. Our clustering objective integrates feature weight learning

into prototype-based clustering. Experimental results show that our proposed algorithm effectively improves clustering quality. Potential future work includes investigating automatic parameter selection method for λ_1 and λ_2 instead of using pre-specified values. Besides, it might be fruitful to incorporate feature order preferences into the process of cluster initialization.

Acknowledgments. This work is supported in part by NSFC grants 60673103 and 60721061.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. Irvine, CA: University of California, Department of Information and Computer Science (2007)
2. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman divergences. *Journal of Machine Learning Research* 6, 1705–1749 (2005)
3. Basu, S., Bilenko, M., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2004)
4. Bertsekas, D.P.: *Nonlinear Programming*, 2nd edn. Athena Scientific (1999)
5. Boyd, S., Vandenberghe, V.: *Convex Optimization*. Cambridge University Press, Cambridge (2004)
6. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: *Proceedings of the Twenty-Second International Conference on Machine Learning* (2005)
7. Domeniconi, C., Gunopulos, D., Ma, S., Yan, B., Al-Razgan, M., Papadopoulos, D.: Locally adaptive metrics for clustering high dimensional data. *Data Mining and Knowledge Discovery* 14(1), 63–97 (2007)
8. Estivill-Castro, V.: Why so many clustering algorithms — a position paper. *SIGKDD Explorations* 4(1), 65–75 (2002)
9. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. *ACM Computing Surveys* 31(3), 264–323 (1999)
10. Kulis, B., Basu, S., Dhillon, I.S., Mooney, R.J.: Semi-supervised graph clustering: a kernel approach. In: *Proceedings of the Twenty-Second International Conference on Machine Learning* (2005)
11. Luo, P., Zhan, G., He, Q., Shi, Z., Lü, K.: On defining partition entropy by inequalities. *IEEE Transactions on Information Theory* 53(9), 3233–3239 (2007)
12. Mardia, K.V., Jupp, P.E.: *Directional Statistics*, 2nd edn. John Wiley and Sons Ltd., Chichester (2000)
13. Modha, D.S., Spangler, W.S.: Feature weighting in k -means clustering. *Machine Learning* 52(3), 217–237 (2003)
14. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8), 888–905 (2000)
15. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3, 583–617 (2002)
16. Wu, M., Schölkopf, B.: A local learning approach for clustering. In: *Advances in Neural Information Processing Systems*, vol. 19 (2006)
17. Zhu, X., Goldberg, A.: Kernel regression with order preferences. In: *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence* (2007)

Distributed Memory Bounded Path Search Algorithms for Pervasive Computing Environments

Anoj Ramasamy Sundar and Colin Keng-Yan Tan

Department of Computer Science, School of Computing,
National University of Singapore
21 Lower Kent Ridge Road, Singapore 119077
`{anojrama, ctank}@comp.nus.edu.sg`

Abstract. A pervasive computing environment consists typically of a large heterogeneous collection of networked devices which can acquire and reason on context information. Embedded devices are used extensively in pervasive environments but they face some key challenges. Common path searching algorithms like A* Search can have exponential number of node expansions. In this paper, we describe a special variant of this problem called Multiple Objective Path Search (MOPS) and propose a memory bounded solution to implement it in a pervasive environment. Experimental results show that an efficient path with 40-60 times less node expansions can be obtained with the proposed solution.

Keywords: Pervasive computing, context-aware services, network services, context reasoning, path finding, memory bounded search.

1 Introduction

The key objective of pervasive computing [1,2,3] is to provide ‘anytime’ anywhere computing where users can interact with the sentient environment and where we decouple users from devices and view applications as just entities that perform task on behalf of users.

Embedded systems play an inherent role in providing seamless connectivity and interactivity to end-users in the pervasive environment. They provide efficient support for acquiring, discovering and interpreting (reasoning) user context information. But the major obstacle for the use of embedded devices in such environments is their limited memory resource. Unlike desktop machines, they generally do not have a typical hard disk to store their application files [4, 5]. Remote Memory Paging techniques have been proposed [6] where unused memory of the remote workstations, personal computers etc in the smart space of a pervasive computing environment are used to store the unused processes/pages of the smaller device, incurring large memory latencies as pages are fetched across a network.

In this paper, we focus on a special graph search problem – **Multiple objective Path search**. The Gas Station Problem [7] discusses several routing problems that generalize shortest path and the Traveling Salesman Problem. It proposes polynomial

time solutions for finding the best path that also minimizes gas cost involved in the path traversal. Other similar work include [8] and [9], where a Hopfield neural network is used in an $n \times n$ configuration to solve the traveling salesman problem, with the provision that a columnar competitive model with a winner-takes-all learning rule is used to eliminate invalid states and the need to find a suitable starting point in the search. In [10] Jin and Wang consider a multi-agent framework of communicating agents that co-operate to perform distributed multi-objective optimization. In our paper we consider a multi-objective path search problem for pervasive embedded devices that have limited memory resources and are hence unable to maintain complete information at any point in time.

2 Multiple Objective Path Search

The objective of a_Multiple Objective Path Search (MOPS) is to find the best path from a Start node to End in a given graph/Map via a set of nodes (V_i) satisfying a set of predefined constraints. In this paper the two main constraints are to complete an intermediate dynamic objective, and minimizing the path distance cost. Our map is represented as an undirected, possibly cycle graph with nodes V and edges E, where each edge bears a cost $d(v_i, v_j)$ representing the distance between nodes v_i and v_j . We consider a **Shopping Problem** where a shopper has to purchase a list of items in his checklist, starting from some node v_1 to an end node v_n in the shortest distance.

Some well known graph traversal algorithms that can be used to solve the MOPS are Breadth-first search, Depth-first search, Dijkstra's and Best-first A* search [13]. There have been constant improvements to these algorithms, over the past. The more interesting variations are the memory bounded versions like the Iterative-deepening-A* (IDA*) and the simplified memory-bounded A* (SMA*) [14].

A* guarantees [13] to find the shortest path, as long as the heuristic estimate, $h(n)$, is admissible that is, it is never greater than the true remaining distance to the goal. The SMA* algorithm is optimal and complete if enough memory is available; otherwise, it returns the best solution that can be found using the given memory. A global variable, *MAX nodes* is used to represent the maximum number of nodes that can be fit into memory. If there is no memory left and the algorithm still needs to generate a successor, the most unpromising node is dropped off from the open list.

SMA* seems to be the likely choice as we are working in a memory constraint domain. Our algorithm should produce the best available result when the device runs out of memory, which SMA* guarantees. In the rest of the paper, A* and the modified SMA* will be used as our base algorithm for discussion.

Preliminary results indicated that SMA* with MAX nodes equal to Map length had a 37% lesser node expansions, 49% smaller open queue size with a 98% success rate and 3 % longer shortest path length compared to A*.

A couple of more interesting observations were made. Firstly, In order to best utilize the available memory, a small improvement can be performed in the SMA* algorithm. The next-successor function can be made to return successors which have the shortest straight line distance to the destination, first. This improvement helps to select better successors first and add the lesser good ones into the open queue in later iterations. This further increases the accuracy when memory is scarce. From here on, we will use this technique in all our SMA derivatives, in the later sections.

Secondly, a major chunk of memory is utilized in storing and manipulating the graph/map. We further have to maintain open and closed queues whose sizes also depend on the size of the map. In real world applications, graph sizes can be huge and it would be unreasonable to compute local searches on a single embedded device. Hence, there is an inherent need for a distributed alternative where the bigger task is divided and assigned to the underlying reasoners, which we propose in the next section.

3 Distributed Multiple Objective Path Search (DMOPS)

From our observations in the previous section it is clear that in order to use embedded reasoners for this task, we need to reduce the problem size. Let us first define our pervasive environment as follows:

- Terrain Map/Graph contains walkable and unwalkable nodes
- Some of the walkable nodes, satisfy a part of the given objective
- The entire path search is done using embedded reasoners distributed in this environment over a P2P network
- Each reasoner is assigned a small chunk of the bigger map, depending on the resources available in them for computation
- Each node that can satisfy one or more of the objective, itself is a reasoner and decides which is the next node to visit from there

The above approach divides the bigger task and distributes it to the reasoners according to the resources they can afford. Each reasoner heuristically decides the next best node to visit, which satisfies both our constraints – Objective completion and shortest path. To simplify the problem, we also assume that Objective completion has a higher priority i.e. finishing it in the least hops. We also assume that sub-components of the initial objective are readily available in a "lot" of nodes, hence backtracking is disallowed. Our prime focus is to develop a memory efficient solution, hence we assume underlying network infrastructure is infinitely fast and has zero communication cost. We will propose a solution later to avoid this assumption.

3.1 Proposed Algorithm

We start off with an objective list (ObjList) and the start (A) and end points (B). Opt [A, B, ObjList] refers to the optimal path from A to B, satisfying all the constraints in the Objective List and within the memory and resource limits of our reasoners. From our assumptions, we can break the problem into two phases – Optimal path to complete the objective (keeping B in mind) and optimal path to finally reach B.

Also, as we are assuming all the nodes that satisfy the objectives are themselves reasoners, the Objective List is updated as we visit each reasoner node that satisfies a part of the original objective. Each reasoner that we visit has two primary tasks. First, search for the next best reasoner for forwarding the sub-problem and then compute the shortest Path to reach it.

The final path generated would be the sum of all the shortest paths computed by each visited node. Let R_n be the last node visited to complete all objectives. We can represent the problem as follows:

$$\text{Opt}[\mathbf{A}, \mathbf{B}, \text{ObjList}] = \text{Opt1}[\mathbf{A}, \mathbf{R}_n, \text{ObjList}] + \text{Opt2}[\mathbf{R}_n, \mathbf{B}, \text{emptyList}]$$

$$\quad \quad \quad \{\text{Phase1}\} \quad + \quad \{\text{Phase2}\} \quad (1)$$

By Dynamic programming, we can represent Phase1 as follows

$$\text{Opt1}[\mathbf{A}, \mathbf{R}_n, \text{ObjList}] = \text{ShortestPath}[\mathbf{A}, \mathbf{R}_1] + \text{Opt1}[\mathbf{R}_1, \mathbf{R}_n, \text{ObjList}_1]$$

$$\begin{aligned} &= \text{ShortestPath}[\mathbf{A}, \mathbf{R}_1] + \dots + \text{ShortestPath}[\mathbf{R}_{i-1}, \mathbf{R}_i] \\ &\quad + \text{Opt1}[\mathbf{R}_i, \mathbf{R}_n, \text{ObjList}_i] \end{aligned} \quad (2)$$

Where, $\text{ObjList}_i \leq \text{ObjList}_{i-1}$.

The selection of the set of reasoners (R_i) that satisfy the Optimality function is an exponential problem. In order to get an efficient and scalable solution, we use a heuristics function (F_1) to prune our search space and decide which reasoner (R_i) to visit next. R_i in turn will compute the next node that needs to be visited from the information available to it. Finally, once all the objectives in the Objective List are satisfied, we compute the best path to reach the end(B) using a heuristic function (F_2).

We can represent Phase2 as follows

$$\text{Opt2}[\mathbf{R}_n, \mathbf{B}, \text{emptyList}]$$

$$= \text{ShortestPath}[\mathbf{R}_n, \mathbf{R}_{n+1}, \text{emptyList}] + \text{Opt2}[\mathbf{R}_{n+1}, \mathbf{B}, \text{emptyList}] \quad (3)$$

Finally,

$$\text{Opt}[\mathbf{A}, \mathbf{B}, \text{ObjList}] = \sum \text{ShortestPath}_{\text{Phase1}} + \sum \text{ShortestPath}_{\text{Phase2}} \quad (4)$$

This ***reason and route*** method has two main advantages. Firstly, the big map is distributed among low level reasoners. Each reasoner sees only a small section of the map for instance, around its geographical location only. Secondly, as the map size is small, $\text{ShortestPath}[R_{i-1}, R_i]$ can now be computed easily by R_{i-1} using A* search or SMA* search as discussed in preceding sections.

The Optimal path algorithm discussed here relies a lot on the functions that select the next node to visit. In phase 1, the goal is to complete the objective as fast as possible (distance or time) and in phase 2; goal is to reach the destination (B).

Let us discuss the next node selection process for both these phases.

3.2 Heuristics Functions for Phase 1 (F_1 Objective Completion)

In this paper, to chose which node to expand next (say from R_i), we use the following heuristics in order of their decreasing priority

H1 = (% of Objective satisfied by R_{i+1} and R_{i+1} 's best child)

H2 = straight line distance from R_i to R_{i+1}

H3 = straight line distance from R_{i+1} to B

As Objective completion has a higher priority, H2 cost and then H3 cost would be used in case of a tie. The obvious advantage of this method is that it reduces network traffic compared to a greedy search, which might be exponential in nature. Here a maximum total of 8x8 nodes are searched. It is easier to implement and the exponential search issue can be avoided. The major disadvantage of this method is that the

path might not be optimum. There is no way to guarantee that local success will ensure an overall success in the future. This might still be useful if the sub-objectives are readily satisfied by a large number of nodes on the map. This is discussed in detail later in the implementation of the shopping problem later.

3.3 Heuristics Functions for phase 2 (F_2 Path Completion)

Let R_n be the last reasoner visited after which the Objective list is marked completed. Now the goal is to find the shortest path to B. We will use a simple straight line distance heuristics. We use this Hcost to determine which child node to visit next:

$$H = (\text{straight line distance from } R_n \text{ to } R_{n+1}) + (\text{straight line distance from } R_{n+1} \text{ to } B)$$

In order to test the performance of our algorithm and the various cost functions, the shopping problem was implemented. The next section describes the problem and our implementation along with the experiment results.

4 Application: Shopping in a Pervasive Environment

Consider this situation. A shopper gets down at a train station A with a list of items to buy. He wants to find out the shortest path to purchase all the items and reach the next station B to catch a train back home.

The two objectives that we need to satisfy

- Buy all the items in the List (ObjList)
- Reach the destination (B)

This Shopping problem can be solved using modified MOPS. One main difference to MOPS is that the rank of future nodes decreases as we visit more nodes in present. Reason being, once an item is bought at shop S_1 , the shopper won't buy it again from shop S_2 . In this paper we solve this common problem using a variant of DMOPS.

We describe the pervasive environment as follows

- Map contains walkable, un-walkable, Reasoner and Shop nodes (V)
- Reasoners (R) distributed on the environment help compute the best path
- Some Shop Nodes could sell the required items (satisfy part of objective)
- All Shop Nodes are also Reasoners and will direct the shopper to the next shop/Reasoner node once he finished purchasing there
- Items in the List are common items and are readily available in the shop Nodes scattered in the Map

Shopper's interaction with the environment is done via a server. A centralised server takes in the shopper's request (position, List) and forwards his request to the nearest reasoner and from there on the shopper gets directions from the reasoners/Shop Nodes which he visits. This server is also responsible for initialising and distributing the big map to all the reasoners depending on their geographical location (*Figure 2*).

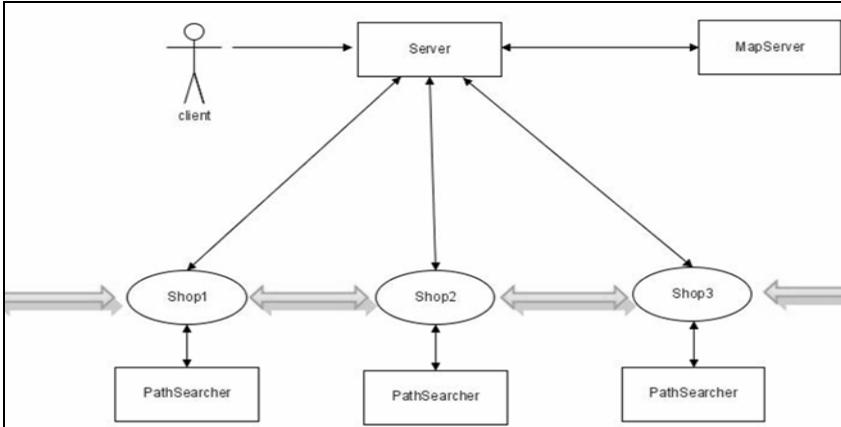


Fig. 1. A Shopper interacting with the pervasive environment

As discussed in the previous section, we used the Nearest Best Neighbour heuristics to determine which reasoner to forward the remaining List and visit next. As the items are assumed to be readily available in the Map, backtracking is not allowed. This was done to save path cost involved in reversing the direction. The Nearest Best Neighbour was preferred as the experiment was conducted with a large map and the items were readily available.

For the Shortest Path component in DMOPS, we used A* and SMA* and compared their performance. We compared their results against a simplified MOPS implementation that uses A* search. This represents a non-distributed, centralised MOPS algorithm.

In this Simplified MOPS implementation, we assumed that path computation takes place on a single reasoner, infinite memory and computation resources are available and the entire Map is available for path search. Objective completion phase still works similar to the DMOPS phase1 search seen earlier, to avoid an exponential search i.e. a maximum immediate search area is defined to locate the best node and forward. Hence, our simplified MOPS can be described as

$$\text{Opt[A,B, ObjList]} = \sum \text{ShortestPath}_{[\text{Phase1}]} + \text{ShortestPath}[R_n, B] \quad (5)$$

4.1 Experiment Parameters

Most of the experiment parameters were randomly generated. A map size of 100 x 100 nodes was defined with random number of obstacles, shop nodes and simple reasoners distributed randomly in the terrain. Reasoners were assigned map sizes of 11 x 11 and 21 x 21. For the SMA* versions, the MAX nodes was set as 5 and 10 respectively. Each reasoner stores a map, with its own location as the centre. Hence, using an 11x11 map for instance, a reasoner can compute the shortest path to a maximum of 5 steps in any particular direction from itself. From our preliminary results in Section 2.2, it is clear that a MAX nodes \geq Map Width gives us a decent success % with a significant reduction in space. A random list of items was generated and assigned to the Shop Nodes.

4.2 Evaluation Criteria

Path Length and Nodes expanded were used as our main parameters for comparison.

- 1. Path finding success rate:** This represents the % of queries whose path will be successfully computed by the algorithm
- 2. Path Length:** This represents the length of the final path generated
- 3. Nodes Expanded:** This gives us an estimate of computation and memory resource spent on computing the final path

4.3 Results

A total of 310 sets of test cases were generated and the following 6 instances were compared.

Table 1. Algorithms and summary of parameters used

Algorithm	Parameters	Representation
MOPS A* search	Search area of 5	OA5
MOPS A* search	Search area of 10	OA10
DMOPS A* search	Map Size 11 x 11	DOA5
DMOPS A* search	Map Size 21 x 21	DOA10
DMOPS SMA* search	MAX nodes 5 (map 11x11)	DOSMA5
DMOPS SMA* search	MAX nodes 10 (map 21x21)	DOSMA10

Instances of similar search area and map size were compared to evaluate the performance of the algorithm.

4.3.1 Path Finding Success Rate

In more than 40% cases, DOA5 and DOSMA5 failed to compute any valid path. DOA10 and DOSMA10 performed reasonably well, failing in about 5% cases.

Table 2. Path finding Success rate comparison

	OA5	OA10	DOA5	DOA10	DOSMA5	DOSMA10
Paths found	265	309	183	294	184	295
Success%	85.21	99.36	58.84	94.53	59.16	94.86

Failure of the DOA5 and DOSMA5 can be accounted to the random distribution of reasoners on the map. Both these algorithms work on a “reason and forward” (*Section 3.1*) mechanism. If density of reasoners distributed on a map is skewed, both these algorithms will fail. In cases of uniformly distributed maps, their success % should increase. See *Section 5: Further Work*.

4.3.2 Path Length

We compared the length of path produced by OA against the two distributed algorithms. For the smaller map versions of these algorithms (DOA5 and DOSMA5), this result was generated from cases where a valid path < 150% length of OA5 was returned. This was done to prune out all bad maps from the results. Map classification is discussed later (*Section 5: Further Work*).

This result clearly shows that having the entire map (as in OA instances) does help in generating the better path.

Table 3. Comparing performance of DOA, DOSMA against OA

DOA10	DOSMA10	DOA5	DOSMA5
9.89% more	9.89 % more	16.59% more	17.64 % more

Both DOA10 and DOSMA10 produce 10% longer paths compared to the OA10. DOA5 and DOSMA5 continued to fare badly due to their small map sizes. Both these algorithms have to deduce the bigger picture using a very small image of the map that they posses. A non-uniform distribution of reasoners on a map further degrades their performance.

4.3.3 Nodes Expanded

We compared the number of nodes expanded in computing the final path. Memory and computation resource needed to compute the path can be directly associated with the nodes expanded during the whole process.

Table 4. Comparing total nodes expanded in DOA5 and DOSMA5 against OA5

DOA5 vs. OA5	DOSMA5 vs. OA5	DOSMA5 vs. DOA5
4.26 times less	45.87 times less	10.77 times less

As represented in *Table 4*, the distributed SMA* version computed the path using 46 times less total node expansions compared to the OA* version. It also used 11 times less resource compared to the DOA*. This is attributed to the MAX nodes limit set in SMA*, which constantly prunes bad nodes from the open list and reduces the number of nodes expanded.

Table 5. Comparing total nodes expanded in DOA10 and DOSMA10 against OA10

DOA10 vs. OA10	DOSMA10 vs. OA10	DOSMA10 vs. DOA10
3.10 times less	57.87 times less	18.67 times less

From *Table 5*, the bigger map versions of distributed SMA* version computed the path using 58 times less node expansions compared to the OA* version. It also used 19 times less resource compared to the DOA*. This again is attributed to the MAX nodes limit set in SMA*, which constantly prunes bad nodes from the open list and reduces the number of nodes expanded.

4.4 Summary

After comparing the three algorithms, we can conclude that the size of maps allocated to reasoners play a key role in increasing the accuracy and path quality of the computed result. The DOSMA algorithm consistently performed well in all three criteria when adequate map sizes were allocated to each reasoner. Hence, if adequately sized maps are assigned, DOSMA can generate reasonably good paths at a fraction of computation and memory cost.

5 Further Work

Sometimes, even assigning a large map to individual reasoners might not be enough. Take the example shown in *Figure 3*. Once all objectives are completed, OA10 reaches the destination following the best available path as it has the entire map to compute the A* shortest path. However in DOA10, each shop node relies on its neighbour reasoner nodes to generate the best path. Non-uniform and skewed maps can lead to a really inefficient path. Hence, there is a need to classify maps and prune bad neighbours which lead to inefficient paths. A simple classification scheme could be used to identify the number of neighbours for a given reasoner in each direction so that any query to that reasoner can be forwarded in any of the 4 directions.

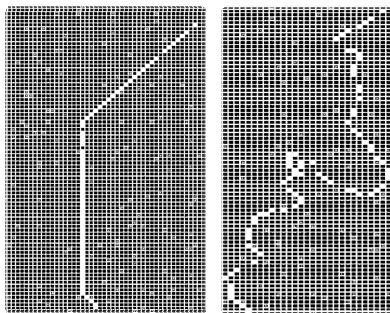


Fig. 2. Path OA10 and DOA10

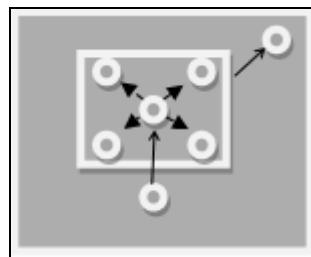


Fig. 3. Map stored by a reasoner (center)

For instance, using the test cases in *Section 4.1*, map classification was done for the individual reasoners. Then, for each of their maps, additional reasoners were added in order to forward queries in all the four directions as in *Figure 4*. With this uniform distribution, path efficiency of DOSMA5 almost matches with OA5. This is an important correction as in a practical pervasive environment, new forwarding reasoners can be added to improve the path efficiency. Small embedded devices like motes are

cheap when manufactured in a large number. They can act as efficient forwarding devices.

In solving the shopping problem, we assumed that all the reasoning nodes are similar devices in terms of their computation power and memory resource. In a practical real world pervasive environment, different types of devices co-exist. For instance, the shop reasoners could be powerful desktop computers whereas the reasoners on the path could be small devices like motes. Hence a good map allocation scheme at the centralized server can help in ensuring a more uniform reasoning environment.

Due to our initial assumption of an infinitely fast underlying network infrastructure and zero communication cost, we have completely ignored the communication cost. This may not be a valid assumption in real world environments. A path learning algorithm can be used that tries to compute paths for new requests from historical search results. To implement it, we first store path searches in a Knowledge Base (KB) and use it to derive sub paths for new search queries. This method will help reduce a lot of network queries as components of the original problem will be pre-computed (by centralised servers) as efficiently as possible and then requests would be sent to the embedded reasoners to compute the remaining path. More work needs to be done in this context as path formulations in some cases could be more complex. Another key challenge in this implementation is the update cost. For instance, when a shop stops selling a particular item or simply stops functioning, all the paths stored in the KB involving the node has to be marked invalid. But in the long run, when the KB is comprehensive and a path update protocol is in place, the path learning algorithm could improve the query processing and computation time considerably.

6 Conclusion

In this paper, we focused on one special graph search problem – Multiple objective Path search and discussed the challenges involved in extending this problem into the embedded domain.

The “*reason and route*” framework proposed here, enables us to perform a large and complex problem in an embedded domain. It also gives an acceptable solution when memory and computation resources are scarce. The results can be improved further with some enhancements as discussed in *Section 5: Further Work*.

In conclusion, the results obtained from the method are highly encouraging. It enables us to perform a large problem by dividing the work to a distributed embedded domain. It also gives an acceptable solution with the fraction of original resource. The major obstacle for the use of embedded devices in such environments earlier was their limited memory resource. With this method, more complex problems can be extended into the embedded domain and hence, enable a seamless pervasive experience in the future.

References

1. Weiser, M.: The Computer for the 21st Century. *Scientific American* (1992)
2. Hopper, A.: Sentient Computing: The Royal Society Clifford Paterson Lecture. AT&T Laboratories Cambridge, Technical Report (1999)

3. Nelson, G.: Context-Aware and Location Systems: PhD Thesis. Cambridge University Computer Lab, UK (1998)
4. Markatos, E.P., Dramitinos, G.: Implementation of a Reliable Memory Pager. In: Proc. USENIX Tech. Conf., San Diego, California (1996)
5. Boling, D.: Minimizing the Memory Footprint of Your Windows CE based Program (1998), <http://www.microsoft.com/msj/0598/memory.htm>
6. Sathiaseelan, A., Radzik, T.: Using Remote Memory Paging for Handheld Devices in a Pervasive Computing Environment. King's College London
7. Khuller, S., Malekian, A., Mestre, J.: To Fill or not to Fill: The Gas Station Problem. In: European Symp. on Algorithms (ESA), Spain (2007)
8. Teoh, E.J., Tang, H.J., Tan, K.C.: A Columnar Competitive Model with Simulated Annealing for Solving Combinatorial Optimization Problems. International Joint Conference on Neural Networks (2006)
9. Tang, H.J., Tan, K.C., Yi, Z.: A Columnar Competitive Model for Solving Combinatorial Optimization Problems. IEEE Transactions on Neural Networks 15, 1568–1574 (2004)
10. Jin, F.J., Wang, L.: A Distributed Multiobjective Optimal Algorithm based on MAS. In: Proceedings of the First International Conference on Innovative Computing, Information and Control (2006)
11. Roman, M., Campbell, R.H.: Gaia: Enabling Active Spaces. In: Proceedings of the 9th ACM SIGOPS European Workshop, Kolding, Denmark (2000)
12. Tao, Y., Papdias, D., Shen, Q.: Continuous Nearest Neighbor Search. In: Proc. 28th Very Large Data Bases Conference (2002)
13. Russell, S., Norvig, N.: Artificial Intelligence: A Modern Approach. Prentice Hall, Englewood Cliffs (1995)
14. Russell, S.: Efficient Memory-Bounded Search Methods. In: Proc. 10th European Conf. On Artificial Intelligence, pp. 1–5. Wiley, Chichester
15. Sheth, A., Ramakrishnan, C.: Semantic (Web) Technolog In Action: Ontology Driven Information Systems for Search, Integration and Analysis. In: IEEE Data Engineering Bulletin, Special issue on Making the Semantic Web Real (2003)
16. Kindberg, T., et al.: People, Places, Things: Web Presence for the Real World. In: Proceedings of the third WMCSA (2000)

Using Cost Distributions to Guide Weight Decay in Local Search for SAT

John Thornton and Duc Nghia Pham

SAFE Program, Queensland Research Lab, NICTA and
Institute for Integrated and Intelligent Systems, Griffith University, QLD, Australia
`{john.thornton,duc-nghia.pham}@nicta.com.au`

Abstract. Although clause weighting local search algorithms have produced some of the best results on a range of challenging satisfiability (SAT) benchmarks, this performance is dependent on the careful hand-tuning of sensitive parameters. When such hand-tuning is not possible, clause weighting algorithms are generally outperformed by self-tuning WalkSAT-based algorithms such as AdaptNovelty⁺ and AdaptG²WSAT.

In this paper we investigate tuning the weight decay parameter of two clause weighting algorithms using the statistical properties of cost distributions that are dynamically accumulated as the search progresses. This method selects a parameter setting both according to the speed of descent in the cost space and according to the shape of the accumulated cost distribution, where we take the shape to be a predictor of future performance. In a wide ranging empirical study we show that this automated approach to parameter tuning can outperform the default settings for two state-of-the-art algorithms that employ clause weighting (PAWS and gNovelty⁺). We also show that these self-tuning algorithms are competitive with three of the best-known self-tuning SAT local search techniques: RSAPS, AdaptNovelty⁺ and AdaptG²WSAT.

Keywords: Local search, clause weighting, automated parameter tuning, satisfiability.

1 Introduction

One way to categorize the currently best performing satisfiability (SAT) local search algorithms is according to the method used to escape local minima. Firstly, there are those approaches that use randomized decision strategies, such as the WalkSAT family of algorithms [1] and the more recent G²WSAT algorithms [2]. Secondly, there are those that use weights to penalize local minima features, such as DLM [3], SAPS [4], GLSSAT [5], and PAWS [6]. To date, clause weighting algorithms have outperformed WalkSAT on many of the standard benchmark problems, but only when careful parameter tuning is allowed. Conversely, WalkSAT-based algorithms have consistently dominated the recent SAT competitions¹, where the hand-tuning of parameters is not possible because the details of the competition problems are not known in advance. This restriction more accurately reflects real-world situations where an answer is required as quickly as possible, rather

¹ <http://www.satcompetition.org/>

than needing to know how quickly we *could* have found an answer if we had known the optimal parameter settings in advance.

One of the main reasons for the success of WalkSAT algorithms is that their performance is primarily influenced by the value of a single *noise* parameter (noise controls the degree of randomness in each flip decision). While the best setting for this parameter varies widely from problem to problem, it can be effectively adapted during the search using a simple heuristic that measures the degree of search stagnation [7]. A similar heuristic was developed for the SAPS clause weighting algorithm [4] but this remains uncompetitive with the best WalkSAT techniques [8].

Of the other current clause weighting algorithms, the pure additive weighting scheme (PAWS) is probably the best candidate for the development of a parameter adapting heuristic because its performance depends on a single *MaxInc* parameter which controls the rate of decay of the clause weights [6]. However, despite considerable effort, no effective online method for adapting *MaxInc* has been discovered. More recently, the gNovelty⁺ algorithm has combined a WalkSAT-based heuristic with a clause weighting mechanism to produce the 2007 SAT competition random satisfiable category winner [8]. gNovelty⁺ uses the WalkSAT heuristic to adapt noise and has a second parameter to control the rate of clause weight decay. Although it is known that gNovelty⁺'s performance is affected by the setting of this second parameter, to date no method has been proposed to adapt it automatically.

In this paper we investigate an online method to automatically adapt clause weight decay using information extracted from distributions of false clause counts recorded at each flip. By accumulating distributions at various parameter settings, we can predict the best setting as the search continues. We have applied this method to both PAWS and gNovelty⁺ in order to improve their average case performance in comparison to the standard default weight decay settings.

In the remainder of the paper we provide more detail on existing approaches to parameter tuning and then provide an in-depth description of our new approach and how it has been incorporated into PAWS and gNovelty⁺. Using an empirical study, we compare the performance of the new approach against PAWS and gNovelty⁺, and against AdaptG²WSAT, AdaptNovelty⁺ and RSAPS. Finally we discuss the results and present our conclusions.

2 Parameter Tuning and Performance Prediction

The literature on predicting algorithm performance can be divided along several axes depending on whether the prediction is based on: (i) an off-line training phase (e.g. [9]) or purely on feedback obtained while solving online instances (e.g. [10])(ii) measuring problem features (e.g. [11]) or on measuring an algorithm's past runtime performance (e.g. [12]) (iii) deciding between a portfolio of algorithms (e.g. [13]) or determining the parameter settings of a single algorithm (e.g. [14]) (iv) predicting performance on a per instance basis (e.g. [14]) or on a per distribution basis (e.g. [11]) (v) using a high or low complexity prediction model (giving rise to so-called "low knowledge" approaches [15]).

While these distinctions cover a broad range of potential methods, there is considerable overlap across axes and between the kinds of machine learning technique that are effective. In relation to the current research, we are interested in using a “low knowledge” approach based on online feedback about runtime performance to predict parameter settings. Our aim is to improve the average performance of a candidate parametrized SAT algorithm in situations where we are only allowed a single run on a problem instance and where the problem characteristics are not known in advance.

Online SAT Implementations: The best known *online* self-tuning local search SAT algorithms have not explicitly predicted performance, but instead have exploited measures of search stagnation. For example, AdaptNovelty⁺ (the self-tuning version of Novelty⁺) adapts its noise parameter according to whether an improvement is observed in the overall best cost after a fixed number of flips, i.e.: if no improvement occurs, the value of the noise parameter is increased, thereby increasing the probability that non-greedy moves are accepted; otherwise, if a new minimum cost solution is found, then the noise value is immediately decreased. Hoos [7] demonstrated experimentally that this adaptive mechanism is effective both with Novelty⁺ and other WalkSAT variants. The same basic mechanism was also used to adapt the probability that clause weights will be multiplicatively reduced in the RSAPS algorithm [4] and, in earlier work, stagnation measures were used in reactive tabu search [16]. Again, referring to the SAT 2007 competition, the best individual local search algorithms (gNovelty⁺ and AdaptG²WSAT) both employ the AdaptNovelty⁺ self-tuning mechanism - making this the state-of-the-art for online adaptation (at least within the SAT local search community). However, in relation to the current research, stagnation measures have not proved effective for tuning the clause weight decay parameter of any clause weighting algorithm except RSAPS (and RSAPS is known to be uncompetitive with gNovelty⁺ or AdaptG²WSAT [8]).

Local Search Invariants: In their influential paper, McAllester et al. [17] reported an invariant statistical relationship in the cost distributions for a range of SAT algorithms on a selection of planning, graph colouring and hard random 3-SAT problems. Here, and in the rest of the paper, we define a *local search cost distribution* to be the distribution of the count of false clauses recorded at each flip during a sequence of local search steps. McAllester et al.’s invariant relationship was calculated as the mean divided by the standard deviation of the local search cost distribution recorded over a large number of runs for a particular algorithm, on both single instances and on groups of similar instances. We term this measure the *range* statistic, where the range specifies the distance of the mean search cost from the origin in standard deviation units. McAllester et al. found that the optimal setting for the noise parameters they were investigating consistently occurred at a value 10% greater than the noise value that minimized the range measure. They consequently conjectured that range could be an effective online *and* off-line performance predictor for tuning local search parameters.

“Low Knowledge” Algorithm Control: Outside the SAT domain there are several other approaches that attempt to predict performance on the basis of search behaviour (e.g. [10,12]). The most interesting of these for current purposes is “low knowledge” algorithm control which uses reinforcement learning to dynamically allocate runtime slices to different algorithms as the search progresses. Each algorithm has a weight that is updated after a given number of iterations according to a reinforcement learning formula that takes the cost improvement per second as input, i.e. the faster an algorithm is descending in the cost space, the greater the increase in its weight and the larger the time slice it is allowed in the next series of iterations. In empirical tests, this method was able to exceed the average performance of the pure algorithms on which it was based. The “low knowledge” approach shows that an online application that only examines the current best cost can effectively allocate time slices between competing algorithms. The main differences between this work and our own, are that we need to decide between different parameter settings rather than different algorithms, and that we cannot accumulate knowledge between runs on different instances.

3 Tuning PAWS Online

The preceding analysis has yielded two promising avenues for further investigation: i) exploiting McAllester et al.’s range statistic as an online guide to parameter performance and ii) using the “low knowledge” approach of dividing online runtime resources according to the speed of descent of different parameter settings in the cost space. The challenge is that both these approaches have previously required multiple runs on the same problem or on distributions of similar problems before they can act as reliable guides. If we limit ourselves to looking at a single run, the stochastic nature of local search means we get little better than vague hints of which direction to move. In addition, the problem of tuning the PAWS *MaxInc* parameter is complicated by the large range of possible values (from 4 to 500) and the sensitivity of the parameter to small changes (for example, see [18]). To counteract these issues we developed two strategies. Firstly, we looked at changing the way that PAWS reduces weight to create a more robust parameter with a smaller range of possible values. And secondly, we broadened our view of the cost data available during each problem run to include the *shape* of the local search cost distribution.

The *MaxThres* Parameter: The effect of periodically reducing clause weights is to reduce the total number of clauses that have weight. An analysis of the runtime behaviour of PAWS on individual problems shows that each *MaxInc* setting yields a fairly stable mean number of false clauses. We therefore experimented with reducing weight whenever the number of false clauses exceeds a given threshold. This produced a new *MaxThres* parameter that exhibited similar performance to *MaxInc* except that it proved more robust to small changes in its value.

The operation of *MaxThres* is detailed in the pseudocode of the *UpdateClauseWeights* function in Algorithm 1. This function is called whenever PAWS

decides it has reached a local minimum and differs from the original PAWS only at lines 5 and 6. Previously, PAWS reduced weight at line 5 if $IncCounter > MaxInc$ and omitted the while loop of line 6. Now the $MaxThres$ parameter causes weight to be reduced when the number of weighted clauses ($|\mathcal{W}|$) and the number of false clauses ($|\mathcal{F}|$) both exceed $MaxThres$ and only after at least $MinInc$ consecutive weight increase phases have been completed ($MinInc$ is fixed at 3). In addition, the while loop at line 6 ensures that each weight reduction phase reduces $|\mathcal{W}|$ to a value less than $MaxThres$ (this step becomes necessary when evaluating the performance of different $MaxThres$ values during the same run).

The new $MaxThres$ parameter alters the behaviour of PAWS by tending to reduce weight relatively less frequently when there are fewer false clauses and relatively more frequently when there are more false clauses. This change produced small differences in performance in comparison with the original PAWS, but on the average the two approaches proved very similar. The main advantage of $MaxThres$ is that we can (on average) obtain equivalent performance with the original PAWS while also reducing the set of parameter values from $\{4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 15\ 20\ 25\ 30\ 35\ 40\ 50\ 75\ 100\ 125\ 250\ \infty\}$ for $MaxInc$ to $\{25\ 50\ 75\ 125\ 250\ 500\ 750\ \infty\}$ for $MaxThres$.

Algorithm 1: UpdateClauseWeights

```

Input:  $\mathcal{F} \leftarrow$  the set of currently false clauses;  $\mathcal{W} \leftarrow$  the set of currently weighted clauses;
Output: updated membership of  $\mathcal{W}$ ; updated clause weights for  $\mathcal{F} \cup \mathcal{W}$ ;
1 for each  $c_i \in \mathcal{F}$  do
2    $Weight(c_i) \leftarrow Weight(c_i) + 1$ ;
3   if  $Weight(c_i) = 2$  then  $\mathcal{W} \leftarrow \mathcal{W} \cup c_i$ ;
4  $IncCounter \leftarrow IncCounter + 1$ ;
5 if  $|\mathcal{W}| > MaxThres$  and  $|\mathcal{F}| > MaxThres$  and  $IncCounter > MinInc$  then
6   while  $|\mathcal{W}| > MaxThres$  do
7     for each  $c_i \in \mathcal{W}$  do
8        $Weight(c_i) \leftarrow Weight(c_i) - 1$ ;
9       if  $Weight(c_i) = 1$  then  $\mathcal{W} \leftarrow \mathcal{W} - c_i$ ;
10   $IncCounter \leftarrow 0$ ;

```

Local Search Cost Distribution Shape: Given a single run, the information available to select a parameter setting is scarce and highly variable. Our first approach to ameliorate this situation was to set up a version of PAWS that progressively accumulates local search cost distributions for each parameter setting and allows the user to change parameter settings and graphically visualize the different cost distributions. From these observations it became clearer what *shape* of cost distribution was most often associated with the best parameter setting. The general rule of thumb is: select the distribution with the smallest mean, given the distribution has a roughly normal shape. As a result of extensive preliminary experimentation, we decided to use skewness and kurtosis statistics as an additional guide for parameter setting. Skewness measures the degree of symmetry of a distribution (where a zero value indicates perfect symmetry) and is calculated as follows:

$$\frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{\sigma} \right)^3$$

In the case of measuring the skewness of a local search cost distribution for a particular *MaxThres* value, n would be the number of flips taken at the selected *MaxThres* value, x_i the number of false clauses observed at flip i , and \bar{x} and σ the mean and standard deviation respectively of the distribution of x_i 's.

Kurtosis measures the degree of “peakedness” of a distribution, where a higher value indicates a sharper peak with flatter tails (in comparison to a standard normal distribution). We calculated kurtosis as follows:

$$\frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{\sigma} \right)^4 - 3$$

Simulated Annealing: Having identified a few promising measures, we required a method to control the parameter value selection process during the lifetime of a single run. Inspired by the “low knowledge” approach, we used two interleaved searches on the same problem, one with a high *MaxThres* setting (750) and the other with a good low default setting (75), as follows: each search starts with its own copy of the same problem initialisation, and then pursues its own separate search trajectory; the two search procedures then compete for processor time according to a simulated annealing (SA) [19] schedule shown in Algorithm 2.

Algorithm 2: DecideUpperOrLowerSetting

```

Input: lowerThres  $\leftarrow$  lower MaxThres setting; upperThres  $\leftarrow$  upper MaxThres setting;
temp  $\leftarrow$  1024; step  $\leftarrow$  400; tempStep  $\leftarrow$  initial steps allocated to upper setting;
upperStep  $\leftarrow$  current steps allocated to upper setting;
lowerStep  $\leftarrow$  current steps allocated to lower setting;
1 if lowerStep  $<$  upperStep then tempStep  $\leftarrow$  tempStep + lowerStep;
2 else tempStep  $\leftarrow$  tempStep + upperStep;
3 while tempStep  $\geq$  step do
4   temp  $\leftarrow$  temp  $\div$  2;
5   step  $\leftarrow$  step  $\times$  2;
6   cost  $\leftarrow$  CostDifference(lowerThres, upperThres);
7   diff  $\leftarrow$  AbsoluteValue(cost);
8   uphillProb  $\leftarrow$   $50e^{-\left(\frac{|diff|}{temp}\right)}$ ;
9   if probability  $\leq$  uphillProb then
10    | if cost  $\geq$  0 then return lowerThres; else return upperThres;
11   else
12    | if cost  $\leq$  0 then return lowerThres; else return upperThres;

```

Here, SA is used to control a decision model that begins by randomly allocating time slices to the two search procedures and then, as the temperature decreases, biases decisions more and more towards respecting the CostDifference measure defined in Algorithm 3. This measure quantifies our notion of local search cost distribution shape. An important point to note here is that all statistics for each distribution (i.e. the mean, standard deviation, skewness and kurtosis) are reset each time the distribution reaches a solution that improves on the previously best minimum cost (for that distribution). This eliminates the initial high variance phase of the search and avoids the distorting effects of outlying cost values. In addition, we ignore the sign of the skewness and kurtosis measures, taking their absolute value only (see *AbsSkew* and *AbsKurt* in Algorithm 3).

The DecideUpperOrLowerSetting procedure controls the PAWS *MaxThres* setting for the first 50,000 flips of the combined search trajectories. During this

Algorithm 3: CostDifference($thres1, thres2$)

```

1  $minCostRatio \leftarrow 10 \times (MinCost(thres1) \div (MinCost(thres1) + MinCost(thres2)))$ ;
2  $rangeRatio \leftarrow 10 \times (Range(thres1) \div (Range(thres1) + Range(thres2)))$ ;
3  $skewRatio \leftarrow 10 \times (AbsSkew(thres1) \div (AbsSkew(thres1) + AbsSkew(thres2)))$ ;
4  $kurtRatio \leftarrow 10 \times (AbsKurt(thres1) \div (AbsKurt(thres1) + AbsKurt(thres2)))$ ;
5 return  $100 - ((9 \times rangeRatio) + (7 \times minCostRatio) + (2 \times (skewRatio + kurtRatio)))$ ;

```

phase, the new PAWS will behave much like its predecessor (with $MaxInc$ set to 10), except that it will “waste” a certain number of flips exploring the non-optimal distribution. Such exploration will help if the best setting is in the upper distribution, but otherwise it will degrade the relative performance.

Binary Search: After the 50,000 flip threshold, both the upper and lower search trajectories are allowed to explore other $MaxThres$ settings within a lower range of {25 50 75 125} and an upper range of {250 500 750 ∞ }. This procedure takes the form of a binary search, such that after every search step of 100 flips (where the value of $MaxThres$ remains fixed) the DecideUpperOrLowerSetting function determines which half of the parameter space will be used next. Then we use the DecideSetting and FindBestCost functions to further subdivide the parameter space into a single setting. For example, if DecideUpperOrLowerSetting selects lower, then we will call:

DecideSetting(FindBestCost(25, 50), FindBestCost(75, 125))

Otherwise we will call:

DecideSetting(FindBestCost(250, 500), FindBestCost(750, ∞))

The DecideSetting function follows the simulated annealing approach of DecideUpperOrLowerSetting with two changes to reflect the finer grain of the decision. Firstly, the annealing function has a consistently higher probability of returning an uphill move, replacing line 8 of Algorithm 2 with:

$$uphillProb \leftarrow 30e^{-(\frac{diff}{temp})} + 20$$

Secondly, the annealing schedule is only reduced according to the number of steps taken since the last minimum cost was discovered for each distribution, replacing lines 1-2 from Algorithm 2 with:

```

if ( $lowerStep < upperStep$ ) then  $tempStep \leftarrow lowerStep$ ;
else  $tempStep \leftarrow upperStep$ ;

```

Finally, we limit the parameter search space on problems with more than 50,000 clauses to only consider 25 or 50 in the lower distribution (the upper distribution parameter range remains unchanged). This reflects empirical observations showing that larger problems tend to have smaller optimal $MaxThres$ settings.

4 Experimental Study

To test the new self-tuning version of PAWS (which we term iPAWS), we selected a range of *random* problems from the SAT competition satisfiable benchmarks and a range of *structured* problems from the SATLIB library. The SAT competition problems are to give an idea of potential performance in the competition’s

satisfiable random category (this is the category where SAT local search algorithms have consistently outperformed complete search techniques), while the other problems allow comparison with the existing SAT literature. Firstly, to form the SATLIB structured benchmark, we took two “flat” graph colouring problems (flat200-median and hard²), two blocksworld planning problems (bw_large.c and d), two logistics planning problems (logistics.c and d), three all interval series problems (ais10, 12 and 14), five hard quasigroup problems (qg1-08, qg2-08, qg5-11, qg6-09, qg7-13), five 16-bit parity learning problems (par16-1-c to par16-5-c), four large graph colouring problems (g125.17, g125.18, g250.15, g250.29), four circuit synthesis formula problems (2bitadd_11, 2bitadd_12, 3bitadd_31, 3bitadd_32) and four Beijing scheduling problems (enddr2-1, enddr2-8, ewddr2-1 and ewddr2-8). Secondly, to form the random benchmark set, we randomly selected 12 k3-SAT instances, 12 k5-SAT instances and 10 k7-SAT instances from the large satisfiable random problems used in the 2005 SAT competition.

To obtain a measure of the improvement in iPAWS over the original PAWS we included a manually tuned PAWS (PAWS(t)) with *MaxInc* optimized for each problem category and a default-valued PAWS (PAWS(d)) with *MaxInc* fixed at 10. We also included the best-known self-tuning SAT local search algorithms: AdaptNovelty⁺ [7], AdaptG²WSAT [2] and RSAPS [4]. Both AdaptNovelty⁺ and AdaptG²WSAT are included for their class leading performance in the recent SAT competitions and RSAPS is included to compare iPAWS with another clause weighting adaptive algorithm. Finally, we included gNovelty⁺, the winner of the random satisfiable category of the 2007 SAT competition, and arguably the best general purpose SAT local search solver currently available [8].

iNovelty⁺: As discussed in the introduction, gNovelty⁺ also performs clause weighting and has a parameter that can control the rate of clause weight decay. Experimental observations have shown that gNovelty⁺’s performance can be significantly enhanced by treating this parameter as a binary switch that either allows weight to accumulate without reduction, or turns off clause weighting altogether [8]. To investigate the applicability of the iPAWS weight decay heuristic to other clause weighting algorithms, we decided to partially implement the iPAWS tuning process into gNovelty⁺. This involved controlling the binary decision about whether or not to accumulate weight in gNovelty⁺ in the same way that iPAWS decides whether to use an upper or lower setting of *MaxThres*. More specifically, gNovelty⁺ was adapted to match iPAWS so that it runs two separate searches on same problem starting point, one with weight accumulation turned on and the other with it turned off. Each search then competes for processor time using the DecideUpperOrLowerSetting procedure defined in Algorithm 2. We term this new weight-tuning version of gNovelty⁺ as iNovelty⁺.

4.1 Results

To obtain an overall measure of performance, we adapted the SAT competition scoring metric (see <http://www.satcompetition.org/2007/rules07.html>) to suit

² We take all designations of median and hard problems from [4].

the situation of our allowing each algorithm 100 runs on each instance with a 600 second cutoff for each run. This required us to divide the SAT competition *solution purse* up in proportion to the number of successful runs for solver s on problem p , as follows:

$$\text{solutionAward}(p, s) \leftarrow \frac{1000 \times \text{successCount}(p, s)}{\sum_{i=1}^n \text{successCount}(p, i)}$$

where *successCount* counts the number of runs where a solution has been found within the standard timeout of 600 seconds. Similarly, the SAT competition *speed purse* is divided as follows:

$$\text{speedAward}(p, s) = \frac{1000 \times \text{speedFactor}(p, s)}{\sum_{i=1}^n \text{speedFactor}(p, i)}$$

where *speedFactor*(p, s) calculates the sum, in seconds, of: $600 \div (1 + \text{solution time})$ for each successful run of algorithm s on problem p . Finally, the SAT competition *series purse* allocates 3000 points for a problem series containing 5 or more problems (i.e. the parity, quasigroup, k3-SAT, k5-SAT and k7-SAT series), otherwise it allocates 1000 points. Here the points are divided equally amongst all algorithms that can find any solution to any problem within a given series. The final measure for each solver is then calculated as the sum of the three purses.

Structured Benchmarks: Table 1 shows the scores for the structured benchmark problems using the SAT competition measure, both overall and on a per series basis. These results clearly show the gNovelty⁺ variants outperform the other solvers regardless of the parameter tuning methods employed. Within the gNovelty⁺ solvers, iNovelty⁺ produced a slight improvement over gNovelty^{+(d)} but this almost entirely rests on iNovelty⁺'s ability to beat the default algorithm on the parity series. There are other signs of improvement on the bw_large and large graph colouring problems, but on the remaining problems the improvements have not outweighed the overhead of performing two interleaved searches.

The iPAWS results show that the self-tuning heuristic has produced a more sizable improvement in comparison to the default PAWS(d), raising the overall score from 4,656 to 7,940. In particular, iPAWS has performed better than PAWS(d) on all the structured series except bw_large (where performance is still similar), having the largest improvement on the parity problems. Comparing iPAWS with the optimally tuned PAWS(t) shows that iPAWS sometimes gains an advantage from being able to adapt to individual problems (particularly in the bitadd and qg series). Overall, however, the prior tuning of PAWS(t) outperforms iPAWS, particularly on the bw_large problems.

Looking at PAWS in comparison with AdaptG²WSAT, AdaptNovelty and RSAPS shows that the improvement on PAWS(d) is enough to move PAWS from being the worst performing solver, to being the best (excepting gNovelty⁺). While the difference between iPAWS and AdaptG²WSAT is slight, the relative effect of the iPAWS self-tuning method is impressive, especially considering the extra effort needed to perform an interleaved search on two problem instantiations, of which one is necessarily exploring the worse half of the parameter space.

Random Benchmarks: Table 2 shows the results for the random benchmark problems (again using the SAT competition metric). Here we have reproduced

Table 1. Scoring of solvers' performance on structured problems

Solver	bitadd	ais	bw _{large}	e*ddr	flat	g*	logistics	par16	qg	Total
gNovelty ⁺ (t)	1, 477.2	916.1	820.7	1, 333.9	594.7	1, 199.4	571.9	1, 855.3	2, 911.5	11, 680.6
gNovelty ⁺ (d)	1, 477.2	916.1	609.1	1, 333.9	594.7	875.4	571.9	466.6	2, 447.9	9, 292.7
iNovelty ⁺	1, 440.0	830.3	677.8	1, 156.0	530.2	1, 072.7	566.4	1, 296.4	1, 941.0	9, 510.7
PAWS (t)	380.1	822.5	835.8	1, 166.9	569.2	1, 298.2	569.2	2, 685.4	805.6	9, 132.8
PAWS (d)	380.1	694.8	324.4	377.5	528.2	578.7	535.3	407.4	829.7	4, 656.1
iPAWS	640.2	824.3	310.8	971.9	537.1	1, 021.8	561.9	2, 011.2	1, 060.6	7, 940.0
AdaptG ² WSAT0	1, 283.5	631.6	546.6	787.6	576.8	1, 252.3	496.6	1, 529.4	749.4	7, 853.7
AdaptNovelty ⁺	1, 348.9	505.1	491.7	708.5	507.7	1, 216.1	555.9	1, 392.4	743.3	7, 469.6
RSAPS	572.8	859.3	383.2	1, 163.9	561.4	485.4	570.8	1, 355.9	1, 511.0	7, 463.8

Table 2. Scoring of solvers' performance on random problems

Series	gNovelty ⁺	PAWS (d)	iPAWS	AdaptG ² WSAT0	AdaptNovelty ⁺	RSAPS
k3-SAT	7, 468.6	11, 364.3	1, 476.5	2, 618.7	4, 071.9	0.0
k5-SAT	5, 892.8	742.3	7, 052.4	7, 174.3	5, 633.6	504.6
k7-SAT	4, 424.2	2, 966.6	3, 600.6	4, 973.8	5, 335.8	1, 699.0
Total	17, 785.7	15, 073.1	12, 129.5	14, 766.9	15, 041.3	2, 203.6

the parameter settings for PAWS ($MaxInc = 10$) and gNovelty⁺ that were used in the original SAT competitions. In this case, gNovelty⁺ used a heuristic that sets its clause weight decay parameter to 1.0 for all 5-SAT and 7-SAT problems (turning clause weighting off) and to 0.4 for all 3-SAT problems (reducing weight after an increase with probability 0.4) [8]. This heuristic means gNovelty⁺ is no longer using a fixed parameter value for these problems and makes a direct comparison with iNovelty⁺ unfair. We therefore propose that iNovelty⁺ use the gNovelty⁺ heuristic whenever it detects uniform 3, 5, and 7-SAT problems (making it equivalent to gNovelty⁺ on these problems).

The results again show gNovelty⁺ outperforming all other solvers by a sizeable margin, with PAWS(d) and AdaptNovelty⁺ coming in a close second and third respectively, followed by AdaptG²WSAT, iPAWS and finally RSAPS (after a large gap). Looking in more detail, we can see the relatively poor performance of iPAWS is due to the 3-SAT problems, otherwise it outperforms PAWS(d) on both the 5-SAT and 7-SAT series. If we allow the gNovelty⁺ heuristic to be legitimate, then a similar heuristic applied to iPAWS could switch it to perform a default PAWS search on all 3-SAT problems. In this case, iPAWS would defeat both PAWS(d) and gNovelty⁺ on the random benchmarks.

5 Discussion and Conclusions

The primary aim of this paper was to develop an effective online method to tune the PAWS weight decay parameter within single runs on single problems. The secondary aim was to explore the use of this method within another clause

weighting algorithm. The results have shown the new iPAWS heuristic to be effective across a range of structured problems and across two of the three classes of uniform random SAT problems. We have also proposed a simple heuristic to improve iPAWS on the 3-SAT benchmarks.

However, our attempt to extend the iPAWS approach to gNovelty⁺ did not produce such dramatic improvements. We propose two reasons for this. Firstly, gNovelty⁺ already uses an adaptive online mechanism to tune the Novelty noise parameter. It may be that the two adaptive mechanisms do not interact to good effect. A promising area of future research would therefore be to use an iPAWS approach to simultaneously tune the gNovelty⁺ noise and weight decay parameters. This also suggests using the local search cost distributions to tune the noise parameters of other WalkSAT algorithms, to test if this would be more effective than using the current stagnation measures. Secondly, the relatively small improvement of iNovelty⁺ over gNovelty^{+(d)} could also be explained by iNovelty⁺ not having the advantage of being able to do a more refined search of the parameter space. As the gNovelty⁺ weight decay parameter has so far proved relatively insensitive to intermediary settings between 0 and 1 - with the exception of preferring a 0.4 setting on 3-SAT problems and a 0.1 setting on the parity 16 problems - this suggests the gNovelty⁺ parameter may not be suitable for an iPAWS-type heuristic.

Overall, the iPAWS heuristic is complex (e.g. in comparison to AdaptNovelty) and relies on a large number of hand-tuned (but robust) settings. This would argue against it if there were an effective, simpler heuristic available. However, after extensive exploration, we were unable to find a more compact combination of measures that correlated well with an optimal weight decay setting *and* were reliable across a wide range of problem types. Also, despite the complexity of the implementation, the underlying principles remain quite simple, i.e. we use the statistical properties of local search cost distributions, accumulated for different parameter settings, to bias future parameter selection decisions, according to a simulated annealing schedule. Nevertheless, it would be worthwhile to search for a more principled way to fix (or remove) the various settings on which the algorithm depends - most obviously by using existing machine learning approaches (e.g. as in [14]).

In conclusion, the paper has introduced a new approach to tuning local search parameters online. The initial implementation has been the product of considerable trial and error and should not be considered definitive. Rather, it is intended to show that the underlying concept is workable and to act as a foundation for further investigation. Nevertheless, the initial results are encouraging, and the new iPAWS algorithm has been shown to be competitive with a range of the best known local search SAT solvers.

Acknowledgements

the financial support from NICTA and the Queensland Government. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

1. Selman, B., Levesque, H., Mitchell, D.: A new method for solving hard satisfiability problems. In: Proceedings of AAAI 1992, pp. 440–446 (1992)
2. Li, C.M., Huang, W.Q.: Diversification and determinism in local search for satisfiability. In: Bacchus, F., Walsh, T. (eds.) SAT 2005. LNCS, vol. 3569, pp. 158–172. Springer, Heidelberg (2005)
3. Wu, Z., Wah, B.: An efficient global-search strategy in discrete Lagrangian methods for solving hard satisfiability problems. In: Proceedings of AAAI 2000, pp. 310–315 (2000)
4. Hutter, F., Tompkins, D., Hoos, H.H.: Scaling and probabilistic smoothing: Efficient dynamic local search for SAT. In: Van Hentenryck, P. (ed.) CP 2002. LNCS, vol. 2470, pp. 233–248. Springer, Heidelberg (2002)
5. Mills, P., Tsang, E.: Guided local search applied to the satisfiability (SAT) problem. In: Proceedings of ASOR 1999, pp. 872–883 (1999)
6. Thornton, J.R., Pham, D.N., Bain, S., Ferreira Jr., V.: Additive versus multiplicative clause weighting for SAT. In: Proceedings of AAAI 2004, pp. 191–196 (2004)
7. Hoos, H.H.: An adaptive noise mechanism for WalkSAT. In: Proceedings of AAAI 2002, pp. 635–660 (2002)
8. Pham, D.N., Thornton, J.R., Gretton, C., Sattar, A.: Advances in local search for satisfiability. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 213–222. Springer, Heidelberg (2007)
9. Xu, L., Hutter, F., Hoos, H.H., Leyton-Brown, K.: The design and analysis of an algorithm portfolio for SAT. In: Bessière, C. (ed.) CP 2007. LNCS, vol. 4741, pp. 712–727. Springer, Heidelberg (2007)
10. Gagliolo, M., Schmidhuber, J.: Dynamic algorithm portfolios. In: Proceedings of AI-MATH 2006 (2006)
11. Hutter, F., Hoos, H.H., Stützle, T.: Automatic algorithm configuration based on local search. In: Proceedings of AAAI 2007, pp. 1152–1157 (2007)
12. Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K.: A racing algorithm for configuring metaheuristics. In: Proceedings of GECCO 2002, pp. 11–18 (2002)
13. Gomes, C.P., Selman, B.: Algorithm portfolios. Artificial Intelligence 126, 43–62 (2001)
14. Hutter, F., Hamadi, Y., Hoos, H.H., Leyton-Brown, K.: Performance prediction and automated tuning of randomized and parametric algorithms. In: Benhamou, F. (ed.) CP 2006. LNCS, vol. 4204, pp. 213–228. Springer, Heidelberg (2006)
15. Carchrae, T., Beck, J.C.: Low-knowledge algorithm control. In: Proceedings of AAAI 2004, pp. 49–54 (2004)
16. Battiti, R., Protasi, M.: Reactive search, a history-sensitive heuristic for MAX-SAT. ACM Journal of Experimental Algorithms 2(Article 2) (1997)
17. McAllester, D.A., Selman, B., Kautz, H.A.: Evidence for invariants in local search. In: Proceedings of AAAI 1997, pp. 321–326 (1997)
18. Thornton, J.: Clause weighting local search for SAT. Journal of Automated Reasoning 35(1-3), 97–142 (2005)
19. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220(4598), 671–680 (1983)

Fault Resolution in Case-Based Reasoning

Ha Manh Tran and Jürgen Schönwälter

Computer Science, Jacobs University Bremen, Germany
`{h.tran, j.schoenwaelder}@jacobs-university.de`

Abstract. We present a study of reasoning methods in Case-Based Reasoning, which can be applied for the communication system fault domain. Inspired by the reasoning approach of the experts in medical diagnosis, we propose a probabilistic reasoning method which comprises two processes: a *ranking* process restricting the scope of a problem and a *selection* process finding promising solutions for the problem. We experimentally evaluate this method and draw lessons from the results to improve it.

Keywords: Case-Based Reasoning, Probabilistic Reasoning, Fault Resolution, Fault Management.

1 Introduction

The Case-Based Reasoning (CBR) [1] approach seeks to find solutions for similar problems by exploiting experience. A CBR system operates four modules: *case retrieval*, *case reuse*, *case revision* and *case retention* on the case database to resolve problems. The case reuse module involves inferring a correct solution for the current problem from experienced solutions. This module contains an adaptation process that differentiates the current problem from the retrieved cases to acquire key differences before adapting them to the retrieved cases to obtain the adapted solutions. Two prevailing approaches for the adaptation process are *transformational reuse* [2,3] and *derivational reuse* [4,5]. The salient characteristic of these approaches is to avoid choosing an identical solution from the retrieved cases, they instead resolve a problem by mapping the source and target case structures using transformational rules, or by replaying a problem-solving process using inference traces captured before. These approaches, however, demand substantial knowledge sources, complicated case representation and processing to which many problem domains cannot afford [6,7]. Developing an efficient reuse module in CBR systems thus remains challenging for many problem domains; in particular, for problem domains where cases are poorly represented.

Our study aims at providing a reasoning method for a distributed CBR system [8,9], which assists operators in finding solutions for faults in large-scale communication systems. This problem domain is concerned with semi-structured fault records; i.e., pre-defined fields are used to keep track of the status of a fault while textual descriptions are used to describe the problem. The multi-vector representation method [9] describes fault cases in vectors that exploit fault symptoms

to facilitate fault retrieval and reasoning. This study focuses on applying the reasoning approach of the experts in medical diagnosis [10] to resolving faults. Briefly, when examining patients, practicing physicians use appropriate examinations for typical clinical situations and formal schemes for difficult decisions. Our proposed method contains two processes: (i) a *ranking* process aims to narrow down the scope of a problem by using the k-Nearest-Neighbor (kNN) algorithm [11] to determine a set of cases that share common symptoms, and (ii) a *selection* process aims to predict promising solutions by using Bayesian computation to evaluate the correlation between those cases and the problem through symptoms. An essential demand of this method is to quickly provide useful information (e.g., similar solutions) for fixing problems.

The rest of the paper is structured as follows: the next section provides some background about reasoning methods in the CBR system, and also revises the multi-vector representation method. Section 3 concentrates on the proposed reasoning method. Section 4 describes the evaluation of this method that includes the creation of datasets, experiments and results. Related work is presented in Section 5 before the paper concludes in Section 6.

2 Background

Most CBR systems operate on a local case database. We are working on a distributed CBR system extending the capability of the conventional CBR system by exploring multiple problem-solving knowledge sources. It takes advantage of computation and storage power at various sites, where independent CBR engines can operate in parallel, thus improving the efficiency of resolving complicated problems and the performance of managing huge federated case databases.

Our distributed CBR system, namely DCBR, takes advantage of P2P technologies to acquire some degree of self-organization, scalability in architecture, and flexibility in search in decentralized and federated environments. DCBR comprises several powerful peers acting as independent CBR engines to explore multiple knowledge sources; i.e., sharing knowledge resources and search facilities with other peers. DCBR also uses the multi-vector representation method to exploit semi-structured fault records for retrieval and reasoning.

This paper focuses on the case reuse module in DCBR, particularly on case adaptation or case reasoning. It is difficult to perform automated adaptation to the communication system fault domain, shortly the fault domain, because fault cases contain semi-structured data and present a variety of problems and solutions without a guarantee for the existence of correct solutions. The transformational approach requires experienced problems structurally similar to the problem or the derivational approach requires adapting traces captured before. Other proposed approaches identify the problem into problem categories [12,13,14,15], or seek promising solutions among experienced solutions [16,17]. Our method aims to provide promising solutions quickly with less support from operators.

2.1 Probabilistic Reasoning

The motivation for applying a probabilistic reasoning (PR) method to CBR systems comes from the demand of making decision under uncertain situations. This method built on probability theory has been used either for indexing cases in case retrieval or for inferring cases in case reuse. We study the application of PR methods to the reasoning engine of CBR systems for dealing with the limitation of knowledge sources. In particular, an engine rests on an incomplete case database to decide which cases potentially resolve the problem, and which case properties properly infer facts related to the possible solution. We have seen several reasoning methods that broadly fall into two categories:

Using the entire case database [12,13,14,15]: This method constructs a probabilistic model based on the whole case database; e.g., determining the problem distribution of a case database based on case properties. The model can then be used to conjecture the class of the problem. The method usually requires a large case database to build the model as accurate as possible. It is appropriate for classifying problems in problem domains that possess a small set of problem categories. In several studies, this method takes advantage of automatic learning algorithms to establish the probabilistic model that can also be used for indexing and retrieving cases in CBR systems.

Using the partial case database [16,17]: This method is suitable for selecting solutions (diagnosing and searching) in problem domains with a variety of problems and solutions. The method relies on a small number of relevant cases obtained by case retrieval and provides more concrete solutions for problems. A probabilistic model can be built up by using the typical properties of the retrieved cases and then be used to select or infer promising solutions; e.g., determining case properties such as symptoms and causes influencing a problem significantly. The probabilistic model can be established by the experts and automatic learning algorithms.

Bayesian Formulas. [18] This part briefly revisits Bayesian formulas that will be used in this paper. Considering H as a hypothesis and $e_n = e^1, \dots, e^n$ as a sequence of evidence pieces obtained from a scenario with an assumption that e^1, \dots, e^n are independent from each other. The posterior probability for the multi-evidence scenario is derived by the conditional probability formula:

$$P(H|e_n) = \frac{P(H)P(e_n|H)}{P(e_n)} = \frac{P(H)\prod_{k=1}^n P(e^k|H)}{P(e_n)} \quad (1)$$

where $P(e_n|H) = \prod_{k=1}^n P(e^k|H)$ since evidence pieces e^k are independent from each other, $P(H)$ is the prior probability of hypothesis H and $[P(e_n)]^{-1}$ is determined by the requirement $P(H|e_n) + P(\neg H|e_n) = 1$. This formula indicates that the belief of hypothesis H upon receiving e^k can be computed by the previous belief $P(H)$ and the likelihood $P(e^k|H)$ that e^k will materialize if H is true.

This formula also serves the purpose of prediction and diagnosis. The prior probability $P(H)$ measures the *predictive* support accorded to H by the background knowledge, while the likelihood $P(e_n|H)$ represents the *diagnostic*

support given to H by the evidence pieces actually observed. The posterior probability $P(H|e_n)$ indicates the strength of belief in a hypothesis H based on the previous knowledge and the observed evidence pieces e_n .

2.2 Multi-Vector Representation

Cases in the fault domain contain semi-structured data. It is difficult to extract symptoms from textual fault data for precise comparison and reliable inference. We have proposed the multi-vector representation method [9] (MVR) to facilitate case retrieval and case reasoning in DCBR. Briefly, a case contains a problem and a solution represented by a set of semantic and feature vectors. We only use problem vectors for case reasoning.

A *Semantic vector* is generated by indexing significant terms using text processing techniques [19,20]. The indexing process involves building term vectors from the textual description of cases, weighting each term by calculating the appearance frequency of this term in cases using the Vector Space Model [20] and producing semantic vectors by removing noise from term vectors using algebraic computation. Since the process works on the basis of terms, the majority of information of symptoms and diagnoses (i.e., debug and probe information) is usually lost, as in the following example:

After upgrading to 0.070-1 udev stopped working on my system (sid, kernel: 2.6.12-6). For example, ipw2200 could not load the firmware anymore:

```
Sep 16 22:15:02 localhost kernel: ipw2200: Detected Intel Wireless Network Connection
Sep 16 22:15:02 localhost kernel: ipw2200: ipw-2.3-boot.fw load failed: reason-2
Sep 16 22:15:02 localhost kernel: ipw2200: Unable to load firmware: 0xFFFFFFFFFE
Sep 16 22:15:02 localhost kernel: ipw2200: failed to register network device
```

A semantic vector cannot express the meaning of *ipw2200*, *0.070-1 udev stopped working*, *unable to load firmware* and *failed to register network device*, etc. This vector is mostly used for case retrieval, and to some extent it can support case classification based on specific keywords; i.e., each keyword points to a list of cases. Case similarity is measured by the cosine function of two vectors.

A *Feature vector* contains field-value pairs to describe symptoms in cases, where fields are pre-defined, domain-specific and values are either binary, numeric or symbolic; e.g., *component: kernel_2.6.12-6*, *package: udev*, *package-version: 0.070-1*, *problem-type: upgrading-failed*, *problem-area: software*, etc. Still, these values cannot capture the complete meaning of symptoms and diagnoses that are crucial for reasoning. The field-value pairs are thus extended to facilitate the expression of symptoms and diagnoses, as in the following example:

```
 $S_1 = \text{Detected Intel Wireless Network Connection}$ 
 $S_2 = \text{ipw-2.3-boot.fw load failed}$ 
 $S_3 = \text{Unable to load firmware}$ 
 $S_4 = \text{Failed to register network device}$ 
 $S_5 = \text{udev stopped working after upgrading to 0.070-1}$ 
```

Case similarity is measured by the sum of weight values of matched field-value pairs of two vectors. The implementation of symptom weight assignment and comparison requires some support from the experts in the beginning. As the

case database has been built up, the situation can be improved by providing fields and values recommended by similar fault cases; i.e., with specific problem types, DCBR can demand the result of specific probes along with new symptoms found by users. This vector plays a key role in case reasoning.

Using the feature and semantic vectors for case retrieval has been studied in [9]. The experimental evaluation has shown that the combination of these vectors improves the performance of case retrieval.

3 Reasoning Method

We propose a probabilistic reasoning method that uses the partial case database for case reasoning in DCBR. Since DCBR has already included a case retrieval method that obtains relevant cases from peers' entire case database [9], we only focus on a case reasoning method that infers promising solutions from a set of the retrieved cases in this study. The solution selection method tends to be more successful for a small, well-constrained set of problems [10]. Moreover, this method can obtain better processing time by working with a small set of cases.

Our PR method contains two *ranking* and *selection* processes. The first process operates on symptoms to figure out cases showing the same symptoms as the problem's, see Fig. 1b (Fig. 1a only demonstrates case retrieval). This process aims to narrow down the scope of the problem by providing a smaller set of promising cases. Note that complicated cases comprise many symptoms and diagnoses, reducing a number of cases leads to a smaller number of symptoms and thus lower computation cost. Formally, given the problem C_p and the relevant cases C_r acquired by case retrieval, both C_p and C_r share a set of common symptoms. The process estimates their similarity by using the k-Nearest-Neighbor algorithm [11] with a similarity function defined as follows:

$$sim(C_r, C_p) = \sum_{i=1}^t w_{ri} w_{pi} \quad (2)$$

where t is number of common symptoms, and w_{ri} and w_{pi} are the weight values of these symptoms in C_r and C_p respectively. The similarity function considers two factors: the number of common symptoms between two cases and the significance of these symptoms in each case.

The second process aims at predicting promising solutions for the problem using Bayesian computation, see Fig. 1c. Given a set of cases C , where a case C_r contains a set of symptoms $\{S_1, \dots, S_k\}$ and a solution, we assume that solutions in C as a set of exhaustive and mutually exclusive hypotheses $\{H_1, \dots, H_n\}$, and that any symptom is the result of a diagnosing probe, e.g., a ping probe provides either the high probability of success or the low probability of success (i.e., failure). The problem contains a set of symptoms $\{S_1, \dots, S_h\}$ without a solution (note that cases and the problem can share the same symptoms). Thus, the puzzle is to find the highest conditional probability of the hypotheses $P(H_i|S_1, \dots, S_h)$ with $i = 1, \dots, n$.

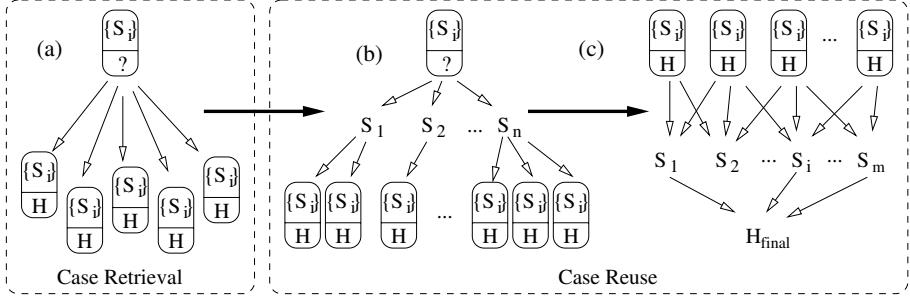


Fig. 1. $\{S_i\}$, H , $?$ and H_{final} denote a set of symptoms in a case, a hypothesis, the unknown hypothesis and the promising hypothesis respectively. (a) Case retrieval based on the evaluation between the problem and cases through vectors, (b) Case ranking based on the evaluation between the problem and cases through symptoms, (c) Case selection based on the correlation among cases and the problem through symptoms.

Considering a set of exhaustive and mutually exclusive hypotheses H_1, \dots, H_n and S_1, \dots, S_h as a set of evidence pieces (or symptoms) obtained from the problem with an assumption that S_1, \dots, S_h are independent from each other. Applying Eq. 1, we obtain:

$$P(H_i|S_1, \dots, S_h) = \frac{P(S_1, \dots, S_h|H_i)P(H_i)}{P(S_1, \dots, S_h)} = \alpha P(H_i) \prod_{j=1}^h P(S_j|H_i) \quad (3)$$

where $P(S_1, \dots, S_h|H_i) = \prod_{j=1}^h P(S_j|H_i)$ since S_j are independent from each other, $P(H_i)$ are the prior probabilities of hypotheses, and $\alpha = [P(S_1, \dots, S_h)]^{-1}$ is determined via the requirement $\sum_{i=1}^n P(H_i|S_1, \dots, S_h) = 1$.

In case the evidence set S contains a new evidence piece S_{new} (e.g., the problem has a new symptom), updating the new evidence piece first computes $P(H_i|S)$ and then uses $P(H_i|S, S_{new})$, as follows:

$$P(H_i|S, S_{new}) = \frac{P(H_i|S)P(S_{new}|S, H_i)}{P(S_{new}|S)} = \beta P(H_i|S)P(S_{new}|H_i) \quad (4)$$

where β is determined by the same method as α , $P(H_i|S)$ is computed as previously, and $P(S_{new}|H_i)$ is determined by the experts.

The algorithms simply reflect the above discussions. Algorithm 1 iterates over cases (2), finds common symptoms (3) and accumulates weight values (6, 7) before ranking cases in the resulting set (8). Algorithm 2 iterates over hypotheses (2), initializes the prior probability with a value $\frac{1}{n}$, where n is the number of hypotheses (3), computes and normalizes the posterior probabilities (4, 5, 6, 7) before ranking cases in the resulting set (8).

Example: We have a set of cases with the following solutions and symptoms related to the *connection failure* problem:

Algorithm 1: Case Ranking

Input: C : a set of cases C_r
 C_r, w_r : sets of symptoms & weight values
 C_p, w_p : sets of symptoms & weight values
Output: R : a ranked set of cases

```

1  $R \leftarrow \emptyset$ 
2 for each  $C_r \in C$  do
3    $S = C_r \cap C_p$ 
4   if  $S \neq \emptyset$  then
5      $T_r = 0$ 
6     for each  $S_i \in S$  do
7        $T_r = T_r + w_{r_i}w_{p_i}$ 
8     insert  $C_r$  in  $R$  in the order of  $T_r$ 
```

Algorithm 2: Case Selection

Input: C : a set of cases C_r & solutions H_r
 V : a list of probability values V_r
 C_r, w_r : sets of symptoms & weight values
 C_p : sets of symptoms $\{S_1, \dots, S_h\}$
Output: F : a final set of solution cases

```

1  $F \leftarrow \emptyset$ 
2 for each  $H_r \in C$  do
3    $V_r = \frac{1}{n}$ 
4   for each  $S_i \in C_p$  do
5      $V_r = V_r w_{r_i}$ 
6   for each  $V_r \in V$  &  $C_r \in C$  do
7      $V_r = V_r \|V\|^{-1}$ 
8   insert  $C_r$  to  $F$  in the order of  $V_r$ 
```

- H_1 = Checking firewall software for blocking connections
 - S_1 = Desktop keeps disconnecting from the Internet
 - S_2 = Desktop and Laptop keeps connecting from the router
 - S_3 = Connection usually goes really slow
 - S_4 = Connection is fine before updating the firewall software
 - S_5 = Router is WHR-HP-G54 and wireless adapter is Linksys WMP54G
- H_2 = Reinstalling networking components (TCP/IP)
 - S_1 = Desktop completely stops connecting to the Internet
 - S_2 = Laptop can connect to desktop and the Internet
 - S_3 = Desktop disconnects to laptop and D-Link router with a limited connectivity
 - S_6 = Desktop uses an Etherlink 10/100 PCI card and laptop uses a wireless adapter
 - S_7 = Registry was damaged on desktop few days ago
- H_3 = Checking router configuration for the IP address range
 - S_1 = Desktop cannot connect to a router and the Internet
 - S_2 = Laptop connects to the router and the Internet
 - S_4 = The firewall software is often updated on those machines
 - S_8 = Desktop gets error message of address already used when renewing

The following table presents the weight values of symptoms to the solutions (note that updated weight values are not bold). This table is for demonstration, we only need weight values related to the problem's symptoms for implementation:

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
H_1	0.1	0.1	0.25	0.448	0.001	0	0.1	0.001
H_2	0.1	0.1	0.3	0.001	0	0.048	0.45	0.001
H_3	0.25	0.245	0	0.005	0	0.05	0	0.45

In order to fulfill the requirement of exhaustive and mutually exclusive hypotheses, we examine a set of hypotheses H_1 , H_2 and H_3 , and ignore other hypotheses; i.e., the conditional probability of other hypotheses is 0. This, however, can be a problem in practice if the examined set of hypotheses does not contain the

desired hypothesis. We also consider a set of evidence pieces S_1, \dots, S_8 obtained by distinct probes independent because the effect of a probe to other evidence pieces is minor, and evidence pieces can only be correlated if probes are not distinct; e.g., if a connection failure occurs, symptoms collected by the *ping* and *ftp* probes can be correlated. Hypotheses possess the same prior probabilities $P(H_i) = (0.33, 0.33, 0.34)$, and the problem contains the following symptoms:

- $H = ?$
- S_1 = Desktop gets connection failure
- S_2 = Other machines still connect to routers and to the Internet,
- S_4 = Desktop updated the firewall software two days ago.

By applying Eq. 3, we obtain $P(H_i|S_1, S_2, S_4) = (0.8719, 0.0019, 0.1261)$ with $i = 1, 2, 3$. The result indicates that the chance of firewall software blocking connections is 87.19% given the symptoms of the problem. Intuitively, a solution is deduced by an incomplete set of symptoms; solutions likely share a subset of symptoms. Bayesian computation distinguishes those solutions by the significance of symptoms in a case and the significance of symptoms among cases.

4 Evaluation

We have created a dataset to evaluate the performance of our reasoning method. The dataset contains bug reports crawled from four bug tracking systems (Trac, Bugzilla, Mantis, and Debian) and stored in a unified data model [21]. For the ranking process, we have generated a set of 50 queried problems. Each queried problem possesses a set of significant keywords and symptoms extracted from the dataset. After retrieving similar bugs from the dataset, see Fig. 1a, we perform the ranking process on the retrieved bugs, and then verify the precision of the process. The precision rate is measured by the ratio of the number of correct bugs obtained to the total number of bugs obtained. A correct bug shares a high number of common symptoms with the queried problem. We ignore the recall rate because all of the retrieved bugs from case retrieval were similar to the queried problem already. We use a threshold θ to obtain bugs, where θ is determined by the average value of ranked values of bugs (T_r) in the ranked set, see Algorithm 1. The small dataset for experiments contains 71,450 bugs.

Fig. 2 depicts the performance of the ranking process based on the precision metric. The line graph increases from 0.65 to 0.78 over 50 problems because correct bugs dominate in all bugs obtained for each problem. The criteria of correct bugs therefore have an impact on this result. The ranking process achieves a precision rate higher than 0.7 on average. Since the resulting bugs from the ranking process contain a high number of common symptoms, the selection process can effectively use them for reasoning. The bar graph also characterizes 83% of problems receiving precision values higher than 0.7 and 17% of problems receiving precision values lower than 0.7. We learned that the bug dataset is wide and diverse in scope and only a few cases specifically focus on the same problem in the fault domain, thus it is difficult to apply the dataset for the selection process.

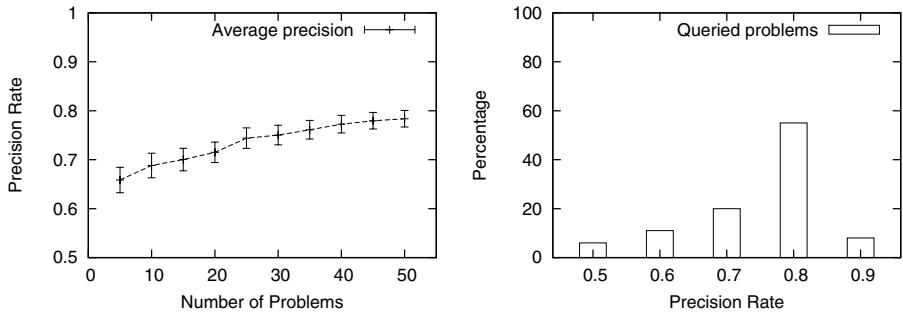


Fig. 2. Average precision by number of problems and problem distribution by precision

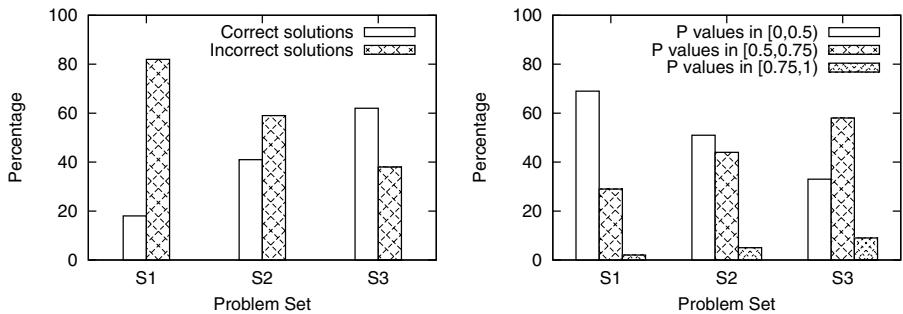


Fig. 3. Solution distribution by number of problems and probability distribution by correct solutions for three problem sets (P denotes probability)

We have created another dataset to evaluate the selection process because the crawled bug dataset currently requires extra work on symptoms extraction and problem classification. This dataset comprises problem scenarios published at a networking forum [22] that concentrates on problems in a small-scale network such as a home network or an office network. This dataset contains 60 *connection failure* cases resulting from various causes: (i) *hardware* including network card malfunctioning, router malfunctioning, bad cable, etc (13 cases), (ii) *software* including firewall blocking, bad network card driver, bad networking component, router upgrading, etc (22 cases), and (iii) *configuration* including bad router setting, bad network setting, bad security setting (25 cases). Cases contain a hypothesis and a set of symptoms with weight values, see the example in Section 3. We have generated three sets of queried problems with one (S_1), two (S_2) and three (S_3) common symptoms extracted from the dataset.

We perform the selection process on the queried problems, and verify the proposed solutions with *correct* or *incorrect*. The left bar graph in Fig. 3 indicates that, with the increasing number of common symptoms in S_1 , S_2 and S_3 , the distribution of incorrect and correct solutions are inverted. S_1 receives 82% of

incorrect solutions and \mathcal{S}_3 receives 38% of incorrect solutions. We observed that while incorrect solutions in \mathcal{S}_1 are caused by ambiguous information (e.g., vague symptoms with high weight values), incorrect solutions in \mathcal{S}_3 are caused by confusing information (e.g., two salient symptoms with high weight values). Thus assigning weight values to symptoms crucially affects the outcome of solutions.

We investigate the probability distribution of correct solutions in \mathcal{S}_1 , \mathcal{S}_2 , and \mathcal{S}_3 to learn further the dataset and the queried problems. The right bar graph in Fig. 3 shows that the number of solutions with $P \geq 0.75$ is very small compared to the number of solutions with $P < 0.75$. A solution with more common symptoms receives high probability due to the accumulation of weight values. A solution with fewer symptoms still receives high probability if these symptoms are distinct. A solution receiving low probability indicates that either the case or the problem provides too general information such as vague symptoms. Reducing these vague symptoms can make the dataset better, thus ameliorating the performance of the selection process.

5 Related Work

Our study involves fault resolution in communication systems and probabilistic reasoning methods in CBR. The study in [23] improves managing faults in computer network using the CBR approach. The idea is to extend the trouble ticket system (TTS) to applying CBR, where trouble tickets are represented in field-value pairs as cases. The retrieval and reasoning processes employ simple similarity functions dealing with binary and numeric values to solve novel faults. The study of Heckerman et al. [16] has proposed an approach to problem diagnosis and troubleshooting for printers using probabilistic reasoning in CBR. Their approach involves indexing the case database, retrieving relevant case properties such as *symptom*, *issue*, *cause*, and constructing Bayesian networks for solving problems. The DUMBO system [24] takes advantage of the knowledge hoarded in TTS to propose solutions for problems. The system contains six types of features to express cases, and provides both similarity and reliability measurements for evaluating cases. However, these systems are relatively limited by several aspects; e.g., the representation of trouble tickets is only suitable for simple feature matching mechanisms, thus restricting the feature exploitation and the reasoning engine, or the knowledge source is only based on local case databases.

Several studies [14,25,15] exploit Bayesian computation to manipulate the case database for ameliorating case retrieval; e.g., indexing cases, computing similarity metrics and extracting contextual information. Other studies also extend Bayesian computation to case classification and adaptation. A probabilistic framework [13] has been built on data-intensive domains for CBR. Case adaptation constructs a Bayesian probability model from the case database. This model is used to perform classification prediction for public domain datasets. The study of Rodríguez et al. [14] has proposed a probabilistic model for indexing and classifying in CBR. Their model uses two Bayesian networks to represent the relationship among *category*, *exemplar* and *feature* corresponding to

cases. Their model determines a new case's class by using Bayesian formulas to compute the probability of categories and exemplars based on the new case's features. Features are limited to binary values only. The study of Lazcano et al. [17] focuses on problem classification using a hybrid method that combines the kNN algorithm with the Bayesian network paradigm. Their approach uses the kNN algorithm to acquire the nearest case based on the discretized case database and the new case, it then propagates the observations of the nearest case to the previously learned Bayesian network as Naive Bayes classifier.

6 Conclusion

We present in this paper a probabilistic reasoning method for the communication system fault domain. When a fault is found in a system, the operator obtains all data related to the fault and endeavors to find the solution quickly. As a matter of fact, fault data usually contains several symptoms that can be exploited to resolve the fault. Motivated by the reasoning approach of the experts in medical diagnosis [10], the proposed probabilistic reasoning method exploits fault symptoms by using the ranking and selection processes based on Bayesian computation. The former process provides a smaller set of similar cases that narrows down the scope of the fault, whereas the latter process evaluates the correlation between cases and the fault to find promising solutions. The simplicity of this method helps providing solutions quickly.

This study is a part of our distributed CBR system [8], which aims at assisting operators in finding solutions for faults in large-scale communication systems. Experiments thus depend on the multi-vector representation method [9] for describing and retrieving cases before running the reasoning processes. The experimental results characterize the performance of the ranking process on a bug dataset crawled from several bug tracking systems [21] and the performance of the selection process on another dataset extracted from a networking forum [22]. We learned that when the bug dataset is wide and diverse in scope, it is difficult to apply the selection process that works on particular classes of problems in the fault domain. We also learned that the performance of the reasoning processes heavily depends on the refinement of the case database; i.e., the case database without vague symptoms and incorrect weight values. Future work focuses on these issues and experiments for a distributed setting.

Acknowledgement. The work reported in this paper is supported by the EC IST-EMANICS Network of Excellence (#26854).

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* 7(1), 39–59 (1994)
2. Winston, P.H.: Learning and Reasoning by Analogy. *Communications of the ACM* 23(12), 689–703 (1980)

3. Koton, P.K.: Using Experience in Learning and Problem Solving. PhD thesis, Laboratory of Computer Science, Massachusetts Institute of Technology (1988)
4. Carbonell, J.G.: Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. In: Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) Machine Learning: An Artificial Intelligence Approach. Morgan Kaufman Publishers, California (1986)
5. Veloso, M.M., Carbonell, J.G.: Derivational Analogy in Prodigy: Automating Case Acquisition, Storage, and Utilization. *Machine Learning* 10, 249–278 (1993)
6. Cunningham, P., Finn, D., Slattery, S.: Knowledge Engineering Requirements in Derivational Analogy. In: Proc. 1st European Workshop on Topics in Case-Based Reasoning, London, UK, pp. 234–245. Springer, Heidelberg (1994)
7. Blumenthal, B., Porter, B.W.: Analysis and Empirical Studies of Derivational Analogy. *AI Journal* 67(2), 287–328 (1994)
8. Tran, H.M., Schönwälder, J.: Distributed Case-Based Reasoning for Fault Management. In: Proc. 1st International Conference on Autonomous Infrastructure, Management and Security, pp. 200–203. Springer, Heidelberg (2007)
9. Tran, H.M., Schönwälder, J.: Fault Representation in Case-Based Reasoning. In: Proc. 18th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, pp. 50–61. Springer, Heidelberg (2007)
10. Szolovits, P., Pauker, S.G.: Categorical and Probabilistic Reasoning in Medical Diagnosis. *Artificial Intelligence* 11(1-2), 115–144 (1978)
11. Cover, T., Hart, P.: Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory* 13(1), 21–27 (1967)
12. Bareiss, R.: Exemplar Based Knowledge Acquisition: A Unified Approach to Concept Representation, Classification, and Learning. Academic Press Professional, Inc., California (1989)
13. Tirri, H., Kontkanen, P., Myllymäki, P.: A Bayesian Framework for Case-Based Reasoning. In: Proc. 3rd European Workshop on Advances in Case-Based Reasoning, London, UK, pp. 413–427. Springer, Heidelberg (1996)
14. Rodríguez, A.F., Vadera, S., Sucar, L.E.: A Probabilistic Model for Case-Based Reasoning. In: Proc. 2nd International Conference on Case-Based Reasoning Research and Development, London, UK, pp. 623–632. Springer, Heidelberg (1997)
15. Gomes, P.: Software Design Retrieval Using Bayesian Networks and WordNet. In: Proc. 7th European Conference on Advances in Case-Based Reasoning, pp. 184–197. Springer, Heidelberg (2004)
16. Heckerman, D., Breese, J.S., Rommelse, K.: Troubleshooting Under Uncertainty. Technical Report MSR-TR-94-07, Microsoft Research (January 1994)
17. Lazkano, E., Sierra, B.: BAYES-NEAREST: A New Hybrid Classifier Combining Bayesian Network and Distance Based Algorithms. In: Proc. 11th Portuguese Conference on Artificial Intelligence, Berlin, Germany, pp. 171–183. Springer, Heidelberg (2003)
18. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)
19. Deerwester, S., Dumais, S., Landauer, T., Furnas, G., Harshman, R.: Indexing by Latent Semantic Analysis. *Journal of the Society for Information Science* 41(6), 391–407 (1990)
20. Berry, M.W., Drmac, Z., Jessup, E.R.: Matrices, Vector Spaces, and Information Retrieval. *SIAM Review* 41(2), 335–362 (1999)
21. Tran, H.M., Chulkov, G., Schönwälder, J.: Crawling Bug Tracker for Semantic Bug Search. In: Proc. 19th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, pp. 55–66. Springer, Heidelberg (2008)

22. Networking Forum (Last access, May 2008),
<http://www.computing.net/networking/wwwboard/wwwboard.html>
23. Lewis, L.M.: A Case-Based Reasoning Approach to the Resolution of Faults in Communication Networks. In: Proc. 3rd International Symposium on Integrated Network Management (IFIP TC6/WG6.6), pp. 671–682. North-Holland, Amsterdam (1993)
24. Melchiors, C., Tarouco, L.M.R.: Fault Management in Computer Networks Using Case-Based Reasoning: DUMBO System. In: Proc. 3rd International Conference on Case-Based Reasoning and Development, pp. 510–524. Springer, Heidelberg (1999)
25. Dingsøyr, T.: Retrieval of Cases by using a Bayesian Network. In: Proc. AAAI Workshop on Case-Based Reasoning Integration (1998)

Constrained Sequence Classification for Lexical Disambiguation

Tran The Truyen, Dinh Q. Phung, and Svetha Venkatesh

Department of Computing, Curtin University of Technology
GPO Box U1987 Perth, Western Australia 6845, Australia
`thetruyen.tran@postgrad.curtin.edu.au,`
`{d.phung,s.venkatesh}@curtin.edu.au`

Abstract. This paper addresses lexical ambiguity with focus on a particular problem known as accent prediction, in that given an accentless sequence, we need to restore correct accents. This can be modelled as a sequence classification problem for which variants of Markov chains can be applied. Although the state space is large (about the vocabulary size), it is highly constrained when conditioned on the data observation. We investigate the application of several methods, including Powered Product-of- N -grams, Structured Perceptron and Conditional Random Fields (CRFs). We empirically show in the Vietnamese case that these methods are fairly robust and efficient. The second-order CRFs achieve best results with about 94% term accuracy.

Keywords: constrained sequence classification, lexical disambiguation, Vietnamese accent restoration, conditional random fields.

1 Introduction

Lexical ambiguity is a common problem in natural language processing because lexical analysis is often a first step for high level understanding. In this paper, we focus on a particular problem known as *accent prediction*¹, although the methods can be similarly adapted to other lexical problems such as case prediction and spelling correction (e.g. see [5]).

Accent prediction here refers to the situation where accents are removed (e.g. by some email preprocessing systems), cannot be entered (e.g. by standard English keyboards), or not explicitly represented in the text (e.g. in Arabic). Here we deal with languages that use Roman characters in writing together with additional accent and diacritical marks. Examples are European languages such as Spanish and French (see [8] for comprehensive list) and Asian languages such as Chinese Pinyin and Vietnamese.

The problem often arises because most keyboards today are designed for English, which means without further help, we can only type the Roman alphabets and get an ‘approximate’ message that is closed to the intended message. The

¹ We use ‘accent’ to refer to either accents or any diacritical marks.

practice is popular in email communication, instant messaging and mobile SMS. For example, a Vietnamese sentence: *bạn hãy thăm Việt Nam ngay hôm nay* ('please visit Vietnam today') will be written as an accentless sequence as *ban hay tham Viet Nam ngay hom nay*. Decoding such a message can be quite hard for both human and machine. For instance, the accentless term *ngay* can easily lead to confusion between the original Vietnamese *ngay* ('now' or 'straight') and the plausible alternative *ngày* ('day').

Thus predicting accents is not only useful to recover lost accents, it also reduces typing burden when it provides online suggestions as a shortcut for multiple key combinations. Our approach to this problem is to apply sequence classification techniques. The approach is expected to be more robust than local methods that look only for local context of surrounding words. In addition, training data is not a problem as it is often readily available without any cost of manual labeling. In this paper, we investigate the application of Powered Product-of- N -grams (PPoNs), Structured Perceptron [2] and Conditional Random Fields [7] (CRFs).

The rest of the paper is organised as follows. Related work and background are reviewed in Section 2. The statistical modelling and the PPoNs are proposed in Section 3. Section 4 details the constrained sequence classification methods. In Section 5, we describe the experiments and results for evaluating the proposed methods. Finally, Section 6 concludes the paper.

2 Background

2.1 Previous Work

The most popular approach to lexical ambiguity is corpus-based, where rules and statistical decisions are estimated from the training data. In the case of accent prediction, this is particularly suitable because training data is often readily available without any manual annotation.

A wide range of classification techniques have been used for the accent prediction problem. A comparative study of local methods including most frequent pattern, Bayesian is reported in [14]. These are limited to Spanish and French, where the ambiguity is not very high. For example, using just most frequent accent pattern gives 98.7% accuracy for Spanish. The same approach for Vietnamese, however, achieves only 71.83% accuracy.

Other classification methods include Memory-Based Learning [4], Weighted Finite-State Transducers [9], Hidden Markov Models [12].

The view of accent prediction as sequence classification was considered in [15]. In this work, the Maximum Entropy (MaxEnt) method [1] is used. This is a local method that is adapted for sequences by including label history as features. Our proposal, on the contrary, is to use global methods for classifying sequences.

With respect to lexical analysis there are two main levels: the word level and letter level [8,13]. While the former may be more natural with smoother language models, the latter is more useful when training data is limited (e.g. for languages

with little electronic corpora), or when we have to deal with unknown words (e.g. in medical text [16]).

We are only aware of the published result for Vietnamese in [4], where the best result is only 75.5% term accuracy, much lower than our result of 94.3%. Some software packages are also available, such as AMPad² and VietPad³ but we have not been able to experimentally compare with our methods using the same setup.

2.2 Vietnamese Writing System

The Vietnamese writing system utilises a set of Roman alphabets (the characters ‘f’, ‘j’, ‘w’ and ‘z’ are not used) and a small set of new symbols and a set of five tonal marks. The five tonal marks are associated with vowels to account for voice stress. Each vowel, and therefore each term, either has zero or one tonal mark. In combination of consonants and vowels, there are about 10^4 unique terms (syllables or unigrams). One or more consecutive terms constitute a word, which is the smallest meaningful text unit. A typical word, especially those in formal writing, has two terms. There are about 10^5 words in the dictionary. Note that word boundaries are not predefined by white spaces as in English. For higher level of understanding, we need to do word segmentation [3], and this is an interesting and important problem of its own right.

3 Problem Modelling

An input sentence $s = (s_1, s_2, \dots, s_T)$ can be considered as a distorted version of the original (but unknown) sentence $v = (v_1, v_2, \dots, v_T)$, where there is typically a correspondence between the original term v_t and the input term s_t . An exception is that the terms are mistakenly swapped during keying, but this is out of scope of this paper.

In particular, in accent prediction an accentless term is generated by de-accenting the accented counterpart

$$s_t = R(v_t) , \quad (1)$$

where $R(v_t)$ is a deterministic function, in that each v_t would yield a unique s_t . In other lexical problems, however, each v_t may correspond to multiple possible assignments of s_t .

The prediction is defined as finding the correct sequence \hat{v} given the distorted sequence s

$$\hat{v}|s = \arg \max_{v \in \mathcal{V}(s)} P(v|s) \quad (2)$$

$$= \arg \max_{v \in \mathcal{V}(s)} P(v)P(s|v) , \quad (3)$$

where $\mathcal{V}(s)$ is the space of all possible correct sentences whose distorted form is s .

² <http://www.echip.com.vn/echiproot/weblh/qcbg/duynghi/ampad/readme.htm>

³ <http://vietunicode.sourceforge.net/download/vietpad/>

3.1 Powered Product-of- N -Grams

In the case of accent prediction we have $P(s|v) = 1$ since the de-accenting is deterministic, and thus

$$\hat{v}|s = \arg \max_{v \in \mathcal{V}(s)} P(v) . \quad (4)$$

The problem is now to estimate the language model $P(v)$. In this subsection, we propose to use n -grams as they are still the simplest and effective method. In general, as n increases, we have better language model but we may need a huge data set to have reliable estimate. One effective strategy is to combine n -gram models as follows⁴

$$P(v) = \frac{1}{Z} \prod_n P_n(v)^{w_n} , \quad (5)$$

where $Z = \sum_v \prod_n P_n(v)^{w_n}$, $w_n \geq 0$ and $P_n(v)$ are distribution given by n -gram model. Let us call this method the Powered Product-of- N -grams (PPoNs). The beauty of PPoNs is that the computational complexity is the same as its components, whilst we can adjust the contribution of the components by tuning the extra parameters w_n . The distribution by the PPoNs is often more peaked than the component parts. For example, if the component models agree on a particular sentence v , the PPoNs would yield a very high or very low probability.

The main drawback of the PPoNs model is that we cannot evaluate the normalisation term Z . It prevents estimating the parameters w_n using standard methods such as maximum likelihood. In this work, we manually tune w_n through trials and errors.

4 Constrained Sequence Classification

In standard sequence classification such as part-of-speech tagging we deal with the full state set, and thus with all possible state paths from the start to the end of the sequence. However, in word prediction the state set is particularly large (e.g. in order of $10^4 - 10^5$), it is not practical to perform dynamic programming because the time complexity is quadratic in the size of the state set. Fortunately, in lexical analysis, for each input term, there is a quite small number of corresponding alternatives. We call the set of alternatives by *proposal set*, which can be estimated from a large enough corpus. For example, in accent prediction, the size of proposal sets are less than 25 in the case of Vietnamese, and 5 in Chinese Pinyin. Thus any state path that does not go through those in the proposal set will be eliminated. In other words, the state space is constrained.

⁴ One reviewer pointed out that PPoNs are similar to smoothing techniques which approximate n -gram distribution by lower-order distributions. This is interesting because we are originally motivated by the ensemble methods from the machine learning view.

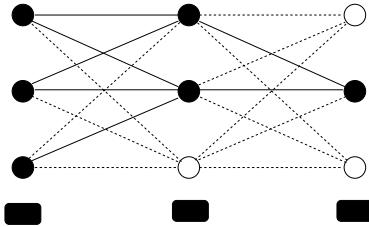


Fig. 1. State paths in constrained first-order Markov chain. Filled circles denotes admissible states, lines denote possible paths, and rounded rectangles denote input terms.

The constraint suggests a better way to model the problem: we do not need to deal with the full state space, rather, for each input sequence, we limit ourselves to the constrained space, *conditioned* on the input sequence. In other words, we estimate the conditional distribution $P(v|s)$ directly.

Denote by $\mathcal{V}(s_t)$ the proposal set for the input term s_t , thus $\mathcal{V}(s) = \mathcal{V}(s_1) \times \mathcal{V}(s_1) \times \dots \times \mathcal{V}(s_T)$.

4.1 Conditional Random Fields Modelling

Conditional random fields (CRFs) are particularly suitable for modeling $P(v|s)$. Assuming the $(n+1)$ th order Markov chain, the CRF distribution is given as

$$P(v|s) = \frac{1}{Z(s)} \exp\left(\sum_c \sum_k \lambda_k f_k(v_c, s)\right), \quad (6)$$

where v_c is the $(n+1)$ -gram occurring in the sentence v , $f_k(v_c, s)$ are feature functions, and $Z(s) = \sum_{v \in \mathcal{V}(s)} \exp(\sum_c \sum_k \lambda_k f_k(v_c, s))$.

For example, we can convert the PPoNs into the conditional form by setting

$$f_k(v_{t-n:t}, s) = \log P_n(v_t | v_{t-n}, \dots, v_{t-1}). \quad (7)$$

In our study of accent prediction, we do not use the accentless input s in the feature function. The n -grams are used features instead

$$f_k(v_{1:n}, s) = \delta(C(v_{1:n}) > \tau), \quad (8)$$

where $\tau \geq 0$ is the threshold for the number of occurrences $C(.)$, and $\delta(.)$ is the indicator function. The thresholding is important to reduce overfitting and to reduce the number of features significantly because the majority of n -grams appear only once in the corpus.

4.2 Learning CRF Models

Given D training sentences, we learn the parameters of CRF models by maximising the regularised likelihood

$$\hat{\lambda} = \arg \max_{\lambda} \mathcal{L}(\lambda), \quad (9)$$

where

$$\mathcal{L}(\lambda) = \frac{1}{D} \sum_{i=1}^D \log P(v^{(i)}|s) - \frac{\|\lambda\|^2}{2\sigma^2}. \quad (10)$$

In this study we are interested in online-learning in which parameters are updated after each sentence. This is important in interactive applications where the system may output several possible alternatives and let the user select the one which is most appropriate to his context. Letting the user correct the prediction will allow the system to gradually adapt the model to the domain.

In this study we investigate two online-learning strategies. The first one uses the stochastic gradient ascent [11] to update the parameter as soon as it see the i th sentence. We call this strategy by online maximum likelihood (ML). To smooth the update with fast learning rates and to control the overfitting at the same time, we add a Gaussian regularisation term with mean 0 and variance σ to the likelihood:

$$\bar{\mathcal{L}}(\lambda) = \mathcal{L}(\lambda) - \frac{\|\lambda\|^2}{2\sigma^2}. \quad (11)$$

This term basically prevents the weight from being too large, and thus it reduces the tendency of the model to fit the training data too well. It also encourages small parameter changes after seeing each sentence, which is crucial for the stability of the algorithm. The parameter update thus becomes

$$\lambda \leftarrow \lambda + \alpha_i \{\nabla \mathcal{L}(\lambda)\} - \frac{\lambda}{\sigma^2}. \quad (12)$$

where $\alpha_i > 0$ is the learning rate. We set $\sigma = 10$ and $\alpha_i = 0.1$ through empirical trials.

The second strategy is based on the Structured Perceptron [2] and is given as

$$\lambda_k \leftarrow \lambda_k + \{f_k(v^i) - f_k(\hat{v}^i)\}, \quad (13)$$

where \hat{v} is from Eq. 2. Note that the Structured Perceptron method does not estimate the maximum likelihood but minimises the classification errors. In our implementation, at each step, a sentence is randomly selected from the corpus. We then compute the final parameter by averaging over the parameters learnt after each pass through all data points.

The details of computing the log-likelihood, the optimal \hat{v} , and the gradients required for CRF learning are presented in Appendix for clarity.

5 Evaluations

5.1 Corpus and Processing

We collect data from Vietnamese online news sources, split it into a training set of 426K sentences and a test set of 28K sentences. The corpus contains a wide

range of subjects worth reporting⁵. The writing styles vary since the materials come from a dozen news sources.

In order to effectively deal with foreign words, acronyms and non-alphabets, we consider only accentless terms which may correspond to some Vietnamese terms. To obtain the accentless vocabulary, we de-accent the terms in a Vietnamese dictionary. The accentless vocabulary has 1.4K accentless terms, which is much smaller than the typical Vietnamese set of terms (around 10K)⁶. Through de-accenting we obtain the proposal sets, each of which is a set of Vietnamese terms corresponding to a particular accentless term. The number of Vietnamese terms which share the same accentless form ranges from 1 to 24, and is about 4 on average.

For testing, we first perform de-accenting to obtain the accentless form and then decode back the accented form. The decoded text is compared against the original. Here we do not distinguish between upper-case and lower-case. The learning and comparison are done in lower-case.

The performance is measured within the accentless vocabulary. The term accuracy is the portion of restored terms that are correct. A restored sentence is considered correct if all of its restored terms (within the accentless vocabulary) are correct.

From the training data we estimate the unigram, bigram and trigram distributions. There are 7K unique unigrams whose accentless form is in the accentless dictionary. We count a bigram if it occurs and one of the component unigrams is in the unigram list. We obtain a bigram list of size 842K. If we remove those bigrams that happen only once in the corpus, the list is reduced to 465K. Similarly, we count a trigram if it occurs and one of the component unigrams is in the unigram list. This gives 3137K unique trigrams. Removing the trigrams with a single occurrence we obtain a trigram list of size 1264K. We then apply Laplace smoothing, where vocabulary sizes for unigrams, bigrams and trigrams are estimated to be 10^4 , 7×10^8 , and 7×10^{13} , respectively. The first estimate is from the 7K unigrams we obtain from the corpus. To estimate the second, recall that the bigrams have two components, one of them must be Vietnamese, and one can be non-Vietnamese. We estimate 10^5 unique non-Vietnamese unigrams in the components of the 842K bigrams from the corpus. Multiplying with the 7K Vietnamese unigrams we obtain 7×10^8 . Similarly, multiplying this with 10^5 to obtain the estimate for the trigram vocabulary size.

5.2 Results

For the Powered Product-of- N -grams method described in Section 3, we do not optimise the feature weights $\lambda_1, \lambda_2, \lambda_3$ corresponding to three component models of unigram, bigram and trigram, respectively. Rather, we set the weight manually, and obtain good performance with:

⁵ They are: politics, social issues, IT, family & life-style, education, science, economics, legal issues, health, world, sports, arts & culture, and personal opinions.

⁶ Note that the evaluation is done *after* restoration, that is, we still use all Vietnamese terms for evaluation.

- first-order PPoNs (unigram & bigram): $w_1 = 1; w_2 = 1$,
- second-order PPoNs (unigram & bigram & trigram): $w_1 = 2; w_2 = 2; w_3 = 1$

First we perform a set of experiments with first-order models, which include the bigrams, the first-order PPoNs and the first-order CRF. One problem with the bigram model is how it handles the unseen bigrams. As the majority of the Vietnamese words used in writing are bigrams, this means that a bigram is not simply random combination of two unigrams. Therefore, most random combinations of two unigrams should have extremely low probability, or at least their probabilities are not equal. The popular Laplace smoothing, on the other hand, tries to assign every unseen bigram an equally small probability under the assumption of prior uniform distribution. This is unrealistic in Vietnamese. To deal with this we assign unseen bigrams with a very low probability, which is practically zero, so that any sequence with unseen bigrams is severely penalised. Although this is not optimal since some plausible bigrams are cut off, it seems to solve the problem. Luckily, the PPoNs does not have this problem, probably because the unseen bigrams will be compensated by the component unigrams.

In the first-order CRF model, we use only the bigram features. For training, we run the Structured Perceptron for 20 iterations and the online ML for 15 iterations over the whole training data set. This is obviously much slower than the bigram and first-order PPoNs models since we need to estimate the bigram distribution using only one run through the data. However, such a cost can be well justified by the higher performance of the CRF model compared with the bigram and the PPoNs as shown in Table 1. In this study, we use 465K bigram features for all the first-order models. The simple unigram model works poorly as expected, and its performance is unacceptable for practical use. The PPoNs, which is just a product of the unigram and the bigram models, works surprisingly well with significant improvement over the bigram model. The CRF model is the winner despite the fact that it uses no more information than that for the PPoNs.

The second set of experiments is performed on second-order models. For the moment, only the second-order PPoNs is used with 7K unigram features, 465K bigram features, and 1264K trigram features. We run the Structured Perceptron

Table 1. Term and sentence accuracy (%) of first/second-order models compared with the baseline unigram model

Model	Term accuracy	Sentence accuracy
Baseline	71.8	6.3
Bigram	90.7	30.9
1st-order PPoNs	92.4	37.6
1st-order CRF (Structured Perceptron)	93.2	38.4
1st-order CRF (Online ML)	93.7	42.0
2st-order PPoNs	93.5	42.7
2st-order CRF (Structured Perceptron)	93.5	41.8
2st-order CRF (Online ML)	94.3	44.8

for 10 iterations and the online ML for 5 iterations. The last rows in Table 1 show the accuracy of the PPoNs and the CRF. The trigram model is not used since it performs fairly poorly, possibly due to the limited corpus. Interestingly, the PPoNs can compensate the poor estimate of the trigrams by using the unigram and bigram components.

Overall for both experiment sets, the CRF trained by online ML performs best. We observe that the Structured Perceptron minimises the error over *training* data quickly since it is specifically designed for this task. The PPoNs, although not as good as the CRF, provides a simple and fast method for model estimation.

A online prototype is available for evaluation at [<http://vietlabs.com>].

6 Discussion

This paper evaluates a set of constrained sequence classification methods with a particular application in accent prediction. The idea of constrained inference in the context of sequence classification has been proposed earlier [6] in which in the prediction phase some of the labels in the sequence (e.g. accents in our case) are deterministically known. Our work can be considered as an extension to this because we consider a subset of labels as constraints, and use the constraints in both learning and prediction phases. We conjecture that the constraints used in learning can produce better a *conditional* language model for the given restoration task.

Although experimental results on Vietnamese so far indicate that the approach is suitable and can achieve high quality in online news domains, there are open rooms for further improvement. First, there are different genres and writing styles, and it is likely that a sequence of accentless terms can correspond to several plausible Vietnamese sequences, depending on the context of use. A very challenging domain is creative writing, especially in poetry, where the authors make deliberate use of word reordering and repetition to achieve stylistic and artistic effect. The most challenging form is perhaps spoken language, especially in the online environments such as chatting and SMS, where the use of language is largely distorted due to the constraints of writing space and personal interests.

An issue not addressed in this work is the analysis of syntax and semantics. It is likely that the analysis will provide more consistent and grammatical results as well as coherence within and between sentences in the document. Through the CRF framework, for example, it is possible to incorporate a richer set of features to address the correlation between sentences in the same paragraph. Also, we can create different models to address different linguistic aspects and then combine them together in the PPoNs approach.

References

1. Berger, A.L., Della Pietra, S.A., Della Pietra, V.J.: A Maximum Entropy approach to natural language processing. *Computational Linguistics* 22, 39–71 (1996)
2. Collins, M.: Discriminative training methods for hidden Markov models: Theory and experiments with the perceptron algorithm. In: Conference on Empirical Methods in Natural Language Processing (EMNLP) (2002)

3. Dien, D., Kiem, H., Toan, N.V.: Vietnamese Word Segmentation. In: NLPRS 2001, pp. 749–756 (2001)
4. De Pauw, G., Wagacha, P.W., de Schryver, G.M.: Automatic Diacritic Restoration for Resource-Scarce Languages. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629. Springer, Heidelberg (2007)
5. Golding, A.R., Roth, D.: Winnow-Based Approach to Context-Sensitive Spelling Correction. Machine Learning 34, 107–130 (1999)
6. Kristjannson, T., Culotta, A., Viola, P., McCallum, A.: Interactive information extraction with constrained conditional random fields. In: 19th National Conference on Artificial Intelligence (AAAI), pp. 412–418 (2004)
7. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: International Conference on Machine learning (ICML), pp. 282–289 (2001)
8. Mihalcea, R., Nastase, V.: Letter level learning for language independent diacritics restoration. In: International Conference On Computational Linguistics, pp. 1–7 (2002)
9. Nelken, R., Shieber, S.M.: Arabic Diacritization Using Weighted Finite-State Transducers. In: ACL Workshop on Computational Approaches to Semitic Languages (2005)
10. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 257–286 (1989)
11. Robbins, H., Monro, S.: A stochastic approximation method. The Annals of Mathematical Statistics 22, 400–407 (1951)
12. Simard, M., Deslauriers, A.: Real-time automatic insertion of accents in French text. Natural Language Engineering 7, 143–165 (2001)
13. Wagacha, P., De Pauw, G., Githinji, P.: A grapheme-based approach for accent prediction in Gikuyu. In: 5th International Conference on Language Resources and Evaluation, Genoa, Italy, pp. 1937–1940 (2006)
14. Yarowsky, D.: A comparison of corpus-based techniques for restoring accents in Spanish and French text. In: Natural Language Processing Using Very Large Corpora. Springer, Heidelberg (1999)
15. Zitouni, I., Sorensen, J.S., Sarikaya, R.: Maximum Entropy based restoration of Arabic diacritics. In: 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL, pp. 577–584 (2006)
16. Zweigenbaum, P., Grabar, N.: Restoring accents in unknown biomedical words: application to the French MeSH thesaurus. International Journal of Medical Informatics 67, 113–126 (2002)

A Constrained Inference

In this appendix, we provide a general account for inference in first and second-order models with constrained state spaces.

Inference in the n -gram models is mostly Viterbi decoding (as in Eq. 2) since maximum likelihood learning of such models is done through frequency counting. This also applies for the PPoNs method since we do not perform further learning after estimating the n -gram components. However, inference in the CRF is needed for estimating the partition function (as in Eq. 6) and the feature expectation as shown in Section 4.2.

A.1 First-Order Models

Associated with each node t in the first-order Markov chain is a positive potential $\phi(v_t, s)$ to account for the statistics of the distorted term v_t given the input s . Similarly, associated with each edge is a positive potential $\psi(v_{t-1}, v_t, s)$ to account for the statistics of the transition from v_{t-1} to v_t . The correspondence between these potentials and the proposed models are as follows

- In the unigram model, $\phi(v_t, s) = P(v_t)$,
- In the bigram model, $\phi(v_1, s) = P(v_1)$, $\phi(v_t) = 1$ for $t > 1$, and $\psi(v_{t-1}, v_t, s) = P(v_t | v_{t-1})$,
- In the bigram PPoNs model, $\phi(v_t, s) = P(v_t)^{\lambda_1}$, $\psi(v_{t-1}, v_t, s) = P(v_t | v_{t-1})^{\lambda_2}$, and
- In the first-order CRF model, $\phi(v_t, s) = \exp(\lambda_k f_k(v_t, s))$, and $\psi(v_{t-1}, v_t, s) = \exp(\lambda_k f_k(v_{t-1}, v_t, s))$.

For the first-order Markov chains, inference can be done using the forward-backward procedure. The forward α_t is defined recursively as

$$\alpha_t(v_t \in \mathcal{V}(s_t) | s) \propto \sum_{v_{t-1} \in \mathcal{V}(s_{t-1})} \alpha_{t-1}(v_{t-1} | s) \phi(v_{t-1}, s) \psi(v_{t-1}, v_t, s), \quad (14)$$

where $\alpha_1(v_1 \in \mathcal{V}(s_1) | s) = 1$; Similarly, the backward β_t is

$$\beta_t(v_t \in \mathcal{V}(s_t) | s) \propto \sum_{v_{t+1} \in \mathcal{V}(s_{t+1})} \beta_{t+1}(v_{t+1} | s) \phi(v_{t+1}, s) \psi(v_t, v_{t+1}, s). \quad (15)$$

For the feature expectations in CRFs (as in Eq. 12), we need to compute the marginal

$$P(v_t | s) \propto \alpha_t(v_t | s) \beta_t(v_t | s) \phi(v_t, s), \quad (16)$$

and the joint marginals

$$P(v_t, v_{t+1} | s) \propto \alpha_t(v_t | s) \beta_t(v_{t+1} | s) \phi(v_t, s) \phi(v_{t+1}, s) \psi(v_t, v_{t+1}, s). \quad (17)$$

The complexity of these procedures is therefore $\mathcal{O}(T|S|^2)$ where $|S| = \max_t |\mathcal{V}(s_t)|$.

To do restoration we can use the Viterbi decoding [10], paying attention to the constraints. Alternatively, we can use the equivalent Pearl’s max-product algorithm where the summations in Eqs. 14 and 15 are replaced by the maximisations. Similar to the forward-backward, this max-product algorithm takes $\mathcal{O}(T|S|^2)$ time.

A.2 Second-Order Model

Now we need to take the trigrams into account by using the extension of the edge potential $\psi(v_{t-1}, v_t, s)$ to incorporate the state v_{t-2} . With a slight abuse of notation, denote by $\psi(v_{t-2}, v_{t-1}, v_t, s)$ the trigram potentials.

- In the trigram model, $\psi(v_{t-2}, v_{t-1}, v_t, s) = P(v_t | v_{t-1}, v_{t-2})$,
- In the bigram PPoNs model, $\psi(v_{t-2}, v_{t-1}, v_t, s) = P(v_t | v_{t-1}, v_{t-2})^{\lambda_3}$, and
- In the second-order CRF model, $\psi(v_{t-2}, v_{t-1}, v_t, s) = \exp(\lambda_k f_k(v_{t-2}, v_{t-1}, v_t, s))$.

Efficient inference in the second-order Markov chains is a bit more tricky since we do not have the chains. First, we need to convert the second-order Markov chains into the first-order equivalence. The conversion is done by joining two successive nodes $\{v_{t-1}, v_t\}$ into a composite-node $V_{t-1} = \{v_{t-1}, v_t\}$. Let the composite-node potential be $\phi(V_{t-1}, s) = \phi(v_{t-1})\psi(v_{t-1}, v_t, s)$ and the composite-edge potential be $\psi(V_{t-1}, V_t, s) = \psi(v_{t-1}, v_t, v_{t+1}, s)$. Given these potentials, it can be seen that we now have a new first-order Markov chain with the combined state space:

$$V_{t-1} \in \mathcal{V}(s_{t-1}, s_t) = \mathcal{V}(s_{t-1}) \times \mathcal{V}(s_t) . \quad (18)$$

The naïve implementation of this Markov chain takes $\mathcal{O}((T-1)|S|^4)$ time in this combined state space. However, by paying attention to the fact that the two composite-states $V_{t-1} = \{v_{t-1}, v_t\}$ and $V_t = \{v_t, v_{t+1}\}$ share the same term v_t , we can implement the forward-backward procedure in $\mathcal{O}((T-1)|S|^3)$ time using

$$\begin{aligned} \alpha_t(V_t | s) &= \sum_{x_{t-1} \in \mathcal{V}(x_{t-1})} \alpha_{t-1}(V_{t-1} | s) \phi(V_{t-1}, s) \psi(V_{t-1}, V_t, s) , \\ \beta_t(V_t | s) &= \sum_{x_{t+2} \in \mathcal{V}(x_{t+2})} \beta_{t+1}(V_{t+1} | s) \phi(V_{t+1}, s) \psi(V_t, V_{t+1}, s) , \end{aligned}$$

and the joint marginals are computed as

$$\begin{aligned} P(V_t | s) &\propto \alpha_t(V_t | s) \beta_t(V_t | s) \phi(V_t, s) , \\ P(V_t, V_{t+1} | s) &\propto \alpha_t(V_t | s) \beta_t(V_{t+1} | s) \phi(V_t, s) \phi(V_{t+1}, s) \psi(V_t, V_{t+1}, s) . \end{aligned}$$

Map Building by Sequential Estimation of Inter-feature Distances

Atsushi Ueta, Takehisa Yairi, Hirofumi Kanazaki, and Kazuo Machida

The University of Tokyo

Komaba 4-6-1, Meguro-ku, Tokyo, 153-8904 Japan

{uetu,yairi,kanazaki,machida}@space.rcast.u-tokyo.ac.jp

Abstract. This paper proposes an alternative solution to a mapping problem in two different cases; when bearing measurement to features (landmarks) and odometry are measured and when local position of features are measured. Our approach named M-SEIFD (Mapping by Sequential Estimation of Inter-Feature Distances) first estimates inter-feature distances, then finds global position of all features by enhanced multi-dimensional scaling (MDS). M-SEIFD is different from the conventional SLAM methods based on Bayesian filtering in that robot self-localization is not compulsory and that M-SEIFD is able to utilize prior information about relative distances among features directly. We show that M-SEIFD is able to achieve a decent map of features both in simulation and in real-world environment with a mobile robot.

Keywords: Mapping, SLAM, mobile robot, multi-dimensional scaling.

1 Introduction

Most of the studies on mobile robot map building have employed the problem formulation of *simultaneous localization and mapping* (SLAM) which states the problem of estimating both robot states and feature positions from a series of sensor measurements. Recently, a number of studies (e.g.[1],[2],[3]) have been made on Bearing-only SLAM (BOSLAM), which is a problem of SLAM using only relative bearing measurements to the features and odometry information of the robot's motion. This is because there is a demand for building affordable robots with low cost sensors such as monocular cameras which provide only bearing data related to environmental features. A conventional approach to BOSLAM can be described as follows. First, prepare motion and measurement models which contain robot's pose and positions of all features as the state variables, and relative bearing measurements to the features as the observation variables. Then, apply some Bayesian filtering method such as Extended Kalman Filter (EKF) to update sequentially the estimates of the state variables.

On the other hand, we propose an alternative solution to the bearing-only mapping problem (we call it *Bearing-odometry mapping* in this paper : BOM), which is named Mapping by Sequential Estimation of Inter-Feature Distances

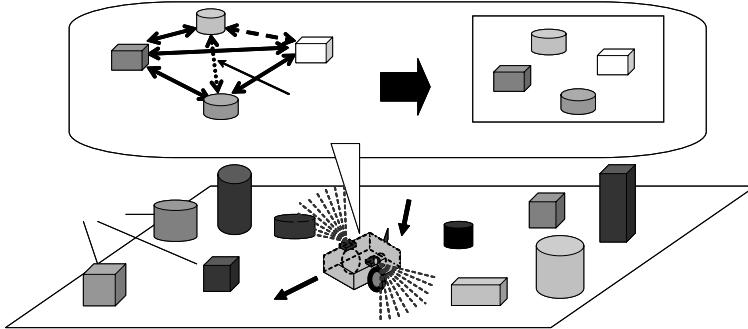


Fig. 1. Assumed map building task of a mobile robot

(M-SEIFD). In M-SEIFD, the robot first attempts to estimate the relative distances among the features sequentially using the bearing measurements and odometry information in each time step, then reconstructs the coordinates of the features by applying enhanced versions of multi-dimensional scaling (MDS) to the matrix of inter-feature distances (Fig. 1). This two step estimation procedure is significantly different from the ordinary SLAM methods which estimate the feature locations and robot pose directly from the measurements. This is the main point of this paper. Our proposed method is to estimate inter-feature distances which are independent of any coordinate system, so it is easy to integrate measurements which are obtained in different coordinate systems or places. This feature-based method can also be applied in the case when feature-positions in robot local frame are measured (*Bearing-range mapping* : BRM). In this case, relative distances among the features are computed directly from local positions of features without odometry data. Another remarkable feature of M-SEIFD is that robot self-localization is not compulsory (but optional; in the previous work, this feature-based method achieved reasonable accuracy in self localization[4]), whereas mapping and localization are inseparable in the conventional methods. The rest of this paper is organized as follows. In Sect. 2, we will briefly review the related work on mobile robot map building including BOSLAM. In Sect. 3, we will describe the principles of the proposed method, such as sequential estimation of inter-feature distances, updates of inter-feature distance matrix, and reconstruction of feature coordinates by MDS. In Sect. 4 and 5, we will demonstrate how the proposed method actually works by simulations and experiments using a real robot. Finally, in Sect. 6, we will conclude with a summary and future works.

2 Related Work

2.1 SLAM

In the last decade, SLAM (simultaneous localization and mapping) based on the probabilistic modeling (or state space models) and Bayesian filtering has been the

mainstream of mobile robot map building research[5]. In the general framework of SLAM, motion and measurement models containing robot's pose and features' positions as state variables are prepared beforehand. Then, those state variables are sequentially estimated from obtained measurements by applying Bayesian inference techniques to the models. As is widely known, several approaches have been developed for solving this SLAM problem, including EKF[6], alternate estimation by EM algorithm[7], and Rao-Blackwellized particle filter[8].

Recently, Bearing-only SLAM (BOSLAM) has drawn much attention in this field, mainly due to the requirement or performing SLAM only with inexpensive sensors such as monocular cameras (e.g.[1],[2],[3]). Although BOSLAM poses several challenges such as landmark (feature) initialization problem because of the insufficient information of single bearing measurement, it is still based on the ordinary SLAM framework above. Actually, Bekris et al.[3] have compared a variety of existing SLAM techniques in BOSLAM setting, and reported that Rao-Blackwellized particle filters are faster and more robust than others. Moreover, a solution based on the sparse least squares problem has lately been proposed[9].

2.2 Embedding Approach

The SLAM framework, as seen above, inherently focuses on the spatial relationships between the robot's poses and features' locations. In contrast, there are other approaches to the map building problem focusing on the spatial relationships among the features themselves. In other words, they attempt to construct a global map by merging and embedding pieces of information of local spatial relationships among features which are obtained by observations. Intuitively, this process is similar to putting puzzle pieces together into a picture. For example, *local map alignment* by Lu and Millios[10] and *relaxation* by Duckett et al.[11] are representative map building techniques based on this idea.

On the other hand, the authors have proposed methods of building feature-based maps by applying multi-dimensional scaling (MDS) and related techniques to the inter-feature distance matrices which are estimated from covisibility[12] and similarity of observation history[13]. While these methods also focus on the relationships among features rather than robot-features relationships, they are more distinct from SLAM in that they do not require robot localization.

The proposed method in this paper can be regarded as a special case of the authors' previous studies in which the inter-feature distance matrix is estimated by quantitative measurements.

3 Proposed Method: M-SEIFD

In this section, we describe the proposed method named Mapping by Sequential Estimation of Inter-Feature Distances (M-SEIFD).

3.1 Problem Definition and Outline

Fig. 2 illustrates an *ideal* relationship among measurements and feature positions, assuming there are no measurement errors in $\{l_t, \phi_t, \theta_{i,t-1}, \theta_{j,t-1}, \theta_{i,t}, \theta_{j,t}\}$

(a), and $\{\mathbf{x}_{F_i}^{R_t}, \mathbf{x}_{F_j}^{R_t}\}$ (b). Note that these measurements are subject to errors and sometimes missing in the real environment.

Bearing-odometry mapping (BOM) is defined as a problem of finding 2-D coordinates of all features $\{\mathbf{x}_{F_i}\}_{i=1,\dots,N}$ for given measurements $\{l_t, \phi_t, \theta_{i,t}\}_{i=1,\dots,N}^{t=1,\dots,T}$ up to time T; whereas given measurements are $\{\mathbf{x}_{F_i}^{R_t}, \mathbf{x}_{F_j}^{R_t}\}_{i < j=2,\dots,N}^{t=1,\dots,T}$ in Bearing-range mapping (BRM). In conventional SLAM methods, not only features' positions but also robot's position at each time step $\{\mathbf{x}_{R_t}\}_{t=1,\dots,T}$ is required to be estimated.

The process of map building by M-SEIFD is summarized as follows:

Step 1. First, in *Inter-Feature Distance Estimation* step (3.2), for each pair of features F_i, F_j , an estimate of squared distance $\hat{d}_{F_i, F_j|t}^2$ with its variances $\hat{\sigma}_{F_i, F_j|t}^2$ is computed from two successive bearing measurements to the features $\{\theta_{i,t-1}, \theta_{j,t-1}, \theta_{i,t}, \theta_{j,t}\}$, and odometry readings $\{l_t, \phi_t\}$ in BOM, or from a single local feature-location measurements $\{\mathbf{x}_{F_i}^{R_t}, \mathbf{x}_{F_j}^{R_t}\}$ in BRM.

Step 2. Next, in *Distance Update* step (3.3), for each pair of features, $\hat{d}_{F_i, F_j|t}^2$ is merged with $\hat{d}_{F_i, F_j|1:t-1}^2$, and a new estimate $\hat{d}_{F_i, F_j|1:t}^2$ is obtained.

Step 3. Finally, in *Coordinates Reconstruction* step (3.4), estimates of feature positions $\{\mathbf{x}_{F_i|1:t}\}_{i=1,\dots,N}$ are found by applying MDS to the set of estimated inter-feature squared distances $\{\hat{d}_{F_i, F_j|1:t}^2\}_{i,j=1,\dots,N}$.

In the rest of this section, we will explain these processes in detail.

3.2 Inter-feature Distance Estimation

In Fig. 2(a), we assume there is no measurement error and the location of a feature F_i in this local coordinate system can be represented in terms of two successive bearing measurements $\theta_{i,t-1}, \theta_{i,t}$ and odometry readings l_t, ϕ_t as,

$$\mathbf{x}_{F_i}^{R_t} = \frac{\sin(\phi_t + \theta_{i,t}) \cdot l_t}{\sin(\phi_t + \theta_{i,t} - \theta_{i,t-1})} [\cos \theta_{i,t-1}, \sin \theta_{i,t-1}]^T \quad (1)$$

This is the well-known principle of triangulation. Using this equation, distance between two arbitrary features d_{F_i, F_j} can be easily computed. In BRM, the distance is computed directly from measurements, $\mathbf{x}_{F_i}^{R_t}$. As actual measurements are subject to measurement errors, we denote the instantaneous estimate of the squared distance at time t by $\hat{d}_{F_i, F_j|t}^2$. And we consider the estimated variance of $\hat{d}_{F_i, F_j|t}^2$ which is denoted by $\hat{\sigma}_{F_i, F_j|t}^2$. If the covariance matrix of the measurement vector is given as $\Sigma_{i,j,t}$, $\hat{\sigma}_{F_i, F_j|t}^2$ can be approximated to the first order using the Jacobian $J \equiv [\frac{\partial(\hat{d}_{F_i, F_j|t}^2)}{\partial l_t}, \frac{\partial(\hat{d}_{F_i, F_j|t}^2)}{\partial \phi_t}, \frac{\partial(\hat{d}_{F_i, F_j|t}^2)}{\partial \theta_{i,t-1}}, \dots]$ as,

$$\hat{\sigma}_{F_i, F_j|t}^2 = J \Sigma_{i,j,t} J^T \quad (2)$$

Intuitively, the variance $\hat{\sigma}_{F_i, F_j|t}^2$ implies the magnitude of estimation error in $\hat{d}_{F_i, F_j|t}^2$. It should be noted that an estimated distance between features is

independent of the local coordinate system or trajectory of the robot. Thanks to this property, the update process of the distances becomes easy as explained later.

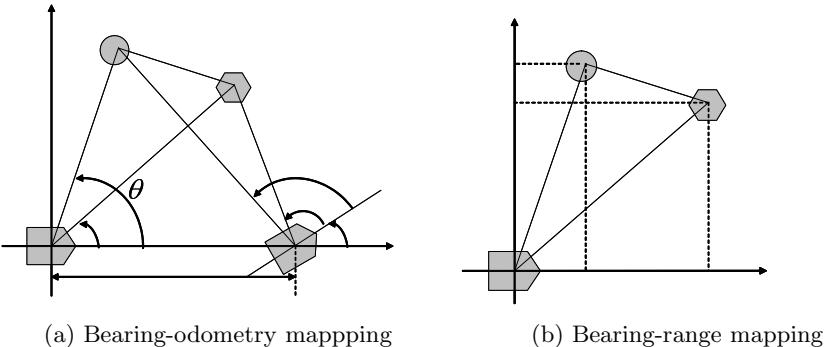


Fig. 2. Ideal relationships among measurements and positions of features and robot

3.3 Update of Distance Estimates

We assume that the estimation error of a squared distance between features at each time step $e_{i,j|t} = \hat{d}_{F_i,F_j|t}^2 - d_{F_i,F_j}^2$ ($t = 1, \dots, T$) is independent of each other. We denote the estimates of squared distance of a pair of features and its variance estimated taking all measurements up to time t by $\hat{d}_{F_i,F_j|1:t}^2$ and $\hat{\sigma}_{F_i,F_j|1:t}^2$, respectively.

Now consider an instantaneous distance estimate $\hat{d}_{F_i,F_j|t+1}^2$ and its variance $\hat{\sigma}_{F_i,F_j|t+1}^2$ are obtained at the next time step $t + 1$. Then a reasonable update rule of $\hat{d}_{F_i,F_j|1:t+1}^2$ and $\hat{\sigma}_{F_i,F_j|1:t+1}^2$ are given as below:

$$\hat{\sigma}_{F_i,F_j|1:t+1}^2 = (\hat{\sigma}_{F_i,F_j|1:t}^{-2} + \hat{\sigma}_{F_i,F_j|t+1}^{-2})^{-1} \quad (3)$$

$$\hat{d}_{F_i,F_j|1:t+1}^2 = \hat{\sigma}_{F_i,F_j|1:t+1}^2 \cdot (\hat{\sigma}_{F_i,F_j|1:t}^{-2} \cdot \hat{d}_{F_i,F_j|1:t}^2 + \hat{\sigma}_{F_i,F_j|t+1}^{-2} \cdot \hat{d}_{F_i,F_j|t+1}^2) \quad (4)$$

This update rule is optimal in the sense that it minimizes the estimated variance $\hat{\sigma}_{F_i,F_j|1:t+1}^2$.

When prior information on the relative distances between a specific pair of features is given, it can be utilized directly by setting the initial estimates $\hat{d}_{F_i,F_j|0}^2$ and $\hat{\sigma}_{F_i,F_j|0}^2$ properly.

Now we define the *estimated squared distance matrix* (ESDM) $\Delta_{F|1:t}$ whose (i, j) element is given by $\hat{d}_{F_i,F_j|1:t}^2$. That is to say,

$$\Delta_{F|1:t} \equiv \begin{bmatrix} 0 & \hat{d}_{F_1,F_2|1:t}^2 & \cdots & \hat{d}_{F_1,F_N|1:t}^2 \\ \hat{d}_{F_1,F_2|1:t}^2 & 0 & \cdots & \hat{d}_{F_2,F_N|1:t}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \hat{d}_{F_1,F_N|1:t}^2 & \hat{d}_{F_2,F_N|1:t}^2 & \cdots & 0 \end{bmatrix} \quad (5)$$

3.4 Coordinates Reconstruction by Multi-dimensional Scaling

The problem of finding coordinates of *items* in a low dimensional space given a matrix of dissimilarities between them is known as *multi-dimensional scaling* (MDS), and a variety of techniques to solve this problem have been developed[14]. In our case, if the squared distances between all pairs of features are given *classical scaling* which is the most famous and fundamental MDS technique can be used. An advantage of using classical scaling is that its solution is guaranteed to globally minimize a loss function called *Strain*. A drawback of classical scaling, on the other hand, is that it is not able to deal with missing elements in the dissimilarity matrix directly. Obviously, this is a serious problem for us, because in the real environment it is very likely that some part of the relative distances between the features are not obtained directly from the measurements due to various constraints. For example, the distance between two features which are very far from each other or occluded by an obstacle is likely to be missing or too inaccurate.

To overcome these obstacles, we propose two approaches, (1) completion of missing or inaccurate elements in the distance matrix by shortest path lengths, and (2) use of another MDS technique called SMACOF. We will explain them in detail.

(1) Distance Matrix Correction by Shortest Path Length

The idea of the first approach is to apply the ordinary classical scaling to a *repaired* distance matrix $\Delta'_{F|1:t}$ whose missing elements are completed by approximated values using available inter-feature distance information. More specifically, we approximate the missing distance between each pair of features by the shortest path length from one to the other. The shortest paths among features can be found efficiently by Floyd-Warshall algorithm. We call this technique Distance Matrix Correction by Shortest Path Length (DMC-SPL) in this paper. DMC-SPL is basically the same technique introduced in ISOMAP[15] which is a non-linear dimensionality reduction method. An advantage using DMC-SPL is that the global optimum solution is always guaranteed once the estimated squared distance matrix is corrected. A major drawback of DMC-SPL is that it is an off-line algorithm because it needs to be recomputed every time a new estimate is obtained.

(2) SMACOF with Weighted Distance Matrix

In this study, we chose SMACOF algorithm[16] among a number of MDS methods using loss functions of this type, because of its performance and efficiency[12]. In SMACOF, the loss function named *raw stress* as below is locally minimized by iterative *majorization* technique.

$$L_{\text{sma}}(\mathbf{X}) = \sum_{i < j} w_{i,j} (\hat{d}_{F_i, F_j | 1:t} - \| \hat{x}_{F_i} - \hat{x}_{F_j} \|)^2 \quad (6)$$

Compared with DMC-SPL (together with classical scaling), it is easy to modify SMACOF into an on-line algorithm as it is inherently an iterative process. A

critical issue for SMACOF, however, is that it depends on initial solution whether it finds only a local minimum.

In the later experiment, we applied SMACOF to the estimated distance matrix with an initial solution which is computed by the classical scaling together with DMC-SPL.

4 Simulation Experiments

We conducted several experiments to evaluate the proposed method in two different cases, *Bearing-odometry mapping* and *Bearing-range mapping* in a simulated environment.

4.1 General Settings

The simulated environment is a square region whose side length is 2.5[m] containing N=50 randomly placed features. At each time step, the robot moves to the next position according to the randomly chosen l_t and ϕ_t , then obtains a set of relative bearing measurements to recognizable features $\{\theta_{i,t}\}$ in BOM, or a set of local feature-location measurements $\{x_{F_i}^{R_t}\}$ in BRM. In this experiment, we simulated the observation uncertainty by adding Gaussian noises to the ideal measurements of $\{l_t, \phi_t, \theta_{i,t}\}$ or $\{x_{F_i}^{R_t}, y_{F_i}^{R_t}\}$. The standard deviations of the noises are $\{\sigma_l, \sigma_\phi, \sigma_\theta, \sigma_x, \sigma_y\} = \{0.03[\text{m}], 3[\text{deg}], 3[\text{deg}], 0.03[\text{m}], 0.03[\text{m}]\}$. In addition, we assumed that the sensor range is limited and each feature is recognizable only if it is within the distance of 1[m] from the robot position. This means that the estimated squared distance matrix $\Delta_{F,1:t}$ necessarily contains a number of missing elements. Fig. 3(a) shows examples of ground truth map of features. We evaluated the accuracy of estimated positions of the features after applying coordinate transformation of translation, rotation and reflection to the initially obtained map so that it minimizes the sum of squared positional errors of all features.

4.2 Results of BOM: Bearing-Odometry Mapping

The purpose of the first experiment is to examine the performance of M-SEIFD. In this experiment, we compared four methods (following Method 1 and Method 2 and each with prior information) of reconstructing the feature coordinates.

Method 1: Classical scaling with DMC-SPL (3.4-(1)) all the step.

Method 2: Same as Method 1 before 50 steps, then SMACOF with weights (3.4-(2)) afterward.

In this simulation, 10% of all the feature pairs (that is $0.1 \times (\frac{50 \times 49}{2}) = 123$ pairs) were randomly chosen and their distances were given with the accuracy of $\sigma = 3[\text{cm}]$ as prior information in advance. This prior information was used to determine initial distance matrix $\Delta_{F,0}$. Fig. 4(a) shows how *mean position errors* (MPEs) change along with the time. MPEs at $t = 300$ in the two cases

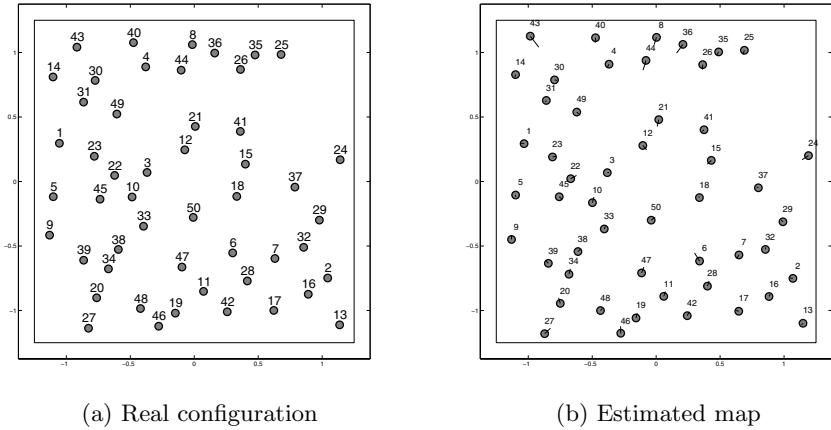


Fig. 3. A real configuration of features (a) and obtained map after 100 steps (b). The differences between estimated and true feature positions are emphasized with thin lines.

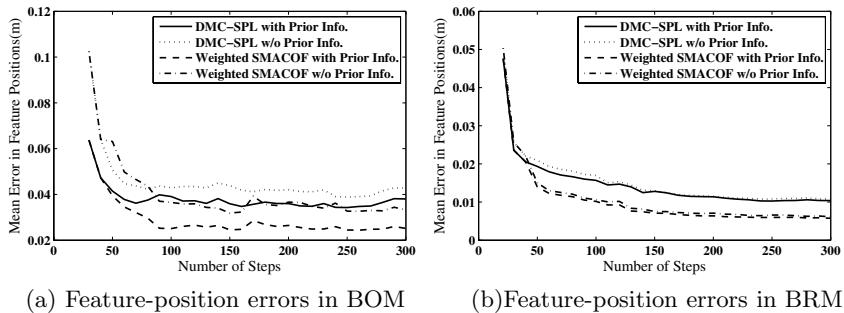


Fig. 4. Mean errors in estimated feature positions in BOM (a) and in BRM (b)

are 4.78[cm] (Method 1) and 2.87[cm] (Method 2), respectively. It is considered that this difference in accuracy is caused mainly by the difference in the loss functions of classical scaling and SMACOF[16]. The result of each method with prior information demonstrated that using this kind of prior knowledge leads to a significant improvement in the estimation accuracy, especially in the early stages.

4.3 Results of BRM: Bearing-Range Mapping

We compared four methods of reconstructing the feature coordinates as well as BOM.

Fig. 4(b) shows how MPEs change along with the time. MPEs at $t = 300$ in the four methods are 1.08[cm] (Method 1), 0.616[cm] (Method 2), 1.04[cm] (Method 1 with prior information) and 0.574[cm] (Method 2 with prior information) respectively. Compared with the results of BOM, maps are estimated with higher accuracy as a whole. This means that M-SEIFD in BRM is subject to

fewer measurement errors and has more information to estimate an inter-feature distance than in BOM. This is because it is measured using four measurements only in a single step without odometry measurement, whereas BOM needs six measurements including odometry measurements from two successive steps.

5 Real Robot Experiments

5.1 General Settings

We used X80 robot produced by Dr Robot inc. (Fig. 5) to verify the proposed approach in real-world environment. It has a cylinder shape with 40[cm] in height and 60[cm] in diameter approximately. The robot has wheel encoders that measure distance traveled and estimate heading. It is equipped with two web cameras, which is well calibrated, placed on the left top and the right top of the robot at about 33[cm] above the floor and each provides 180-degree field of view. The workspace is an $2[\text{m}] \times 2[\text{m}]$ area with 20 features. ARTag markers (the size is $7[\text{cm}] \times 7[\text{cm}]$), or visual marker, are attached to each feature at 45[cm] in height for feature detection. ARTag marker[17] is a 2D visual marker and useful for Augmented Reality (AR), robot navigation and general applications where the relative pose between a camera and a object is required. Features in the image frame are detected by finding tags' edges and checked against the stored template to identify tags and measure rotation and translation between a tag and a camera. The robot was controlled by an operator who drove the robot using a wireless link. It moved approximate 40[cm] in each step. In each measurement step, both cameras were rotated by about 20[deg] in 14 times to achieve 360-degree ranging capability. This experiment consists of 35 steps which enabled the robot to move around twice in the workspace.



Fig. 5. The above is a robot used for our experiments with two cameras attached to both sides. Tags attached to white boxes around the robot are ARTag markers.

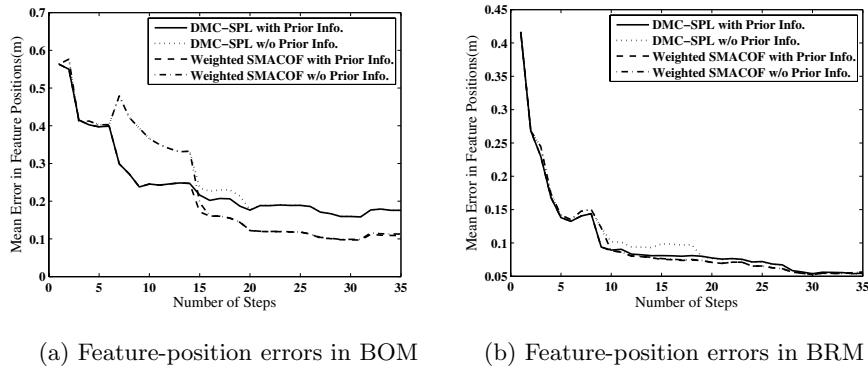


Fig. 6. Mean errors in estimated feature-position in BOM (a) and in BRM (b)

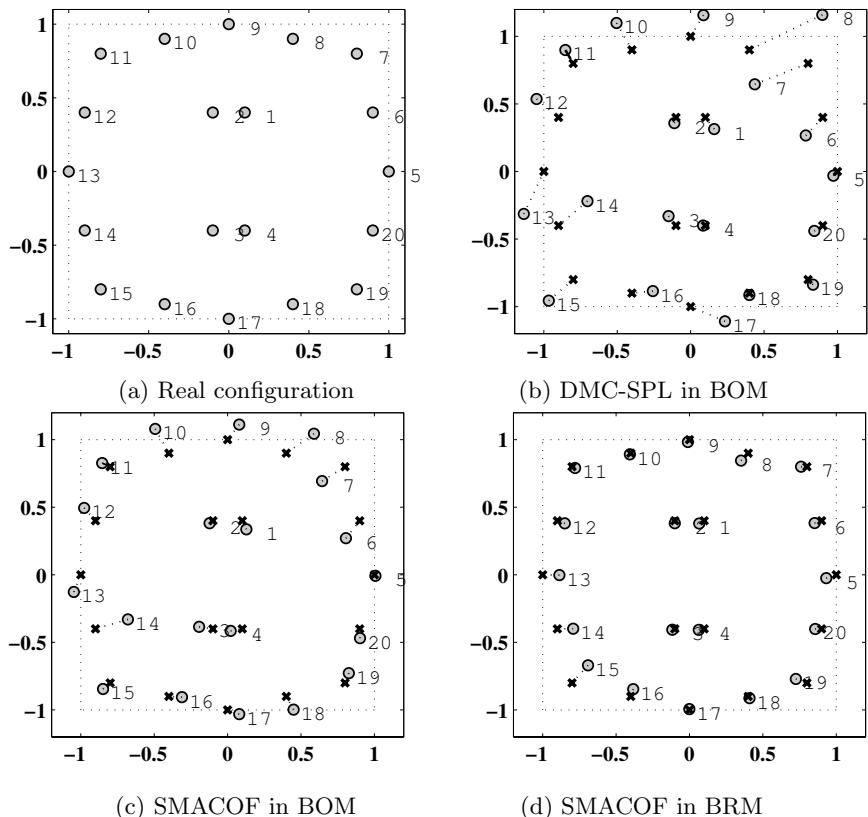


Fig. 7. A real configuration of 20 features (a) and obtained maps by classical scaling with DMC-SPL in BOM (b), SMACOF in BOM (c), and SMACOF in BRM (d)

5.2 Results of BOM: Bearing-Odometry Mapping

We compared four methods of reconstructing the feature coordinates as well as the simulation experiment. In Method 2, it is changed from Method 1 into SMACOF at 10th step. As the prior information, 10 feature pairs (1-2, 3-4, 5-6, ..., 19-20) out of 190 pairs were given. Fig. 6(a) shows how MPEs change as the observation data increases in each method. MPEs at 35th step in each method are 17.6[cm] (Method 1), 11.3[cm] (Method 2), 17.5[cm] (Method 1 with prior information) and 10.9[cm] (Method 2 with prior information) respectively. Fig. 7(b) and (c) illustrates a constructed map after 35 steps by Method 1 and by Method 2, respectively. (Fig. 7(a) shows ground truth of 20 features.) The accuracy of Method 1 was far from that of simulation, but Method 2 had almost similar accuracy to simulation in such small number of steps. We can see usefulness of prior information in the early stages.

5.3 Results of BRM: Bearing-Range Mapping

ARTag makers give the local position of detected features. So, this case seems to be natural in this experiment (in BOM, we reduced information of measurements of range data). As we expected, this case outperformed BOM. Fig. 6(b) shows how MPEs change as the observation data increases in each method. MPEs at 35th step in each method showed almost same results, 5.1[cm] approximately. Fig. 7(d) illustrates a constructed map after 35 steps.

6 Conclusion

In this paper, we proposed an alternative approach to a mapping problem in two different cases, bearing-odometry mapping and bearing-range mapping. The key ideas are to estimate the inter-feature distance matrix from measurements to use multi-dimensional scaling for reconstructing the coordinates of the features from the distance matrix. Our proposed method M-SEIFD showed an achievement of mapping both in simulation and in real-world environment, especially in the second case, bearing-range mapping. It also has unique properties that prior information of inter-feature distances is efficiently utilized and that the robot self-localization is not necessary.

There are still many interesting issues related to this method, such as data association problem, extension to 3D mapping, and hybridization with the conventional SLAM methods.

References

1. Deans, M., Hebert, M.: Experimental comparison of techniques for localization and mapping using a bearing-only sensor. In: International Conference on Experimental Robotics (2000)

2. Davison, A.J.: Real-time simultaneous localisation and mapping with a single camera. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, pp. 1403–1410 (2003)
3. Bekris, K.E., Glick, M., Kavraki, L.E.: Evaluation of algorithms for bearing-only SLAM. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation, pp. 1937–1943 (2006)
4. Yairi, T., Kanazaki, H.: Bearing-only mapping by sequential triangulation and multi-dimensional scaling. In: Proceedings of the 2008 IEEE International Conference on Robotics and Automation (2008)
5. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics. MIT Press, Cambridge (2005)
6. Leonard, J., Feder, H.: A computationally efficient method for large-scale concurrent mapping and localization. In: Proceedings of International Symposium on Robotics and Research (1999)
7. Thrun, S., Fox, D., Burgard, W.: A probabilistic approach to concurrent mapping and localization for mobile robots. Machine Learning 31, 29–53 (1998)
8. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: Proceedings of AAAI 2002, pp. 593–598 (2002)
9. Dellaert, F., Kaess, M.: Square root SAM: Simultaneous localization and mapping via square root information smoothing. International Journal of Robotics Research 25(12), 1181–1203 (2006)
10. Lu, F., Milios, E.: Globally consistent range scan alignment for environment mapping. Autonomous Robots 4, 333–349 (1997)
11. Duckett, T., Marsland, S., Shapiro, J.: Learning globally consistent maps by relaxation. In: Proceedings of the IEEE International Conference on Robotics and Automation (2000)
12. Yairi, T.: Covisibility-based map learning method for mobile robots. In: Proceedings of Pacific Rim International Conference on Artificial Intelligence, pp. 703–712 (2004)
13. Yairi, T.: Map building without localization by dimensionality reduction techniques. In: Proceedings of the 24th International Conference on Machine Learning, pp. 1071–1078 (2007)
14. Cox, T., Cox, M.: Multidimensional Scaling. Chapman & Hall/Crc, Boca Raton
15. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
16. DeLeeuw, J.: Applications of convex analysis to multidimensional scaling. Recent developments in statistics, 133–145 (1977)
17. Fiala, M.: ARTag, a fiducial marker system using digital techniques. In: Proceedings of the IEEE International Computer Society Conference on Computer Vision and Pattern Recognition, pp. 590–596 (2005)

Document-Based HITS Model for Multi-document Summarization

Xiaojun Wan

Institute of Computer Science and Technology,
Peking University, Beijing 100871, China
wanxiaojun@icst.pku.edu.cn

Abstract. The PageRank model has been successfully exploited for multi-document summarization by making use of the link relationships between sentences in the document set, under the assumption that all the sentences are indistinguishable from each other. However, different documents in the set are usually not equally important, and the sentences in an important document are deemed more salient than the sentences in a trivial document. This paper proposes the document-based HITS model (DocHITS) to fully leverage the document-level information by considering documents and sentences as hubs and authorities. Experimental results on the DUC2001 and DUC2002 datasets demonstrate the good effectiveness of our proposed model.

1 Introduction

Generic multi-document summarization aims to produce a summary delivering the majority of information content from a set of documents, without any prior knowledge. Automated multi-document summarization has drawn much attention in recent years. Multi-document summary is usually used to provide concise topic description about a cluster of documents and facilitate the users to browse the document cluster. For example, a number of news services, such as GoogleNews¹, NewsInEssence², have been developed to group news articles into news topics, and then produce a short summary for each news topic. The users can easily understand the topic they have interest in by taking a look at the short summary.

A particular challenge for generic multi-document summarization is how to extract and merge the globally important information from different documents. The documents might contain much information unrelated to the main topic. Hence we need effective summarization methods to analyze the information stored in different documents and extract the important information related to the main topic. In other words, a good summary is expected to preserve the globally important information contained in the documents as much as possible, and at the same time keep the information as novel as possible.

In recent years, multi-document summarization has been widely explored in the natural language processing and information retrieval communities. A series of work-

¹ <http://news.google.com>

² <http://lada.si.umich.edu:8080/clair/nie1/nie.cgi>

shops and conferences on automatic text summarization (e.g. DUC³), special topic sessions in ACL, COLING, and SIGIR have advanced the technology and produced a couple of experimental online systems. Generally speaking, the methods can be abstractive summarization or extractive summarization. Extractive summarization is a simple but robust method for text summarization and it involves assigning saliency scores to some units (e.g. sentences, paragraphs) of the documents and extracting those with highest scores, while abstraction summarization usually needs information fusion, sentence compression and reformulation. In this study, we focus on extractive summarization.

Most recently, the PageRank Model has been successfully applied for multi-document summarization by making use of the “voting” or “recommendations” between sentences in the documents [3, 14, 17]. The model first constructs a directed or undirected graph to reflect the relationships between the sentences and then applies the PageRank algorithm [15] to compute the rank scores for the sentences. The sentences with large rank scores are chosen into the summary. However, the model makes uniform use of the sentences in different documents, i.e. all the sentences are ranked without considering the document-level information. Actually, given a document set, different documents are not equally important. For example, the documents close to the main topic of the document set are usually more important than the documents far away from the main topic of the document set. This document-level information is deemed to have great influence on the sentence ranking process. In brief, the document-level information cannot be taken into account in the existing PageRank Model.

In order to address the limitations of the PageRank Model, we propose the document-based HITS Model to incorporate the document-level information in this study. The documents and sentences are considered as hubs and authorities, respectively, and then the HITS algorithm [7] is applied on the document-to-sentence bipartite graph to compute the saliency scores of the sentences. Experiments on the DUC2001 and DUC2002 datasets have been performed and the results demonstrate the good effectiveness of the proposed model. Our proposed model can significantly outperform the baseline PageRank model and the baseline sentence-based HITS model over all three ROUGE metrics. The parameters in the proposed model have also been investigated through experiments and the results show the robustness of the proposed model.

The rest of this paper is organized as follows: Section 2 briefly introduces the related work. The basic PageRank model is introduced in Section 3 and the document-based HITS model is proposed in Section 4. We describe the experiments and results in Sections 5 and 6, respectively. Lastly we conclude this paper in Section 7.

2 Related Work

A variety of extractive multi-document summarization methods have been developed recently. The centroid-based method [16] is one of the most popular extractive summarization methods. MEAD⁴ is an implementation of the centroid-based method that

³ <http://duc.nist.gov>

⁴ <http://www.summarization.com/mead/>

scores sentences based on sentence-level and inter-sentence features, including cluster centroids, position, TFIDF, etc. NeATS [10] uses sentence position, term frequency, topic signature and term clustering to select important content, and use MMR [4] to remove redundancy. To further explore user interface issues, iNeATS [9] is developed based on NeATS. XDoX [6] is a cross document summarizer designed specifically to summarize large document sets by identifying the most salient themes within the set by passage clustering and then composes an extraction summary, which reflects these main themes. The passages are clustered based on n-gram matching. Much other work also explores to find topic themes in the documents for summarization, e.g. Harabagiu and Lacatusu [5] investigate five different topic representations and introduce a novel representation of topics based on topic themes. In addition, Marcu [13] selects important sentences based on the discourse structure of the text. TNO’s system [8] scores sentences by combining a unigram language model approach with a Bayesian classifier based on surface features.

Most recently, the graph-based ranking methods have been proposed to rank sentences or passages based on the “votes” or “recommendations” between each other. Websumm [12] uses a graph-connectivity model and operates under the assumption that nodes which are connected to many other nodes are likely to carry salient information. LexPageRank [3] is an approach for computing sentence importance based on the concept of eigenvector centrality. It constructs a sentence connectivity matrix and computes sentence importance based on an algorithm similar to PageRank [15]. Mihalcea and Tarau [14] also propose a similar algorithm based on PageRank to compute sentence importance for single document summarization, and for multi-document summarization, they use a meta-summarization process to summarize the meta-document produced by assembling all the single summary of each document. Wan and Yang [17] improve the graph-ranking algorithm by differentiating intra-document links and inter-document links between sentences. All these methods make use of the relationships between sentences and select sentences according to the “votes” or “recommendations” from their neighboring sentences.

Other related work includes topic-focused document summarization [2], which aims to produce summary biased to a given topic or query.

3 The PageRank Model

The basic idea of the PageRank model is that of “voting” or “recommendation” between the sentences. A link between two sentences is considered as a vote cast from one sentence to the other sentence. The score of a sentence is determined by the votes that are cast for it, and the scores of the sentences casting these votes.

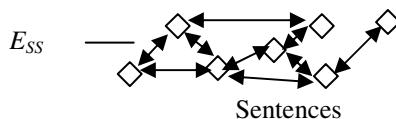


Fig. 1. One-layer link graph

Formally, given a document set D , let $G=(S, E_{ss})$ be an undirected graph to reflect the relationships between sentences in the document set, as shown in Figure 1. $S=\{s_i \mid 1 \leq i \leq n\}$ is the set of vertices and each vertex s_i in S is a sentence in the document set. $E_{ss}=\{e_{ij} \mid s_i, s_j \in S \text{ and } i \neq j\}$ is the set of edges. Each edge e_{ij} in E_{ss} is associated with an affinity weight aff_{ij} between sentences s_i and s_j . The weight is computed by using the standard cosine measure [1] between the two sentences as follows.

$$aff_{ij} = sim_{cosine}(s_i, s_j) = \frac{\vec{s}_i \cdot \vec{s}_j}{|\vec{s}_i| \times |\vec{s}_j|} \quad (1)$$

where \vec{s}_i and \vec{s}_j are the corresponding term vectors for sentences s_i and s_j , respectively. The term weight in the vector is set to the TFIDF value of the term in the sentence. Here, we have $aff_{ij} = aff_{ji}$ because G is an undirected graph.

We use an affinity matrix $M = (M_{ij})_{|S| \times |S|}$ to describe G with each entry corresponding to the weight of an edge in the graph, i.e., $M_{ij} = aff_{ij}$ if $i \neq j$, and otherwise $M_{ii} = 0$. Then M is normalized to \tilde{M} to make the sum of each row equal to 1. The saliency score $SenScore(s_i)$ for sentence s_i can be deduced from those of all other sentences linked with it and it can be formulated in a recursive form as in the PageRank algorithm:

$$SenScore(s_i) = \mu \cdot \sum_{all \ j \neq i} SenScore(s_j) \cdot \tilde{M}_{ji} + \frac{(1-\mu)}{|S|} \quad (2)$$

And the matrix form is:

$$\vec{\lambda} = \mu \tilde{M}^T \vec{\lambda} + \frac{(1-\mu)}{|S|} \vec{e} \quad (3)$$

where $\vec{\lambda} = [SenScore(s_i)]_{|S| \times 1}$ is the vector of sentence saliency scores. \vec{e} is a vector with all elements equaling to 1. μ is the damping factor usually set to 0.85, as in the PageRank algorithm.

The above process can be considered as a Markov chain by taking the sentences as the states and the corresponding transition matrix is given by $\mu \tilde{M}^T + \frac{(1-\mu)}{|S|} \vec{e} \vec{e}^T$.

The stationary probability distribution of each state is obtained by the principal eigenvector of the transition matrix. For numerical computation of the scores, the initial scores of all sentences are set to 1 and Equation (2) is used to compute the new scores until convergence.

We can see that the PageRank Model is built on the single-layer sentence graph and the transition probability between two sentences in the Markov chain depends only on the sentences themselves, not taking into account the document-level information.

4 The Proposed Document-Based HITS Model

Different from the PageRank Model, the HITS model distinguishes the hubs and authorities in the objects. A hub object has links to many good authorities, and an authority object has high quality content and there are many hubs linking to it. The hub scores and authority scores are computed in a reinforcement way.

In this study, we consider the documents as hubs and the sentences as authorities. Figure 2 gives the bipartite graph representation, where the upper layer is the hubs and the lower layer is the authorities. The HITS model can naturally take into account the document-level information by making use of the sentence-to-document relationships.

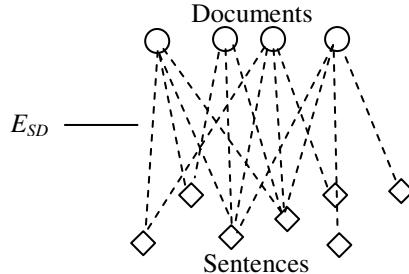


Fig. 2. Bipartite link graph

Formally, the representation for the bipartite graph is denoted as $G^* = \langle S, D, E_{SD} \rangle$, where $S = \{s_i \mid 1 \leq i \leq n\}$ is the set of sentences (i.e. authorities) and $D = \{d_j \mid 1 \leq j \leq m\}$ is the set of documents (i.e. hubs); $E_{SD} = \{e_{ij} \mid s_i \in S, d_j \in D\}$ corresponds to the correlations between any sentence and any document. Each edge e_{ij} is associated with a weight w_{ij} denoting the strength of the relationship between sentence s_i and document d_j . The weight w_{ij} is computed by using the standard cosine measure. We let $L = (L_{i,j})_{|S| \times |D|}$ denote the association matrix and L is defined as follows.

$$L_{i,j} = w_{ij} = \text{sim}_{\text{cosine}}(s_i, d_j) \quad (4)$$

Then the authority score $\text{AuthScore}^{(t+1)}(s_i)$ of sentence s_i and the hub score $\text{HubScore}^{(t+1)}(d_j)$ of document d_j at the $(t+1)^{\text{th}}$ iteration are computed based on the hub scores and authority scores at the t^{th} iteration as follows:

$$\text{AuthScore}^{(t+1)}(s_i) = \sum_{d_j \in D} w_{ij} \cdot \text{HubScore}^{(t)}(d_j) \quad (5)$$

$$\text{HubScore}^{(t+1)}(d_j) = \sum_{s_i \in S} w_{ij} \cdot \text{AuthScore}^{(t)}(s_i) \quad (6)$$

And the matrix form is as follows:

$$A^{(t+1)} = L H^{(t)} \quad (7)$$

$$H^{(t+1)} = L^T A^{(t)} \quad (8)$$

where $A^{(t)} = [\text{AuthScore}(s_i)]_{|S| \times 1}$ is the vector of authority scores for the sentences at the t^{th} iteration and $H^{(t)} = [\text{HubScore}(d_j)]_{|D| \times 1}$ is the vector of hub scores for the

documents at the t^{th} iteration. In order to guarantee the convergence of the iterative form, A and H are normalized after each iteration as follows:

$$A^{(t+1)} = A^{(t+1)} / \|A^{(t+1)}\| \quad (9)$$

$$H^{(t+1)} = H^{(t+1)} / \|H^{(t+1)}\| \quad (10)$$

It can be proved that the authority vector A converges to the dominant eigenvector of the authority matrix LL^T , and the hub vector H converges to the dominant eigenvector of the hub matrix L^TL . For numerical computation of the scores, the initial authority scores of all sentences and the initial hub scores of all documents are set to 1 and the above iterative steps are used to compute the new scores until convergence. Usually the convergence of the iteration algorithm is achieved when the difference between the scores computed at two successive iterations for any sentences and documents falls below a given threshold (0.0001 in this study).

Finally, we use the authority scores as the saliency scores for the sentences, i.e., we have $\text{SenScore}(s_i) = \text{AuthScore}(s_i)$.

The above document-based HITS model can take into account the document-level information by making use of the mutual influences between documents and sentences. The importance of a document is decided by the saliency of the sentences that relate with the document, and the saliency of a sentence is decided by the importance of the documents that relate with the sentence.

5 Evaluation Setup

5.1 Summary Extraction

Note that after the saliency scores of sentences have been obtained using the above PageRank model or the HITS model, a variant version of the MMR algorithm [4] is used to remove redundancy and choose both informative and novel sentences into the summary. The basic idea of the algorithm is to decrease the final rank score of less informative sentences by the part conveyed from the most informative one. The algorithm goes in a greedy way as follows [17]:

1. Initialize two sets $A=\emptyset$, $B=\{s_i \mid i=1, 2, \dots, n\}$, and each sentence's rank score is initialized to its saliency score computed by the above PageRank or HITS model, i.e. $\text{RankScore}(s_i) = \text{SenScore}(s_i)$, $i=1, 2, \dots, n$.
2. Sort the sentences in B by their current rank scores in descending order.
3. Suppose s_i is the highest ranked sentence, i.e. the first sentence in the ranked list. Move the sentence s_i from B to A , and then the diversity penalty is imposed on the rank score of each sentence linked with s_i as follows:

For each sentence s_j in B , $j \neq i$

$$\text{RankScore}(s_j) = \text{RankScore}(s_j) - \omega \cdot \tilde{M}_{ji} \cdot \text{SenScore}(s_i) \quad (11)$$

where \tilde{M} is the normalized affinity matrix as defined in Section 3, reflecting the similarity relationships between all sentences. $\omega > 0$ is the penalty degree factor.

The larger ω is, the greater penalty is imposed on the rank score. If $\omega=0$, no diversity penalty is imposed at all. ω is typically set to 10 in the experiments.

4. Go to step 2 and iterate until $B = \emptyset$ or the iteration count reaches a predefined maximum number.

After the final rank scores are obtained for all the sentences, a few sentences with highest final rank scores are chosen to produce the summary according to the summary length limit.

5.2 Datasets and Metrics

Generic multi-document summarization has been one of the fundamental tasks in DUC 2001⁵ and DUC 2002⁶ (i.e. task 2 in DUC 2001 and task 2 in DUC 2002), and we used the two tasks for evaluation. DUC2001 provided 30 document sets and DUC 2002 provided 59 document sets (D088 is excluded from the original 60 document sets by NIST) and generic abstracts of each document set with lengths of approximately 100 words or less were required to be created. The documents were news articles collected from TREC-9. The sentences in each article have been separated and the sentence information has been stored into files. The summary of the datasets are shown in Table 1.

Table 1. Summary of datasets

	DUC 2001	DUC 2002
Task	Task 2	Task 2
Number of documents	309	567
Number of clusters	30	59
Data source	TREC-9	TREC-9
Summary length	100 words	100 words

We used the ROUGE [11] toolkit (i.e. ROUGEeval-1.4.2 in this study) for evaluation, which has been widely adopted by DUC for automatic summarization evaluation. It measured summary quality by counting overlapping units such as the n-gram, word sequences and word pairs between the candidate summary and the reference summary. ROUGE-N was an n-gram recall measure computed as follows:

$$ROUGE-N = \frac{\sum_{S \in \text{RefSum}} \sum_{n\text{-gram} \in S} Count_{match}(n\text{-gram})}{\sum_{S \in \text{RefSum}} \sum_{n\text{-gram} \in S} Count(n\text{-gram})} \quad (12)$$

where n stands for the length of the n-gram, and $Count_{match}(n\text{-gram})$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. $Count(n\text{-gram})$ is the number of n-grams in the reference summaries.

The ROUGE toolkit reported separate scores for 1, 2, 3 and 4-gram, and also for longest common subsequence co-occurrences. Among these different scores,

⁵ <http://www-nlpir.nist.gov/projects/duc/guidelines/2001.html>

⁶ <http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html>

unigram-based ROUGE score (ROUGE-1) has been shown to agree with human judgment most [11]. We showed three of the ROUGE metrics in the experimental results: ROUGE-1 (unigram-based), ROUGE-2 (bigram-based), and ROUGE-W (based on weighted longest common subsequence, weight=1.2). In order to truncate summaries longer than length limit, we used the “-l” option in ROUGE toolkit.

6 Evaluation Results

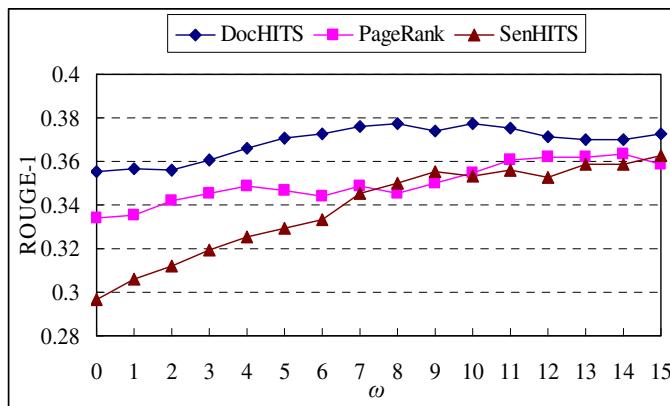
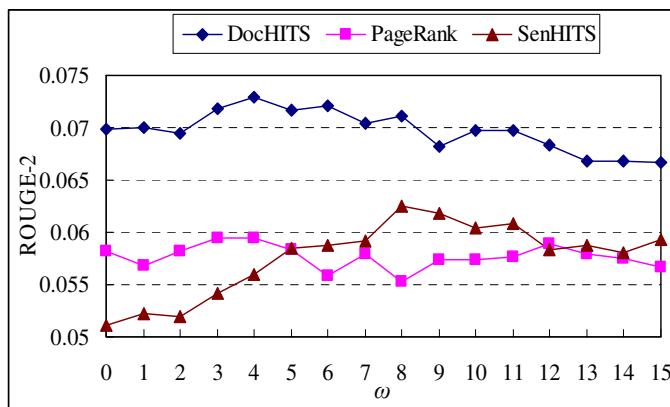
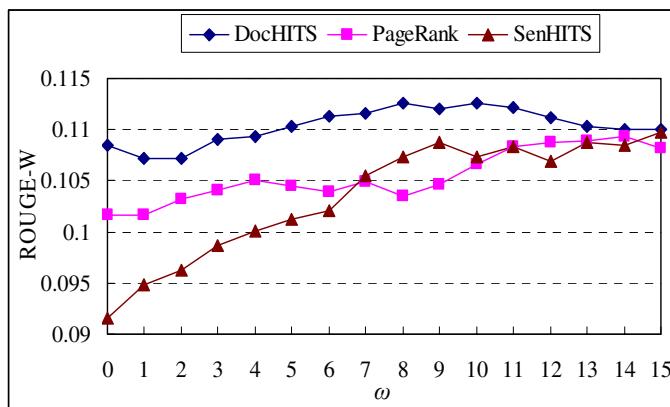
In the experiments, the proposed document-based HITS model (DocHITS) is compared with the following typical baseline systems: the PageRank model (PageRank), the sentence-based HITS model (SenHITS), the top three performing systems and two baseline systems defined by DUC. The SenHITS model does not take into account the document-level information. It considers sentences as both hubs and authorities by assigning a hub score and an authority score to each sentence, and then uses the reinforcement algorithm for the DocHITS model in Section 4 to iteratively compute the saliency scores of the sentences. The top three performing systems are the systems with highest ROUGE scores, chosen from the performing systems on each task respectively (i.e. SystemN, SystemP and SystemT in Table 2; System19, System26, System28 in Table 3). The lead baseline and coverage baseline are two baselines employed in the generic multi-document summarization tasks of DUC2001 and DUC2002. The lead baseline takes the first sentences one by one in the last document in the collection, where documents are assumed to be ordered chronologically. And the coverage baseline takes the first sentence one by one from the first document to the last document. Tables 2 and 3 show the comparison results on DUC2001 and DUC2002, respectively.

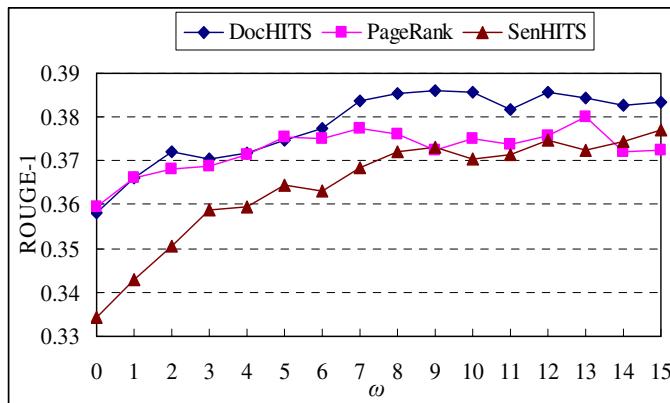
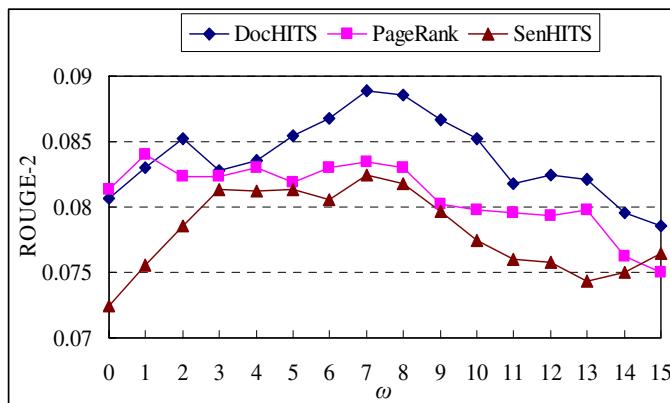
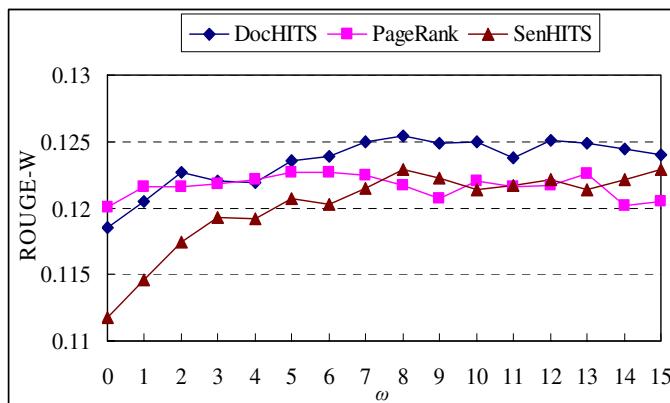
Table 2. Comparison results on DUC2001

System	ROUGE-1	ROUGE-2	ROUGE-W
DocHITS	0.37744	0.06966	0.11252
PageRank	0.35474	0.05733	0.10667
SenHITS	0.35320	0.06037	0.10737
SystemN	0.33910	0.06853	0.10240
SystemP	0.33332	0.06651	0.10068
SystemT	0.33029	0.07862	0.10215
Coverage	0.33130	0.06898	0.10182
Lead	0.29419	0.04033	0.08880

Table 3. Comparison results on DUC2002

System	ROUGE-1	ROUGE-2	ROUGE-W
DocHITS	0.38569	0.08519	0.12500
PageRank	0.37510	0.07973	0.12198
SenHITS	0.37057	0.07739	0.12132
System26	0.35151	0.07642	0.11448
System19	0.34504	0.07936	0.11332
System28	0.34355	0.07521	0.10956
Coverage	0.32894	0.07148	0.10847
Lead	0.28684	0.05283	0.09525

**Fig. 3.** ROUGE-1 vs. ω on DUC2001**Fig. 4.** ROUGE-2 vs. ω on DUC2001**Fig. 5.** ROUGE-W vs. ω on DUC2001

**Fig. 6.** ROUGE-1 vs. ω on DUC2002**Fig. 7.** ROUGE-2 vs. ω on DUC2002**Fig. 8.** ROUGE-W vs. ω on DUC2002

Seen from Tables 2 and 3, both the HITS models (DocHITS & SenHITS) and the PageRank model perform much better than the top performing systems and the coverage and lead baselines. Our proposed DocHITS model outperforms the PageRank model and the SenHITS model over all three metrics on both datasets⁷, which demonstrates that the document-level information does benefit the sentence ranking process and help to extract salient sentences. Our proposed document-based HITS model is validated to be very effective to leverage the document-level information in the summarization process.

In the above experiments, the penalty factor ω is typically set to 10 for all the DocHITS, PageRank and SenHITS models. In order to investigate how the factor influences the summarization performances, we vary the parameter ω from 0 to 15. Figures 3-5 show the ROUGE-1, ROUGE-2 and ROUGE-W performances with respect to ω on the DUC2001 dataset, respectively. And Figures 6-8 show the ROUGE-1, ROUGE-2 and ROUGE-W performances with respect to ω on the DUC2002 dataset, respectively. With the increase of ω , the greater penalty is imposed on the sentences.

Seen from the figures, our proposed DocHITS model can always perform better than the baseline PageRank and SenHITS models on both datasets, no matter how the penalty factor ω is set. The results demonstrate the robustness of our proposed document-based HITS model for multi-document summarization. The document-level information is further validated to be always beneficial to extract salient sentences from multiple documents.

7 Conclusion and Future Work

This paper proposes a novel document-based HITS model for multi-document summarization, which can naturally incorporate the document-level information in the sentence ranking process. The experimental results on DUC2001 and DUC2002 demonstrate the good effectiveness of the proposed model.

In future work, we will use more fine-grained model to exploiting the paragraph-level information to improve the summarization performance.

Acknowledgments. This work was supported by the National Science Foundation of China (No.60703064), the Research Fund for the Doctoral Program of Higher Education of China (No.20070001059) and the National High Technology Research and Development Program of China (No.2008AA01Z421).

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press and Addison Wesley (1999)
2. Daumé, H., Marcu, D.: Bayesian query-focused summarization. In: Proceedings of COLING-ACL 2006 (2006)
3. Erkan, G., Radev, D.: LexPageRank: prestige in multi-document text summarization. In: Proceedings of EMNLP 2004 (2004)

⁷ The ROUGE-1 improvements of DocHITS over PageRank and SenHITS are statistically significant.

4. Goldstein, J., Kantrowitz, M., Mittal, V., Carbonell, J.: Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In: Proceedings of ACM SIGIR 1999 (1999)
5. Harabagiu, S., Lacatusu, F.: Topic themes for multi-document summarization. In: Proceedings of SIGIR 2005 (2005)
6. Hardy, H., Shimizu, N., Strzalkowski, T., Ting, L., Wise, G.B., Zhang, X.: Cross-document summarization by concept classification. In: Proceedings of SIGIR 2002 (2002)
7. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46(5), 604–632 (1999)
8. Kraaij, W., Spitters, M., van der Heijden, M.: Combining a mixture language model and Naïve Bayes for multi-document summarization. In: SIGIR2001 Workshop on Text Summarization
9. Leuski, A., Lin, C.-Y., Hovy, E.: iNeATS: interactive multi-document summarization. In: Proceedings of ACL 2003 (2003)
10. Lin, C.-Y., Hovy, E.H.: From Single to Multi-document Summarization: A Prototype System and its Evaluation. In: Proceedings of ACL 2002 (2002)
11. Lin, C.-Y., Hovy, E.H.: Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In: Proceedings of HLT-NAACL 2003 (2003)
12. Mani, I., Bloedorn, E.: Summarizing Similarities and Differences Among Related Documents. *Information Retrieval* 1(1) (2000)
13. Marcu, D.: Discourse-based summarization in DUC–2001. In: SIGIR 2001 Workshop on Text Summarization (2001)
14. Mihalcea, R., Tarau, P.: A language independent algorithm for single and multiple document summarization. In: Proceedings of IJCNLP 2005 (2005)
15. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Libraries (1998)
16. Radev, D.R., Jing, H.Y., Stys, M., Tam, D.: Centroid-based summarization of multiple documents. *Information Processing and Management* 40, 919–938 (2004)
17. Wan, X., Yang, J.: Improved affinity graph based multi-document summarization. In: Proceedings of HLT-NAACL 2006 (2006)

External Force for Active Contours: Gradient Vector Convolution

Yuanquan Wang^{1,2} and Yunde Jia²

¹ Tianjin Key Laboratory of Intelligent Computing and Novel Software Technology,

School of Computer Science, Tianjin University of Technology, Tianjin 300191, PRC

² Beijing Laboratory of Intelligent Information Technology, School of Computer Science,
Beijing Institute of Technology, Beijing 100081, PRC

Abstract. Active contours, or Snakes, have been widely used in image processing and computer vision and intensively studied over the last two decades. The philosophy of these models involves designing the internal and external forces and the external force drives the contours to locate objects in images. This paper presents a novel external force called gradient vector convolution (GVC) for active contours. The proposed method is motivated by gradient vector flow (GVF) and possesses some advantages of GVF, such as enlarged capture range, initialization insensitivity and high performance on concavity convergence; in addition, it can be implemented in real time owing to its convolution mechanism. Some experiments are presented to demonstrate the effectiveness of the proposed method.

Keywords: active contour, gradient vector flow (GVF), gradient vector convolution, image segmentation.

1 Introduction

Image segmentation is one of the fundamental problems in image processing and computer vision, its main goal is to divide an image into several meaningful regions such that each region is homogeneous with respect to a certain criterion. Exploring accurate and efficient segmentation algorithms remains open because of image inhomogeneity and variability of object shapes. During the last twenty years, variational methods have been broadly studied for image segmentation and promising results are obtained [1~3]. The advantages of variational methods lie in their flexibility in modeling and various existing numerical implementation. The rationale of variational methods is to minimize an image-dependent energy functional; therefore, image segmentation is transferred to a mathematical optimization problem.

Active contour models, or snakes, have been one of the most successful variational models for image segmentation [1,3,4], and they are elastic curves with internal and external energies; under these energies, the curves move and change their shapes to seek their local minimum energy. The internal energy keeps the contour continuous and smooth during deformation, while the external force drives the contour toward an object boundary or other desired features within an image. By constraining extracted

boundaries to be smooth and by incorporating other prior information about the object shape, active contours are robust to both image noise and boundary gaps. According to the representation of active contours, there are two kinds of active contours: parametric active contour (PAC) which adopts an explicit representation and geometric active contour (GAC) resorting to implicit manner. GAC is superior to the PAC in terms of topological changing issue; when there is no topology change, PAC still has some advantages such as noise robustness and low computation load and one can get directly meaningful description of objects from PAC.

Since the external energy, or external force, plays a leading role in driving the snake contour to approach objects, it is widely studied in literatures. For example, Xu and Prince [5] proposed the GVF external force which outperforms the other gradient-based methods in capture range enlarging and concavities convergence and becomes the focus of many research [6]. Park [7] and Yuan [8] almost simultaneously proposed the virtual electric field (VEF) external force, in which each pixel is considered as a static charge. The VEF possesses some properties similar to the GVF, but has much shorter computational time than the GVF. Very recently, Li and Acton [9] proposed an extended version of the VEF model by modifying the distance metric in VEF. Sum and Cheung [10] proposed the boundary vector field external force, under this external force framework, the snake contour evolves in two phases. Jalba *et al.* [11] recently proposed the charged particle model (CPM), where each pixel is also considered as a static charge. Later, the concepts proposed in [11] were generalized to geometric models [12].

In this paper, a convolution-based external force called gradient vector convolution (GVC) is proposed. The GVC method is motivated by gradient vector flow (GVF) and possesses some advantages of the GVF such as enlarged capture range, initialization insensitivity, concavity convergence, but its computational cost is low owing to its convolution mechanism. Some experiments are presented to demonstrate these advantages.

The remainder of this paper is organized as follows: the snake model and GVF external force are briefly reviewed in Section 2; in Section 3, we detail the proposed method, including the philosophies and experimental results and Section 4 concludes this paper.

2 Brief Review of the Snake Model and GVF External Force

A snake contour is an elastic curve that moves and changes its shape to minimize the following energy [1]:

$$E_{\text{snake}} = \int \frac{1}{2} (\alpha |\mathbf{c}_s|^2 + \beta |\mathbf{c}_{ss}|^2) + E_{\text{ext}}(\mathbf{c}(s)) ds . \quad (1)$$

where $\mathbf{c}(s) = [x(s) \ y(s)]$, $s \in [0,1]$, is the snake contour parameterized by arc length, $\mathbf{c}_s(s)$ and $\mathbf{c}_{ss}(s)$ are the first and second derivative of $\mathbf{c}(s)$ with respect to s and positively weighted by α and β respectively. $E_{\text{ext}}(\mathbf{c}(s))$ is the image potential which may result from various events, e.g., lines and edges. By calculus of variation, the Euler equation to minimize E_{snake} is

$$\alpha \mathbf{c}_{ss}(s) - \beta \mathbf{c}_{ssss}(s) - \nabla E_{ext} = 0 . \quad (2)$$

This can be considered as a force balance equation as follows:

$$\mathbf{F}_{int} + \mathbf{F}_{ext} = 0 . \quad (3)$$

where $\mathbf{F}_{int} = \alpha \mathbf{c}_{ss}(s) - \beta \mathbf{c}_{ssss}(s)$ and $\mathbf{F}_{ext} = -\nabla E_{ext}$. The internal force \mathbf{F}_{int} makes the snake contour to be smooth while the external force \mathbf{F}_{ext} attracts the snake to the desired image features.

In a departure from this perspective, the gradient vector flow external force is introduced to replace $-\nabla E_{ext}$ with a new vector $\mathbf{v}(x,y) = [u(x,y), v(x,y)]$ which is derived by minimizing the following function [5]:

$$\mathcal{E} = \iint \mu |\nabla \mathbf{v}|^2 + |\nabla f|^2 |\mathbf{v} - \nabla f|^2 dx dy . \quad (4)$$

where f is the edge map of image I , usually, $f = |\nabla G_\sigma * I|$ μ is a positive weight. Using calculus of variation, the Euler equations seeking the minimum of \mathcal{E} are

$$\begin{aligned} u_t &= \mu \nabla^2 u - (f_x^2 + f_y^2)(u - f_x) \\ v_t &= \mu \nabla^2 v - (f_x^2 + f_y^2)(v - f_y) \end{aligned} . \quad (5)$$

where ∇^2 is the Laplacian operator. The snake model with \mathbf{v} as external force is called GVF snake.

3 GVC: Gradient Vector Convolution External Force

3.1 GVC: Rationale and Experiments

The external force of the original snake model is the gradient of image edge maps, therefore, its magnitude is large only in the immediate vicinity of the edges and dies out rapidly when moving away from the image boundaries, even zero in homogeneous regions [5], this implies the capture range of the original snake is very small, so that the initial contour should be close to the desired boundaries and the snake contour even cannot dive into deep concavities. Also, image noise can cause spurious edges and the contour can easily be trapped by these false edges. To address these difficulties, Xu and Prince [5] proposed the GVF external force by diffusing the gradient vector; as a result, the original gradient vector is smoothed and extended far from image edges.

Although the performance of GVF is really high at capture range enlarging and concavities convergence, but it is computationally expensive in that one has to iteratively solve the generalized diffusion Eqs.(5) on the whole image. In order to solve this problem, we argue that one can extend the gradient vector and suppress noise by convolving the gradient vector with a certain kernel. Thanks to fast Fourier transform, this convolution operation would be implemented in real time and the snake model would benefit much from this convolution operation in computation time. We refer to

this convolution based external force as *gradient vector convolution*, in short, GVC. Denote the convolution kernel by $K(x,y)$, GVC takes the following form:

$$\begin{cases} u = f_x \otimes K(x,y) \\ v = f_y \otimes K(x,y) \end{cases} \quad (6)$$

where $[f_x, f_y]$ is the gradient vector of image edge map f . In practice, we take $K(x, y) = 1/r_h^n$, $h, n \in R^+$, $r_h = \sqrt{x^2 + y^2 + h}$, which always works well in terms of extending and smoothing gradient vector. Generally, large ' n ' makes the potential to decay fast with distance and vice versa; this property allows the GVC snake adapting to different applications. The factor ' h ' plays a role analogous to scale space filtering, the greater the value of h , the greater the smoothing effect on the results; this property suggests that GVC would be very robust to noise. In addition, the kernel size also affects the final GVC, the larger the size, the greater the smoothing effect on final GVC.

We will conduct several experiments to demonstrate the properties of GVC. These experiments presented here are implemented in MATLAB 6.5 and run on a 1.83GHz CPU with 512M RAM. We first calculate the GVC on the room and u-shape images used in [5] to demonstrate concavity convergence and capture range enlarging. The parameters for GVC are: $h = 0$, $n = 2.0$, the kernel size is the same as that of the image. Size of both images is 64×64 . Fig.1 shows the results. It can be seen from these results that GVC works very well and similar to GVF (see [5]). But the execution time of GVC for both images is $0.046s$ while that of GVF is $0.235s$ with 50 iterations.

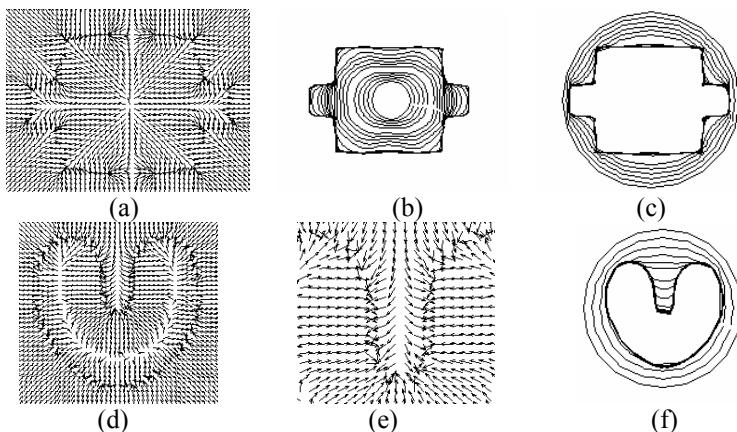


Fig. 1. GVC External force and convergence of snakes. (a)~(c) On room image; (d)~(f) On u-shape image.

As a second example, a comparative study of GVC and GVF is conducted on the heart image used in [5], shown in Fig.2(a). The heart image is first blurred with a Gaussian kernel with standard deviation 3.0, and the edge map is shown in Fig.2(b).

The GVC is calculated with $n=2.0$, $h=2.0$, and execution time is $0.531s$, as shown in Fig.2(c). A circle is used as initial contour, the snake contour can correctly locate the left ventricle and the result is in Fig.2(d). The GVF is also calculated, for all GVF in this experiment, $\mu=0.15$ and time step is 0.5. Fig.2(e) presents the GVF after 200 iterations, and execution time is $6.407s$. The same circle as in Fig.2(d) is used as initialization, but due to the critical points in GVF [13], the snake contour collapses to a tangled curve, see Fig.2 (f), therefore, a larger circle including the critical points is employed as initialization and the segmentation result is fairly satisfactory, see Fig.2(g). This result implies that the critical point issue of GVC snake is less serious than that of the GVF snake. In addition, comparing Fig.2 (d) and (g), although the initial contour in Fig.2 (g) is larger, but the segmentation result of the GVC snake is much better than that of the GVF snake at the 12- and 1-o'clock position. We also observe this phenomenon in the room image that GVC snake can detect the corner more accurately than that of the GVF snake (see the result in [5] and in Fig.1). What's more, let's keep in mind: *the computation time of the GVC is much shorter than that of the GVF*.

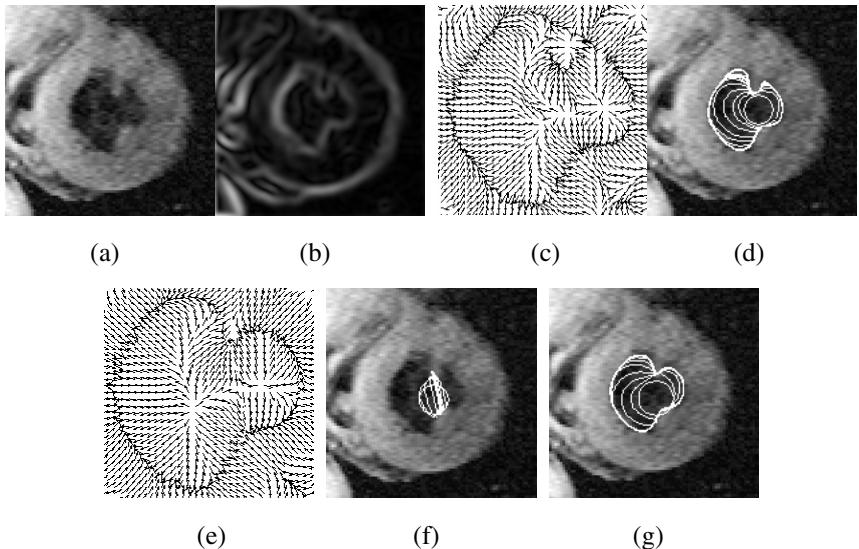


Fig. 2. Comparison of GVC and GVF on heart image. See text for details.

3.2 Remarks on GVC

It is well-known that for any bounded $g \in C(R^2)$, the linear diffusion process

$$\begin{cases} u_t = \Delta u \\ u(x,0) = g(x) \end{cases}. \quad (7)$$

possesses the unique solution

$$u(x, t) = \begin{cases} g(x) & t = 0 \\ (G_{\sqrt{2t}} \otimes g)(x) & t > 0 \end{cases}. \quad (8)$$

where G_σ is the Gaussian kernel with standard deviation σ . This fact tells us that the solution of diffusion equation can be obtained by convolution operation. Meanwhile, just as stated by Xu and Prince [5], the GVF equation (Eqs.(5)) is a biased version of Eqs.(7), can the solution of Eqs.(5) also be obtained via convolution? Unfortunately, we can not explore the exact relationship between the solution of Eqs.(5) and a certain convolution, but this motivates us to convolve the gradient vector with a certain kernel to extend the gradient vector and suppress noise so as to facilitate initialization of snake model. In some sense, the GVC can be considered as an approximation of GVF; it is very interesting that the GVC, an approximation, outperforms the GVF, the original one, particularly in computational time. The GVC external force can serve as an alternative to GVF, not only for active contours, but also for other applications such as finding symmetry axes [14] and extraction of curve skeletons [15].

4 Conclusion

In this paper, we have presented a novel external force called gradient vector convolution (GVC) for active contours. The GVC is obtained by convolving the gradient vector with a certain kernel and a practical form of the kernel is also presented. We have also showed qualitatively that the GVC is an approximation of the GVF. Experiments demonstrate that the GVC snake outperforms the GVF snake, in particular in computational time. For further study, we will launch an in-depth theoretical analysis of the GVC and explore the performance of the GVC for more applications.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under grants 60602050.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snake: active contour models. *Int'l J. Computer Vision* 1(4), 321–331 (1988)
2. Tsai, A., Yezzi, J.A., Willsky, A.S.: Curve Evolution Implementation of the Mumford–Shah Functional for Image Segmentation, Denoising, Interpolation, and Magnification. *IEEE TIP* 10(8), 1169–1186 (2001)
3. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. *Int'l J. Computer Vision* 22, 61–79 (1997)
4. Xie, X., Mirmehdi, M.: MAC: Magnetostatic Active Contour Model. *IEEE TPAMI* 30(4), 632–646 (2008)
5. Xu, C., Prince, J.L.: Snakes, Shapes and gradient vector flow. *IEEE TIP* 17(3), 359–369 (1998)
6. Han, X., Xu, C., Prince, J.L.: Fast numerical scheme for gradient vector flow computation using a multigrid method. *IET IP* 1(1), 48–55 (2007)

7. Park, H.K., Chung, M.J.: External force of snake: virtual electric field. *Electronics Letters* 38(24), 1500–1502 (2002)
8. Yuan, D., Lu, S.: Lu, Siwei: Simulated static electric field (SSEF) snake for deformable models. In: The 16 th International Conference on Pattern Recognition, pp. 83–86. IEEE press, New York (2002)
9. Bing, L., Acton, S.T.: Active Contour External Force Using Vector Field Convolution for Image Segmentation. *IEEE TIP* 16(8), 2096–2106 (2007)
10. Sum, K.W., Cheung, Y.S.: Boundary vector field for parametric active contours. *Pattern Recognition* 40, 1635–1645 (2007)
11. Jalba, A.C., Wilkinson, H.F., Roerdink, J.: BTM: CPM: A deformable model for shape recovery and segmentation based on charged particles. *IEEE TPAMI* 26(10), 1320–1335 (2004)
12. Yang, R., Mirmehdi, M.: A Charged Geometric Model for Active Contours. In: The 18th International Conference on Pattern Recognition, pp. 183–186. IEEE press, New York (2006)
13. Wang, Y., Liang, J., Jia, Y.: On the critical point of gradient vector flow snake model. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part II. LNCS, vol. 4844, pp. 754–763. Springer, Heidelberg (2007)
14. Prasad, V.S.N., Yegnanarayana, B.: Finding Axes of symmetry from potential fields. *IEEE trans IP* 13(12), 1559–1566 (2004)
15. Hassouna, M., Farag, A.A.: On the Extraction of Curve Skeletons using Gradient Vector Flow. In: IEEE 11th International Conference on Computer Vision, pp. 1–8. IEEE Press, New York (2007)

Representation = Grounded Information

Mary-Anne Williams

Innovation and Enterprise Research Laboratory
University of Technology, Sydney
NSW 2007, Australia
Mary-Anne@it.uts.edu.au

Abstract. The grounding problem remains one of the most fundamental issues in the field of Artificial Intelligence. We argue that representations are grounded information and that an intelligent system should be able to make and manage its own representations. A perusal of the literature reveals much confusion and little progress in understanding the grounding problem. In this paper we delineate between information and representation where a representation is grounded information; as a result we provide much needed clarity and a new base from which to conduct an innovative analysis of grounding that delivers a novel and insightful understanding of intelligence that can be used to guide and inform the design and construction of robust autonomous intelligent systems with intention and that know what they are doing and why they are doing it.

Keywords: Robotics, Perception, Knowledge Representation, Intelligent Systems.

1 Introduction

Knowledge representation is a key pillar of knowledge management, artificial intelligence, cognitive science and software engineering. The field of Artificial Intelligence (AI) explores ways to build representations of information from the world's richness and to manage these representations over time for a range of purposes from decision making to actuation. Like humans, a truly intelligent system should be able to build its own representations. This paper argues that the field of AI should devote more effort and resources to the development of a practical understanding of how to design systems that can build their own representations, possess intention and know what they are doing and why.

Despite its fundamental importance there are few formal frameworks for exploring and evaluating knowledge representations. Even worse there is no common agreement on what a representation actually is or should be, not even an informal description of a representation. Terms like information, representation, and models are often used interchangeably which only fuels the confusion. In this paper we distinguish information from representation, and classify models as either informational or representational.

In the early 1990s Brooks challenged the prevailing assumption that representations are needed for intelligence. His work generated much lively debate and argument. His so-called behavior-based architecture is founded on the idea that a system

need not build and maintain models of the world because the world is its own best model and an intelligent system can simply sense the world directly, instead of investing in building and maintaining a so called world model [2, 3]. Despite giving the status quo a shakeup and putting robot behavior and experience alongside rationality on the research agenda, Brooks' approach has not lead to major advances or scientific breakthroughs. Furthermore, whilst his robot designs were not based on logical representations they were not devoid of symbolic representation e.g. they used entities in computer memory such as numbers.

In this paper we put forward the proposal that representations are grounded information, and that human intelligence stems from our astounding ability to build representations, i.e. to ground information. In fact, we ground most information so effortlessly that we take the process and the capability to do so for granted, and as a result we often conflate information, models and representations, and even worse our *perception of reality* with *reality* itself. When we try to build artificially intelligent systems like robots and design their grounding capabilities we come up against the complexity and challenge that the grounding problem presents. This paper takes an important step towards clarifying the underlying concepts and terminology that we can use to advance our understanding of the grounding problem.

AI researchers have designed and built impressive robotic systems that demonstrate complex skills and that can perform sophisticated tasks; however these systems are limited in their capacity for autonomous representation construction, because the robot's mechanisms to ground information are carefully crafted by system designers and the robots operate with significant human assistance and intervention over their whole lifetime. Few robots can build representations on the fly by themselves. As a result most systems perform poorly in novel or unexpected situations because they cannot create effective representations of their new experiences.

The idea that representations are grounded information provides a crucial distinction between the basic concepts of *representation* and *information*. We believe that the conflation of these concepts has lead to unnecessary confusion and a lack of clarity in the literature and the grounding debate, and as a result our understanding of grounding has been hampered and led to a lack of progress towards the development of genuinely intelligent systems that know what they are doing and that can build their own representations autonomously via their own experiences, rather than relying on human designers to interpret the world and craft their representations for them.

It is fair to say that the reason logic-based methods have come to dominate the field of Knowledge Representation is because they provide mechanisms that help "make sense" of information by providing guidelines that minimize the number of interpretations, the number of representations that can be legally constructed, or provide a truth-based semantics which can be used to interpret information using a straightforward procedure. In addition significant strides have been made in areas like databases and the Semantic Web because the representation-information dichotomy underlies conceptual schema that embed "meaning" into information so that it can be made sense of and converted to representations during subsequent processing. In these highly constrained relatively simple systems there is often a 1:1 relationship between information and representation, but this relationship is lost when we move outside simple domains into more open, complex and dynamic environments of the kind that mobile robots inhabit. There is little work that explicitly recognizes the difference between information and representation in the field of robotics.

$$\frac{\partial^2 \Psi}{\partial x^2} + \frac{8\pi^2 m}{h^2} (E - V) \Psi = 0$$

Shrodinger Wave Function

Position Energy Potential Energy

Second derivative with respect to X

Fig. 1. Schrödinger's Equation

To emphasize the need to recognize and address the issues that the information-representation dichotomy raises consider Schrödinger's Equation (in Fig. 1.), the source of a rich representational model that predicts the future behavior of a dynamic system; written on a piece of paper it is merely a source of information. In most of the literature no distinction is made between the model as information in the external world and the model as representation in the mind of a cognitive agent like a person. There are many levels, perspectives and lens through which we can ground the information encapsulated in Figure 1; the pixel level, the character level, the general mathematical lens, or the physics lens. The information conveyed in Figure 1 would almost certainly be grounded differently and therefore converted into different representations, by a child, an AI researcher, a mathematician, a chemist, a physicist, an historian, and a cat.

2 The Grounding Problem in Retrospect

In this section we provide a brief history of the progress towards an understanding of grounding, and we argue that the next best constructive step is the development of a framework based on well defined concepts which can be used to design and analyse grounding capabilities, and to explicitly compare grounding across systems and agents. The key purpose being to develop cognitive agents, like robots, that can build and manage their own representations all by themselves, i.e. autonomously.

Grounding has been occurring at least as long as there have been living systems on the Earth. One could take the view that evolution has been busy at work pursuing the challenge of inventing increasingly better designs and technologies for grounding, which so far has culminated in the human body and mind.

In contrast to Brooks we consider cockroaches to have representations that help them determine how to respond to information captured by their senses, e.g. light, chemicals, and physical surfaces. There is no evidence that cockroaches develop and maintain world models, in the same way that people or a robot does, however, they do have representations that allow them to interpret information impinging on their senses so as to support their behaviour and survival. Even simple nematodes such as *Caenorhabditis Elegans* have a rudimentary memory which they use to represent chemical gradients and to control chemotaxis [8]. Clearly the *better* a systems' grounding capability relative to its environment, the more successful it will be and the more likely it will multiply and pass its DNA to the next generation. Evolution rewards and propagates well grounded systems over poorly grounded systems where a

well grounded system has or creates high quality representations that help it make sense of its inner and outer worlds [8]. If a system has a memory, it probably has representations that need to be grounded. Representations can take a wide range of forms, we are not suggesting they are necessarily symbolic, and our approach makes no particular assumptions about the form or content of representations. Our approach can be used to analyse grounding capabilities of cockroaches just as well as the Mars Rovers, Spirit and Opportunity, or to compare a chimpanzee's grounding capabilities with that of a human's [22].

The Ancient Greeks were fascinated by grounding; they pondered the relationship between ideas and reality incessantly and obsessively. We have made some progress in understanding grounding over the ensuing millennia but a thorough and comprehensive understanding still alludes us today. Socrates, Plato, and Aristotle were interested in the relationship between physical reality and the reality of ideas; they gave ideas the highest value and saw physical reality as the imperfect reflection of ideal reality. From this base scientists and philosophers across the ages have explored grounding. Undoubtedly, Galileo had a sophisticated and unique understanding of the interplay between concrete abstract ideas and reality due to his broad knowledge and experience in mathematics, theoretical physics, experimental physics, and astronomy. So it is a great pity that he lived during the Roman Inquisition which meant that he had to obfuscate many of his theories and thoughts on such matters. A younger contemporary Rene Descartes, however, was able to propose a dualistic' understanding of reality, and later Leibnitz considered reality as mental phenomena and insightfully wrote "... matter and motion are not substances or things as much as they are the phenomena of perceivers, the reality of which is situated in the harmony of the perceivers with themselves (at different times) and with other perceivers."

From a more formal perspective Tarski invented a theory of truth based on syntax and semantics [17], and for his system *truth is reality*. Peirce wanted more and developed his theory of semiotics with the extra feature of a referent. According to Davidson, a Tarskian *theory of truth* for a language is *a theory of meaning* for that language. The Deflationists believe that to assert a statement is true is the same as asserting the statement itself [18].

Albert Einstein, like Leibnitz, was also a phenomenalist and playfully stated "Reality is merely an illusion, albeit a very persistent one." Feynman viewed the world as teeming with information like energised waves and particles whizzing around us of which only a tiny percentage is captured by our sensors. He often noted that there was a close relationship between the laws of physics and reality, and that the laws of physics were only ever mere conjectures, since they could never be proved correct, only proven false when contradicted by experimental evidence. One of the more inventive approaches to describing reality is the Clifford Pickover's Hawking Reality Scale [14] where reality as people experience it is 0 on the exponential Reality Scale, and successive 10 units on the scale account for a reality one hundred times more distorted than the last. Hawking himself, continues to maintain that something as incomprehensible as quantum mechanics, "should concern us all, because it is a completely different picture of the physical universe and of reality itself".

Searle reinvigorated the grounding debate in Artificial Intelligence with the now well known and provocative Chinese Room thought experiment [15] where a non-Chinese speaking man trapped in a room, answers questions in Chinese by following

stepwise instructions with access to a Chinese questions and answers database. Apart from the fact that, such a database does not exist, Searle convincingly demonstrated that the man gave the impression he understood Chinese, but in actual fact he was unable to attribute meaning to the questions encoded in Chinese directly. Instead he was merely following instructions and had absolutely no understanding of the content of the questions or the answers. It is noteworthy that although he was unable to attribute the same meaning to the Chinese characters as say a Mandarin speaker might, he was able to ground the characters in order to match the correct response.

Brooks [2, 3] attacked the high level symbolic approach, the dominant approach at the time, in an attempt to make a case for his layered activity based bottom-up approach to intelligence. According to Brooks “the world is its own best model” [3]. He is a strong proponent of the need to ground systems in the physical world, and he continues to maintain that it is only through physical grounding that a system can give meaning to its processing. Harnad introduced the term “symbolic grounding” and appears to have taken the debate off in the direction of symbolic processing and something he calls “intrinsic meaning” [10]. Steels [19] has argued that labelling the grounding problem, the symbol grounding problem, has lead to widespread confusion because there is no unique or widely accepted interpretation of “symbol” and a broad range of interpretations span researchers from AI to Linguistics. In particular, the nonsymbolic systems described by Harnad, namely neural nets, are implemented as computer programs and one could argue that they are as symbolic as a logic-based knowledge base or an Arabic dictionary.

Rapid recent progress in the field of robotics has raised the grounding problem once more [5, 9], since intelligent robots need to ground information and build representations autonomously. The main challenge is how to discover how to make representations from information so that we can build artificially intelligent systems that can do it. For humans, a representation is the result of grounding information where grounding involves *making sense* of information. Not only do humans make sense of raw information from sensors, but we are aware we do it, and are even conscious of some of our representation making. Unfortunately, we are not typically aware of *how* we do it and as a result trying to replicate it in artificial systems is a challenge.

The importance of the distinction between representation and information is cleverly illustrated by the Belgian Surrealist René Magritte in his painting reproduced in Figure 2 below. First the object of interest is the image/picture on the page which is just a bunch of pixels/ink dots organized in a certain way. Consider a computer-based photocopy machine which is able to capture the information in the image and reproduce it with a high degree of fidelity, and able to make an almost exact copy. Should we say it can make representations? It has a form of memory/storage where the captured information takes on meaning for the machine, so it seems reasonable to suggest that a modern photocopier can represent attributes of pixels in an image. At the other end of the spectrum people can trivially “make sense” of the pixels and “see” an object, some might recognize as a pipe, some can represent and read the French text “*Ceci n'est pas une pipe*”. Some people may ignore the text, others try to interpret it; some may conclude it is a mistake, a contradiction, a joke, or an allusion, and others may see truth in the statement, after all a picture of a pipe, is not a pipe. Presented with this picture reflected photons impinge on the rods and cones in our further processing in the brain to create a conscious representation.



Fig. 2. Ceci n'est pas une pipe. (This is not a pipe) by René Magritte 1926.

The information continuously captured by our senses is used to produce our perceptions, concepts and knowledge to produce rich and complex representations, and we are only conscious of some of it, mostly the end product of all the complex processing. We are constantly bombarded with information; it impinges on our senses in steady streams and fluctuating bursts. Our grounding capability meticulously finds patterns in the information streaming in through our senses, and we carefully (and magically) massage the information we capture from multiple sensors into (meaningful) representations of physical and abstract things that we are able to subsequently comprehend and interpret.

There is a significant difference between information and representation; one can think of them as the input and output to a grounding process which for many applications could be conceived as a service and, therefore, it makes sense to consider the quality of a grounding service. Williams *et al* develop a Grounding Framework for evaluating how well a system grounds its information in [22].

3 Information and Representation

Information is non-representational; it creates the so-called raw feels that our senses detect internally and externally, and also particles/waves/energy we cannot detect with our limited human senses, e.g. radio waves. We are immersed in a sea of information and we are able *to sense* some directly like electromagnetic radiation in the so-called visible part of the spectrum, *to conceive and imagine* information beyond our senses such as radio waves and *to create* information in our own minds that we sometimes transpose for others to sense and make sense of such as the drawing of a dragon. We place no restrictions on information; it can be anything: a photon, a particle, a wave, a physical object, an abstract object, a class, an event, an action, relationship, or a representation.

Making sense of information creates representations. Making sense can involve recognizing simple patterns like a snake from information impinging on its heat sensors or a university chemistry student trying to understand the wave function, ψ , in Schrödinger's Equation. Representations for our purposes are open and the only requirement is that they ground some information. Representations can be generated from sensorimotor information, logic-based information, linguistic expressions, artefacts like a web page, a cave painting, a wristwatch, a hammer. Representations of physical objects are important but so too are representations of abstract entities such

as object functionalities and relationships between objects, and descriptions of ways to interact with specific objects such as affordances.

For the purpose of understanding the process of grounding it is insightful to classify representations into two important classes: *cued* and *detached* [6]. Cued representations are associated with the perception of things that are present, and detached representations are associated with entities that are not currently perceived. Sensations are immediate sensorimotor impressions, perceptions are grounded sensorimotor impressions, and simulations are detached representations. Sensations provide systems with information from the external (e.g. environment) and internal (e.g. body) world that a system can use to build representations and awareness. They exist in the present, are localised in the body/system, and are modality specific, e.g. visual, auditory, not both. Perceptions can ground more information than raw sensorimotor information, e.g. they can merge information from multiple sources to create multi-modal representations. They can represent accumulated sensorimotor information and sensorimotor information reinforced with simulations [6, 8]. Sensations are raw feels and involve signals from external sensors or from inside the system itself, but perceptions can use additional information derived from previous experiences and/or outcomes of learning. In contrast to sensation, perception can be cross-modal and can generate persistent representations which give raise to important cognitive capabilities such as object permanence. A detached representation is a simulation or abstract concept; something that can be utilized regardless of whether what it represents is present or not. A detached representation can even stand for something that does not exist at all. It turns out that humans readily use detached representations however other animals tend to use cued representations, and as a result they require a less responsive memory and they take fewer things into consideration when generating future possibilities during planning [8, 13] because they tend to plan with cued representations. A cognitive agent using detached representations will be more powerful in terms of explanation, prediction, planning, and collaboration provided they have high quality grounding capabilities and sufficient information processing power to build and manage them. Representations can be generated from information that has been gathered from a wide range of sources e.g. internal and external sensors, internal and external effectors, external instruments, and external systems. In addition they can result from fusing sensorimotor information with high level representations such as perceptions, concepts and linguistic expressions. Consider a doctor who not only grounds his own sensorimotor information, but information acquired from colleagues, books, lab tests, instruments that measure temperature, heart beat, blood pressure and oxygen content of the blood. Detached representations are extremely powerful. They can be manipulated independently of the external world, i.e. can be conceived and do not need to be directly perceived. Some examples of important detached representations are absent objects, past and potential future world states.

4 Grounding Capabilities

Grounding impacts how and what representations are created. For example, for a system to plan it needs representations of potential future needs and future world states [6, 8]. Chimpanzees can construct elaborate but limited plans where elements in the plan are usually constructed from cued representations of objects. Our current understanding suggest that chimpanzees do not tend to make representations of future

needs based on the observation that they do not tend to carry tools such as stones that can be used to crack nuts, from place to place [6].

We define the process of *grounding* to be the construction, management and interpretation of representations. Systems from airline reservations to autonomous mobile robots rely on well grounded representations [22]. A grounding capability provides basic infrastructure for cognition and intelligence. Consequently, what, how, and how well, information is grounded is of significant interest and crucial importance in AI. Grounding can be goal specific, domain specific, context specific, and system specific. Having defined grounding and grounded information we can focus on exploring the nature of the quality of the representation. A representation is created by a system's grounding capability, i.e. the capability to make sense of information.

Clearly, the ability to make representations is closely related to what we call intelligence. In order to build a truly intelligent system we need to ensure that once deployed the system can generate high quality representations on the fly without the assistance or direct intervention of humans, even in novel situations.

Cognitive agents manage representations about their internal states and the external world they inhabit; they use these representations to perform and achieve their design goals. As the complexity of the world they seek to exploit increases they have a growing need for more sophisticated methods to generate and manage representations effectively over time in the face of uncertainty, inconsistency, incompleteness, and change in complex and dynamic environments.

Significant work in the area of logic-based AI has been carried out in the quest to address these issues, however how these high level constructs are grounded so that systems can manage them autonomously over time in complex and dynamic environments remains an important and open challenge. Much of AI explores agent beliefs and many of the problems belief management faces stem from the conflation of the information and its corresponding representation.

In Intellectual Property Law ideas and their expression are treated entirely differently. This is known as the idea-expression dichotomy [20]. For example, one cannot copyright an *idea*, only the *expression of an idea*. In AI however, the expression of belief is often treated as the belief, but *isa(John,father)* is not a belief, in the same way the object depicted in Figure 1 is not a pipe. Beliefs are *ideas* while *isa(John,father)* and the pipe as depicted in Figure 2 are *expressions of ideas*. The distinction between a representation and information, or idea and expression, becomes crucially important when we try to design and build any intelligent system, not just AI systems on robots.

Having defined the process of grounding and an information-representation dichotomy, we can focus on the quality of the process of grounding. In order to be intelligent a system needs to be suitably grounded. Poorly grounded representations give rise to deluded systems that exhibit poor quality decision making, an inability to communicate and collaboration, and potentially dangerous behaviour. Representations are grounded information and are said to possess the property of *groundedness*. Groundedness is rich and complex, and it has a definitive impact on the possible attributes and behaviours that a system can display, and capabilities it can possess. Clearly, systems are not just grounded or ungrounded, but they can exhibit degrees of groundedness in a range of different dimensions. It makes sense to say of two grounded systems that one is more grounded than another.

5 Evaluating Grounding

The grounding process needs to be analysed and assessed on three important aspects: the construction, the management and the interpretation of representations. For representation construction we note that most contemporary artificially intelligent systems do not construct, or even manage their own representations, instead the mechanisms to do so are crafted by human designers; little of the construction and management mechanism is autonomous even in agents that use unsupervised machine learning. Once human designers deploy the mechanisms for systems to build, manage and interpret representations, systems can perform highly sophisticated tasks and their performance reflects the abilities of the designers.

A crucial ingredient for intelligence is the ability to make sense of information, in other words to ground information into representations. Even Brooksian robots, like Genus Khan, attempt to make sense of the world and produce representations in this sense. Many aspects of intelligence rely on representations for example:

- Knowing what you know, e.g. *I know that my son was born in 1998*, what you don't know, e.g. *where my son is right now*, what you are doing e.g. *writing*, how you are doing it e.g. *on a computer*, why you are doing it e.g. *its fun*, what your goals are, e.g. *to convey innovative ideas*, how to achieve your goals, why they are your goals, etc.
- Knowing what others know, what others don't know, what others are doing (and how and why), what interests others and what will get their attention (and why).
- Anticipating future developments; intelligent agents need to anticipate events and eventualities by exploiting their ability to represent and rehearse the future; to conduct risk assessment and risk management.

Intelligence involves various abilities including intention, motivation, adaptability, reasoning, foresight, insight, reactivity, proactivity, creativity, problem solving, judgement, decision making, planning, strategizing, learning, empathy, communication, collaboration. The quality of these powers and abilities is directly dependent on the quality of the systems' grounding capabilities, the representations they make and their ability to interpret them. High quality grounding is crucial to intelligent behaviour; moreover the ability to reason about other agents grounding capabilities is crucial for meaningful communication, and effective collaboration.

Contemporary robots do not experience the world like us because their grounding capabilities are significantly different to our own. For example, as illustrated in Figure 3 a typical soccer playing AIBO lives in a colour coded world and does not have stereo vision, and as a result the its perception of the objects like the ball is as a 2D blob in an image, and not as a 3D sphere. It is important to note that even though the soccer playing robots live in a colour coded world colours are completely dependent on ambient lighting conditions so that lowering the ambient lighting makes the ball appear a darker colour and eventually not orange at all. People tend to conceptualise colour as an intrinsic property of objects but it is not intrinsic at all.

Our perceptions of reality are uniquely human, and when we design and analyse the grounding capabilities of other systems, like soccer playing robots, we often implicitly compare them to ourselves and fail to clearly identify the assumptions we make in that comparison. Reality is experienced and represented (perceived and



Fig. 3. (a) Raw image derived from a robots camera (b) Perceptual representation of the ball, beacon and

conceived) differently by different systems; it is dependent on sensory, motor and cognitive capabilities. Systems with similar grounding capabilities probably experience the world similarly, e.g. you and me; chimpanzees and gorillas. Humans have a tendency to anthropomorphise other agents; this essentially amounts to projecting our grounding capabilities onto other entities like Honda’s humanoid Asimo, as a means to design and interpret their behaviour.

We conceptualize grounding as a capability since it can be more than a transformative step by step process; it can be a complex capability that is experienced and distributed. A grounding capability could be an agent or a group of agents like a team or organization. It is sometimes useful to formally describe grounding capabilities and we use a simple relational description because we want to illustrate the main ideas and to build agents that can reason about the grounding capabilities of other agents.

Grounding capabilities are those capabilities required to produce representations. We say grounding capability G grounds information I to generate representation R , and denote it as $\text{grounds}(G, R, I)$ and we define a relation *represents* such that $R = \text{represents}(I)$. Note, some special cases may require the relation *represents* to be a function and clearly I may be a label for the actual information, e.g. when I is a form of energy. Consider the pain *Agent007* feels when someone punches him, we can say $\text{grounds}(\text{Agent007}, \text{pain}, \text{punch_impact})$, but the string “pain” hardly represents the pain he feels, and the word punch leaves much unrepresented about that action, especially if *Agent007* would like to anticipate and avoid it next time. These difficulties and deficiencies need to be addressed in system design, serious problems arise if we just say ‘pain’ = pain and ‘punch impact’ = punch_impact as the Deflationists do. In robotic system design in particular it is often crucially important to identify and describe grounding via experience and to analyse a particular grounding or representation by making it explicit. Other examples include $\text{grounds}(\text{Agent007}, \text{“Doctor”}, \text{image(doctor.jpg)})$ which says *Agent007* uses the word “Doctor” to represent image information contained in a file called *doctor.jpg*. We can also use *grounds* to make statements about a common or social grounding which is a shared among a group of agents such as the Australian Medical Board (AMB) $\text{grounds}(\text{AMB}, \text{“Doctor”}, \text{image(doctor.jpg)})$.

Conceptualizing grounding as a capability raises the following obvious and important research questions: (i) what are the important features of a grounding capability, (ii) how can different grounding capabilities be compared and measured, (iii) how can we understand grounding capabilities better, (iv) how can we reason about the different grounding capabilities, and (v) how can we construct a grounding capability for artificially developed systems so that they can build their own representations.

Ownership of representations is important and the ability to reason about ownership and privacy for example is a powerful ability.

On this view if grounding creates representations from information then the search for better representations for humans and computers is a valuable line of research, but a more valuable challenge to pursue is a better understanding of how information can be grounded to create representations by an artificial system through its own experience all by itself. The relationship between making representations and intention also raises many important research questions.

We need to develop tools and techniques that help us explore the process of grounding, only then will we be able to develop systems that can ground themselves effectively, i.e. generate, manage and interpret their own high quality representations.

6 Discussion

A poorly grounded system will struggle to exhibit intelligence, no matter how sophisticated its decision making ability. Despite its significance, a trawl of the literature soon highlights that grounding is still a poorly understood concept and capability. Even worse confusion abounds. Similarly for intelligence, for example, there is no widely accepted definition of intelligence – in this paper we have argued that intelligence involves the ability to make representations. Given the importance of grounding and its major impact in system design it is surprising that it is rarely addressed head. Progress towards building intelligent systems with rich flexible representations has been retarded because the concept of grounding is complex and poorly understood.

The quality of representations singularly determines the choice, quality and power of problem solving approaches and the scope for creativity, innovation, and intelligence. The field of AI has taken some impressive steps towards building systems that can ground themselves but in order to take those achievements to the next level we need to have a better understanding of grounding so as to build robust autonomously grounded systems that can adapt and respond appropriately in novel and unexpected situations.

We provided an innovative conceptualization of representation that delivers significant value given there was no such conceptualization previously; it identifies an important distinction between information and representation much like the distinction between an idea and its expression in Copyright Law. Furthermore, it improves our understanding of intelligence and cognition, and has implications for systems design. In addition, it highlights the need for future research to operationalise the operator *grounds*, since it is this operator that makes representations.

We argue that without a clear and deep understanding of grounding, and practical methods for systems to use to autonomously make representations the achievement of Artificial Intelligence will remain a dream and beyond our grasp.

Acknowledgments. This paper was written while the author was funded on a Pauli Fellowship at the Technical University of Vienna which was made possible by Georg Gottlob and Thomas Eiter. The author is extremely grateful for the opportunity to be part of this prestigious fellowship program.

References

1. Barsalou, L.: Perceptual Symbol Systems. *Behavioural & Brain Sciences* 22, 577–660 (1999)
2. Brooks, R.A.: The Engineering of Physical Grounding. In: Proceedings of 15th Annual Meeting of the Cognitive Science Society, pp. 153–154. Lawrence Erlbaum, Hillsdale (1993)
3. Brooks, R.A.: The relationship between matter and life. *Nature* (6818), 409–411 (2001)
4. Chang, M., Dissanayake, G., Gurram, D., Hadzic, F., Healy, G., Karol, A., Stanton, C., Trieu, M., Williams, M.-A., Zeman, A.: Robot World Cup Soccer 2004: The Magic of UTS Unleashed! (2004), <http://www.unleashed.it.uts.edu.au/TeamReports>
5. Chen, X., Lui, W., Williams, M.-A. (eds.): Special Issue on Practical Cognitive Agents and Robots. *Journal of Autonomous Agents and Multi-Agent Systems* (in press, 2009)
6. Gärdenfors, P.: How Homo became Sapiens. MIT Press, Cambridge (2004)
7. Gärdenfors, P., Williams, M.-A.: Reasoning about Categories in Conceptual Spaces. In: Proceedings of the IJCAI. Morgan Kaufmann, San Francisco (2001)
8. Gärdenfors, P., Williams, M.-A.: Communication, Planning and Collaboration based on Representations and Simulations. In: Khlethos, D., Schalley, A. (eds.) *Language and Cognitive Structure*, Benjamins, p. 56 (2007)
9. Gärdenfors, P., Williams, M.-A.: Building Rich and Grounded Robot World Models from Sensors and Knowledge Resources: A Conceptual Spaces Approach. In: Proc. of International Symposium on Autonomous Minirobots for Research & Edutainment (2003)
10. Harnad, S.: The symbol grounding problem. *Physica D* 42, 335–346 (1990)
11. Karol, A., Nebel, B., Stanton, C., Williams, M.-A.: Case-Based Game Play in the RoboCup Four-Legged League: Part I The Theoretical Model. In: The Proceedings of the International RoboCup Symposium. Springer, Heidelberg (2004)
12. McCarthy, J., Hayes, P.J.: Some Philosophical Problems from the Standpoint of Artificial Intelligence. In: Michie, D. (ed.) *Machine Intelligence*, vol. 4. American Elsevier, Amsterdam (1969)
13. Newell, A.: Physical Symbol Systems. *Cognitive Science* V4 (1980)
14. Pickover, C.: Liquid Earth, <http://www.pickover.com>
15. Searle, J.: Minds, brains and programs. *Behavioural & Brain Sciences* 3, 417–457 (1980)
16. Sharkey, N.E., Jackson, S.A.: Grounding Computational Engines. *Artificial Intelligence Review* 10, 65–82 (1996)
17. Tarski, A.: The Concept of Truth in Formalized Languages, Logic, Semantics, Metamathematics. Oxford University Press, Oxford (1956)
18. Stanford Online Encyclopaedia of Philosophy, plato.stanford.edu
19. Steels, L.: Perceptually grounded meaning creation. In: Tokoro, M. (ed.) *Proceedings of the International Conference on Multi-Agent Systems*. AAAI Press, Menlo Park (1996b)
20. Miller, A.R., Davis, M.H.: Intellectual Property: Patents, Trademarks, and Copyright, 3rd edn. West/Wadsworth, New York (2000)
21. Trieu, M., Williams, M.-A.: Grounded Representation Driven Robot Design. In: Proceedings of the Robot Soccer World Cup (2007)
22. Williams, M.-A., Gärdenfors, P., McCarthy, J., Karol, A., Stanton, C.: A Framework for Grounding Representations. In: The Proceedings of IJCAI Workshop on Agents in Real-Time and Dynamic Environments (2005)

Learning from the Past with Experiment Databases

Joaquin Vanschoren¹, Bernhard Pfahringer², and Geoffrey Holmes²

¹ Computer Science Dept., K.U. Leuven, Leuven, Belgium

² Computer Science Dept., University of Waikato, Hamilton, New Zealand
`joaquin.vanschoren@cs.kuleuven.be, {bernhard, geoff}@cs.waikato.ac.nz`

Abstract. Thousands of Machine Learning research papers contain experimental comparisons that usually have been conducted with a single focus of interest, often losing detailed results after publication. Yet, when collecting all these past experiments in experiment databases, they can readily be reused for additional and possibly much broader investigation. In this paper, we make use of such a database to answer various interesting research questions about learning algorithms and to verify a number of recent studies. Alongside performing elaborate comparisons of algorithms, we also investigate the effects of algorithm parameters and data properties, and seek deeper insights into the behavior of learning algorithms by studying their learning curves and bias-variance profiles.

Keywords: machine learning, meta-learning, data analysis.

1 Introduction

“Study the past”, Confucius said, “if you would divine the future”. Also in machine learning, studying the results of earlier analysis is essential to gain a deeper understanding of learning methods.

As learning algorithms are typically heuristic in nature, learning experiments are needed to investigate the (combined) effects of different kinds of data, different preprocessing methods and, in many cases, different parameter settings. With so many factors influencing an algorithm’s behavior, these experiments are (or should be) quite general, which means that they probably have more uses than originally intended and, when brought together, may lead to many new insights.

To make these learning experiments available for future use, they can be submitted to an experiment database [1]. This is a database specifically designed to store learning experiments, including all details about the algorithms used, parameter settings, dataset properties, preprocessing methods, evaluation procedures and results. When new learning experiments, as well as meta-level descriptions of its components, are submitted to the database, they are automatically stored in a well-organized way, creating a detailed map of known learning approaches, their performance on past problems, and all known theoretical properties. All this information can then be accessed by writing the right database query (e.g. in SQL). As we will demonstrate, this provides a very versatile means

to investigate large amounts of experimental results, both under very specific and very general conditions.

The concept of experiment databases has been introduced before, and detailed information is available on how to design [1] and query [10] them, and how experimental information can be formally described in order to be submitted to public experiment repositories [11]. In this paper, we employ such a large, publicly available experiment database, with the goal of learning from those past results and answering a range of fundamental questions in machine learning research.

More specifically, we distinguish between three types of studies, increasingly making use of the available meta-level descriptions, and offering increasingly generalizable results¹:

1. Model-level analysis. These studies evaluate the produced models through a range of performance measures, but typically consider only individual datasets and algorithms. They typically try to identify HOW a specific algorithm performs, either on average or under specific conditions.
2. Data-level analysis. These studies investigate how known or measured data properties, not individual datasets, affect the performance of specific algorithms. They identify WHEN (on which kinds of data) an algorithm can be expected to behave a certain way.
3. Method-level analysis. These studies don't look at individual algorithms, but take general properties of the algorithms (eg. their bias-variance profile) into account, using these properties to identify WHY an algorithm behaves a certain way.

In the next section, we first give a short overview of the experiments available in the database. The three ensuing sections cover the different types of studies mentioned above. Section 6 concludes.

2 A Repository of Learning Experiments

The experiment database used in this study contains about 500.000 experiments of supervised classification. It stores 54 classification algorithms from the WEKA platform[12], together with all their parameters. It also holds 86 commonly used classification datasets taken from the UCI repository, described by 56 data characteristics, most of which are mentioned in [7]. Moreover, it contains a range of preprocessed datasets created by sampling the original datasets by removing 10%, 20%,... of their instances, and by applying feature selection with Correlation-based Feature Subset Selection using a best-first search method [4].

As for the available experiments, it contains the results of running all algorithms, with default parameter settings, on all datasets. Furthermore, the algorithms SMO (a support vector machine trainer), MultilayerPerceptron, J48 (C4.5), 1R, Random Forests, Bagging and Boosting are varied over their most

¹ A similar distinction is identified by Van Someren [9].

important parameter settings². For all these algorithms, 20 sensible values were defined for each parameter³, and the algorithms were run using those values while keeping the other parameters on their default value. In the case of J48, Bagging and 1R, parameters were additionally varied randomly to achieve at least 1000 experiments per algorithm on each dataset. For all randomized algorithms, each experiment was repeated 20 times with different random seeds. All experiments were evaluated with 10-fold cross-validation, using the same folds on each dataset, and a large subset was additionally evaluated with a bias-variance analysis.

The database is publicly available on <http://expdb.cs.kuleuven.be/>. All SQL queries used in this paper (not printed here because of space constraints) are also available there, and most of the graphs can be instantly generated online as well.

3 Model-Level Analysis

In the first type of study, we are interested in how individual algorithms perform on specific datasets. This is the most common type of study, typically used to benchmark, compare or rank algorithms, but also to investigate how specific parameter settings affect performance.

3.1 Comparing Algorithms

To compare the performance of all algorithms on one specific dataset, we could select the name of the algorithm used and the predictive accuracy recorded in all experiments on, for instance, the dataset ‘letter’. For more detail, we can also select the kernel in the case of a SVM and the base-learner in the case of an ensemble. We order the results by their performance and plot the results in Fig. 1.

Since the returned results are always as general as the query allows, we now have a complete overview of how each algorithm performed. Next to their optimal performance, it is also immediately clear how much variance is caused by suboptimal parameter settings (at least for those algorithms whose parameters were varied). For instance, when looking at SVMs, it is clear that especially the RBF-kernel is of great use here, while the polynomial kernel is much less interesting⁴. However, there is still much variation in the performance of the SVM’s, so it might be interesting to investigate this in more detail. Also, while most algorithms vary smoothly as their parameters are altered, there seem to be large jumps in the performances of SVMs and RandomForests, which are, in all likelihood, caused by parameters that heavily affect their performance. Moreover,

² For the ensemble methods, all non-ensemble learners were used as possible base-learners, each with default parameter settings.

³ Reasonable ranges for these parameter values were chosen based on the experimenter’s experience with the involved algorithms, and can be retrieved from the database’s website.

⁴ RBF kernels are popular in letter recognition problems.

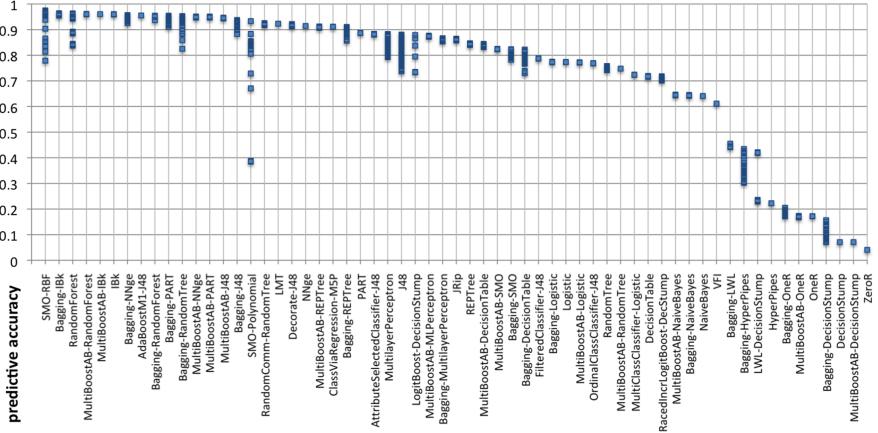


Fig. 1. Performance of all algorithms on dataset ‘letter’, including base-learners and kernels

when looking at bagging and boosting, it is clear that some base-learners are much more interesting than others. For instance, it appears that while bagging and boosting give an extra edge to the Nearest Neighbor algorithms, the effect is rather limited, and the same holds for Logistic Regression. Conversely, bagging RandomTree seems to be hugely profitable, but this does not hold for boosting. It also seems more rewarding to fine-tune RandomForests, MultiLayerPerceptrons and SVMs than to bag or boost their default setting. Still, this is only one dataset, further querying is needed. Given the generality of the returned results, each query is likely to highlight things we were not expecting, providing interesting cases for further study.

3.2 Investigating Parameter Effects

First, we examine the effect of the parameters of the RBF kernel. Based on the first query, we can zoom in on the SVM’s results by adding a constraint, and additionally asking for the value of the parameter we are interested in. Selecting the value of the gamma parameter and plotting the results, we obtain Fig. 2. While we are doing that, we can just as easily ask for the effect of this parameter on a number of other datasets as well.

When comparing the effect of gamma to the variation in RBF-kernel performance in the previous plot, we see that the variation corresponds exactly with the variation caused by this parameter. On the ‘letter’ dataset, performance increases when increasing gamma up to value 20, after which it slowly declines. The other curves show that the effect on other datasets is very different. On some datasets, performance increases until reaching a maximum and then slowly declines, while on other datasets, performance immediately starts decreasing up to a point, after which it quickly drops down to default accuracy. Looking at the number of attributes in each dataset (shown in brackets) seems to show some correlation, which we will investigate further in Sect.4.1.

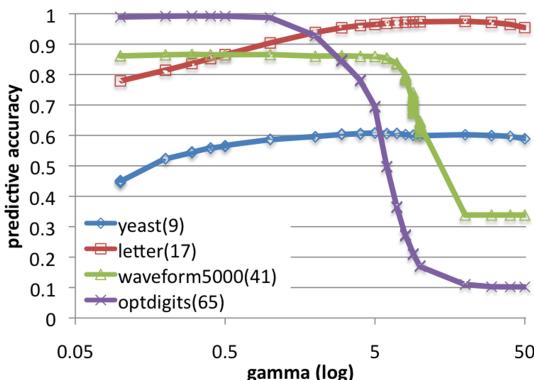


Fig. 2. The effect of parameter gamma of the RBF-kernel in SVMs on a number of different datasets, with their number of attributes shown in brackets

3.3 Ranking Algorithms

Previous queries investigated the performance of algorithms under rather specific conditions. Yet, by just dropping the constraints on the datasets used, we can query for their performance over a large number of different problems.

An interesting and sizable comparison of supervised learning algorithms was performed by Caruana and Niculescu-Mizil [3]. Most interestingly, this study compares across different performance measures by normalizing all performance metrics between the baseline performance and the best observed performance on each dataset. Using the aggregation functions of SQL, we can do this normalization right inside a query⁵.

To verify the conclusions of [3], we select all datasets, and all algorithms whose parameters were varied (see Sect.2). We also added naive bayes, logistic regression and 1NN, but only as a point of comparison, their ranking should not be interpreted as optimal. As for the performance metrics, we used predictive accuracy, F-score, precision and recall, the last three of which were averaged over all classes. We then queried for the maximal (normalized) performance of each algorithm for each metric, averaged over all datasets, and then averaged over all metrics to obtain the overall score for each algorithm. The results are shown in Fig.3.

Taking care not to overload the figure, we compacted groups of similar and similarly performing algorithms, indicated with an asterix (*). The overall best performing algorithms are mostly bagged and, to a lesser extent, boosted ensembles. Especially bagged trees⁶ perform very well, which corresponds nicely to [3]. Another shared conclusion is that boosting full trees performs dramatically better than boosting stumps. One notable difference is that RandomForests and

⁵ We normalized between the performance of the algorithm ZeroR and the maximum observed performance over all algorithms on each dataset.

⁶ These are PART, LMT, NBTree, J48 and similar tree-based learners.

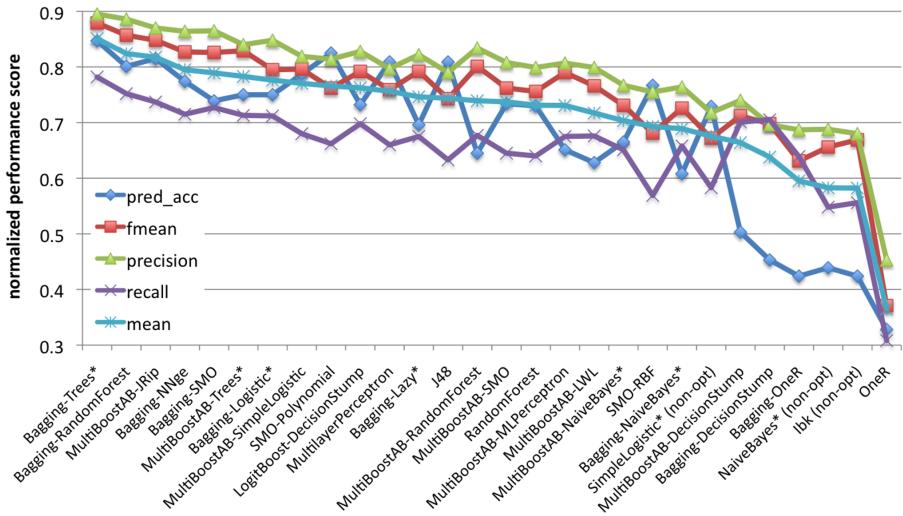


Fig. 3. Ranking of algorithms over all datasets and over different performance metrics

MultiLayerPerceptrons perform much worse in our study, while J48 seems to perform better. A possible explanation lies in the fact that we use a somewhat different set of performance measures in the ranking, although it may also depend on the (non-binary) datasets used. Furthermore, this study contains many more algorithms, in particular, the bagged versions of other strong learners (BayesNet, RandomForest, MultiLayerPerceptron, etc.) which perform very well in this ranking. Note however that these bagged versions primarily improve on precision and recall, while the original base-learners perform better on accuracy.

While this is a very comprehensive comparison of learning algorithms, each such comparison is still only a snapshot in time. However, as new algorithms, datasets and experiments are added to the database, one can at any time rerun the query and immediately see how things have evolved.

4 Data-Level Analysis

While the queries in the previous section allow the examination of the behavior of learning algorithms to a high degree of detail, they give no indication of exactly *when* (on which kind of datasets) a certain behavior is to be expected. In order to obtain results that generalize over different datasets, we need to look at the properties of each individual dataset, and investigate how they affect learning performance.

4.1 Investigating the Effect of Specific Data Properties

In a first such study, we examine what causes the ‘performance jumps’ that we noticed with the Random Forest algorithm in Fig. 1. Querying for the effect of

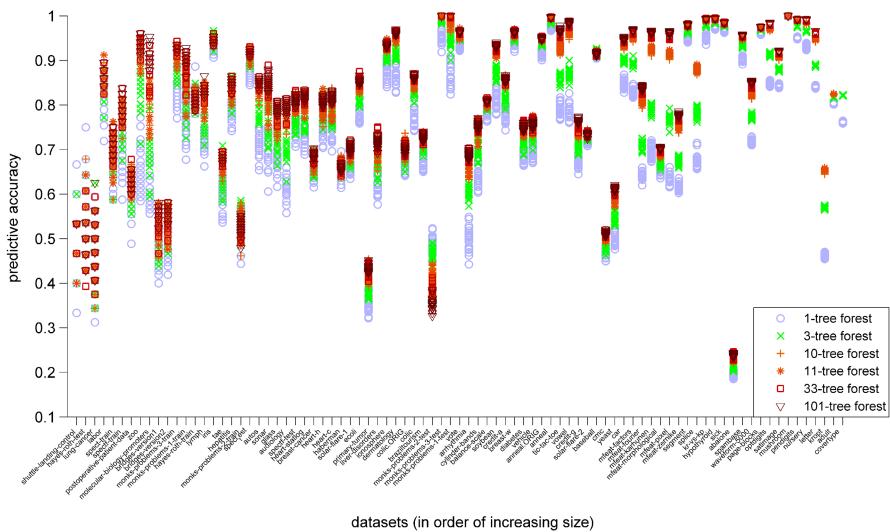


Fig. 4. The effect of dataset size and the number of trees for random forests

the number of trees in the forest on all datasets, ordering from small to large yields Fig. 4.

This shows that predictive accuracy increases with the number of trees, usually leveling off between 33 and 101 trees⁷. We also see that as dataset size increases, the accuracies for a given forest size vary less as trees become more stable on large datasets, eventually causing clear performance jumps on very large datasets. However, for very small datasets, the benefit of using more trees is overpowered by the randomness in the trees. All this illustrates that even quite simple queries can give a very detailed picture of an algorithm's behavior, showing the combined effects of parameters and data properties.

A second effect we can investigate is whether the optimal value for the gamma-parameter of the RBF-kernel is indeed linked to the number of attributes in the dataset. After querying for the relationship between the gamma-value corresponding with the optimal performance and the number of attributes in the dataset used, we get Fig. 5.

Although the number of attributes and the optimal gamma-value are not directly correlated, it is clear that high optimal gamma values predominantly occur on datasets with a small number of attributes. A possible explanation for this lies in the fact that SMO normalizes all attributes into the interval [0,1]. Therefore, the maximal squared distance between two examples, $\sum (a_i - b_i)^2$ for every attribute i , is equal to the number of attributes. Since the RBF-Kernel computes $e^{(-\gamma * \sum (a_i - b_i)^2)}$, the kernel value will go to zero very quickly for large gamma-values and a large number of attributes, making the non-zero

⁷ `monks-problems-2-test` is a notable exception: obtaining less than 50% accuracy on a binary problem, it actually performs worse as more trees are included.

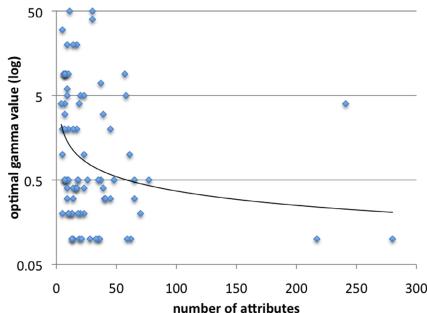


Fig. 5. The effect of the number of attributes on the optimal gamma-value

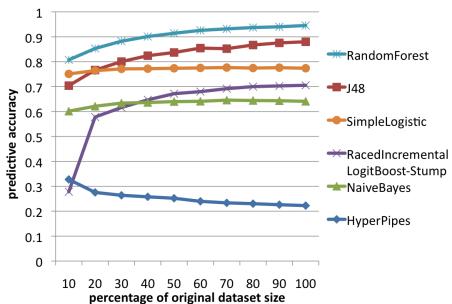


Fig. 6. Learning curves on the Letter-dataset

neighborhood around a support vector very small. Consequently, SMO will overfit these support vectors, resulting in low accuracies. This suggests that the RBF kernel should take the number of attributes into account to make the default gamma value more suitable across a range of datasets. It also illustrates how the experiment database allows the investigation of algorithms in detail and assist their development.

4.2 Investigating the Effect of Preprocessing Methods

Since the database can also store preprocessing methods, we can investigate their effect on the performance of learning algorithms. For instance, to investigate if the results in Fig. 2 are also valid on smaller samples of the ‘letter’ dataset, we can query for the results on downsampled versions of the dataset, yielding a learning curve for each algorithm, as shown in Fig. 6. It is now clear that the ranking of algorithms also depends on the size of the sample. While logistic regression is initially stronger than J48, the latter keeps on improving when given more data⁸. Also note that RandomForest is consequently better, that RacedIncrementalLogitBoost has a particularly steep learning curve, crossing two other curves, and that the performance of the HyperPipes algorithm actually worsens given more data.

4.3 Mining for Patterns in Learning Behavior

Instead of studying different dataset properties independently, we could also use data mining techniques to relate the effect of many different properties to an algorithm’s performance. For instance, when looking at Fig. 3, we see that OneR performs obviously much worse than the other algorithms. Still, some earlier studies, most notably one by Holte [5], found very little performance differences between OneR and the more complex J48. To study this discrepancy

⁸ This confirms earlier analysis by Perlich et al. [8], even though in that study, the dataset was transformed to a binary problem.

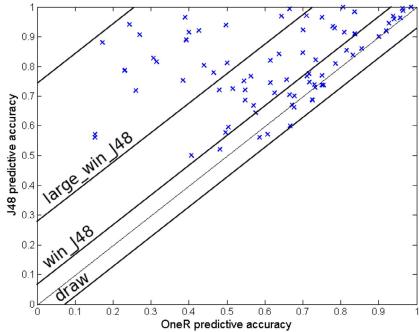


Fig. 7. J48’s performance against OneR’s for all datasets, discretized into 3 classes: draw (red), win_J48 (yellow), large_win_J48 (green)

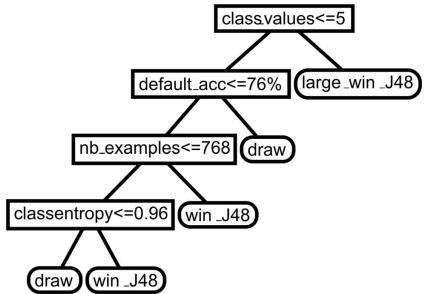


Fig. 8. A meta-decision tree predicting J48’s superiority over OneR based on dataset characteristics

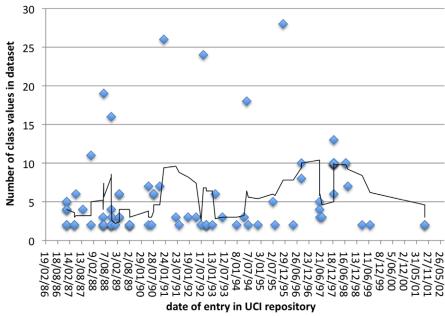


Fig. 9. Number of classes in UCI datasets over time and moving average

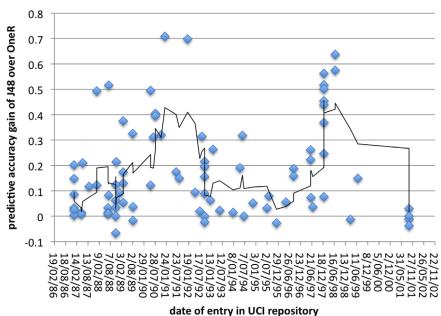


Fig. 10. Gain of J48 over OneR over time and moving average

in more detail, we can query for the default performance of OneR and J48 on all UCI datasets, and plot them against each other, as shown in Fig. 7. This shows that on some datasets, the performances are similar (crossing near the diagonal), while on other, J48 is the clear winner. Discretizing these results into three classes as shown in Fig. 7, and querying for the characteristics of each dataset, we can train a meta-decision tree predicting on which kinds of datasets J48 has the advantage (see Fig. 8). From this we learn that a high number of class values often leads to a large win of J48 over 1R [1].

When we query for the date (the date it was entered into the UCI repository) and number of classes of each dataset (see Fig. 9), we find a likely explanation for the earlier reported results. The datasets in [5] were all from the period 1988-1989, and few of those datasets have many class values. Fig. 10 displays the average gain of J48 over OneR per year, showing that depending on the period in which a study is performed, or better, the datasets known at that time, different results may be expected.

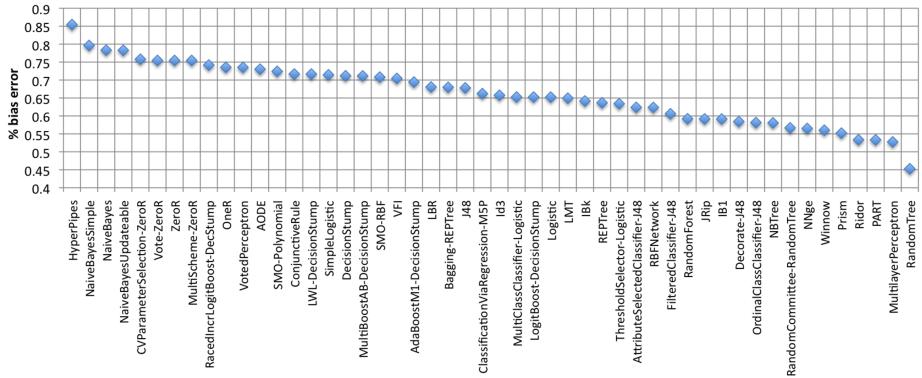


Fig. 11. The average percentage of bias-related error for each algorithm averaged over all datasets

5 Method Level Analysis

While the results in the previous section are clearly more generalizable towards the datasets used, they don't explain *why* algorithms behave a certain way. They only consider individual algorithms and thus do not generalize over different techniques. Hence, we need to extend the description of learning algorithms with a range of algorithm properties, and include these in our queries.

5.1 Bias-Variance Profiles

One very interesting property of an algorithm is its bias-variance profile[6]. Since the database contains a large number of bias-variance decomposition experiments⁹, we can give a realistic, numerical assessment of how capable each algorithm is in reducing bias and variance error. In Fig. 11 we show, for each algorithm, the proportion of the total error that can be attributed to bias error, using default parameter settings and averaged over all datasets.

The algorithms are ordered from large bias (low variance), to low bias (high variance). NaiveBayes is, as expected, one of the algorithms with the strongest variance management, but poor bias management, while RandomTree has very good bias management, but generates more variance error. When looking at the ensemble methods, it also shows that Bagging is a variance-reduction method, as it causes REPTree to shift significantly to the left. Conversely, Boosting reduces bias, shifting DecisionStump to the right in AdaBoostM1 and LogitBoost.

⁹ The database stores both Kohavi-Wolpert's and Webb's definition of bias/variance, but we use the former in our queries.

5.2 Investigating Bias-Variance Effects

As a final study, we investigate the claim by Brain and Webb [2] that on large datasets, the bias-component of the error becomes the most important factor, and that we should use algorithms with high bias management to tackle them. To verify this, we look for a connection between the dataset size and the proportion of bias error in the total error of a number of algorithms, using the previous figure to select algorithms with very different bias-variance profiles. Averaging the bias-variance results over datasets of similar size for each algorithm produces the result shown in Fig. 12. It shows that bias error is of varying significance on small datasets, but steadily increases in importance on larger datasets, for all algorithms. This validates the previous study on a larger set of datasets. In this case (on UCI datasets), bias becomes the most important factor on datasets larger than 50000 examples, no matter which algorithm is used. As such, it is indeed advisable to look to algorithms with good bias management when dealing with large datasets.

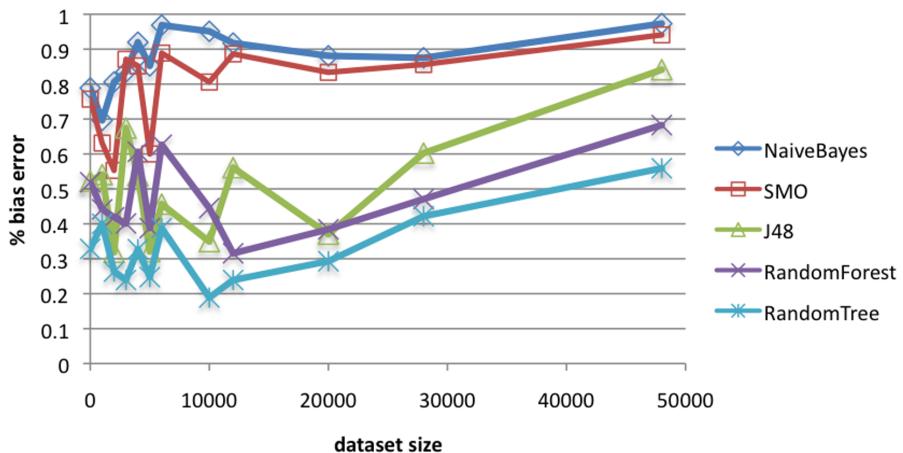


Fig. 12. The average percentage of bias-related error in algorithms as a function of dataset size

6 Conclusions

Much can be learned by looking at past learning experiments, and the creation of repositories of learning experiments provides an effective way of tapping into this information, often yielding surprising new insights or generating interesting research questions. In a series of increasingly in-depth studies, we first used such a repository to perform an elaborate comparison and ranking of supervised classification algorithms. Next, the available data characteristics were used to investigate their effects on learning performance and we discovered relationships that

suggest further improvements on learning algorithms, as well as meta-models of algorithm performance. Taking preprocessing methods into account, we also found crossing learning curves for several algorithms. Finally, we studied the bias-variance profiles of learning algorithms, and provided further evidence that managing bias error is particularly important on large datasets. We are confident that many more interesting results can be discovered by learning from past experiments. In the words of Albert Einstein, “Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning.”

References

1. Blockeel, H., Vanschoren, J.: Experiment databases: Towards an improved experimental methodology in machine learning. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenić, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 6–17. Springer, Heidelberg (2007)
2. Brain, D., Webb, G.: The Need for Low Bias Algorithms in Classification Learning from Large Data Sets. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS (LNAI), vol. 2431, pp. 62–73. Springer, Heidelberg (2002)
3. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: ICML 2006: Proc. of the 23rd Intl. Conf. on Mach. Learn., pp. 161–168 (2006)
4. Hall, M.A.: Correlation-based Feature Selection for Machine Learning. Ph.D diss. Hamilton, NZ: Waikato University, Department of Computer Science (1998)
5. Holte, R.: Very simple classification rules perform well on most commonly used datasets. Machine Learning 11, 63–91 (1993)
6. Kalousis, A., Hilario, M.: Building Algorithm Profiles for prior Model Selection in Knowledge Discovery Systems. Engineering Intelligent Systems 8(2) (2000)
7. Peng, Y., et al.: Improved Dataset Characterisation for Meta-Learning. In: Lange, S., Satoh, K., Smith, C.H. (eds.) DS 2002. LNCS, vol. 2534, pp. 141–152. Springer, Heidelberg (2002)
8. Perlich, C., Provost, F., Siminoff, J.: Tree induction vs. logistic regression: A learning curve analysis. Journal of Machine Learning Research 4, 211–255 (2003)
9. Van Someren, M.: Model Class Selection and Construction: Beyond the Procrustean Approach to Machine Learning Applications. In: Palioras, G., Karkaletsis, V., Spyropoulos, C.D. (eds.) ACAI 1999. LNCS (LNAI), vol. 2049, pp. 196–217. Springer, Heidelberg (2001)
10. Vanschoren, J., Blockeel, H.: Investigating learning behavior with experiment databases. In: Proceedings of Data Analysis, Machine Learning and Applications: GfKL, pp. 421–429. Springer, Heidelberg (2007)
11. Vanschoren, J., Blockeel, H., Pfahringer, B., Holmes, G.: Organizing the world’s machine learning information. In: Proceedings of Communications in Computer and Information Science: ISOLA 2008. Springer, Heidelberg (2008)
12. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

An Argumentation Framework Based on Conditional Priorities

Quoc Bao Vo

Centre for Complex Systems and Services (CS3)
Faculty of Information & Communication Technologies
Swinburne University of Technology, Australia
bvo@swin.edu.au

Abstract. We propose a framework to allow an agent to cope with inconsistent beliefs and to handle conflicting inferences. Our approach is based on a well-established line of research on assumption-based argumentation frameworks and defeasible reasoning. We propose a language to allow defeasible assumptions and context-sensitive priorities to be explicitly expressed and reasoned about by the agent. Our work reveals some interesting problems to conditional priority-based argumentation and establishes the fundamental properties of these frameworks. We also establish a sufficient condition for a conditional priority-based argumentation to have a unique stable extension based on the notion of stratification.

1 Introduction

Most information we obtain, via observation or communication, bears some degree of uncertainty. On the other hand, no one has complete knowledge of the world and about other agents. Such informational incompleteness together with the need for the agent to act effectively in its environment requires the agent to derive further information by some forms of reasoning. These forms of reasoning are generally defeasible as they are mostly based on the agent's common sense rather than hard facts and sound inferences.

There has been a vast literature, mainly within the field of Artificial Intelligence (AI), in which these issues have been discussed and addressed. These works include works in defeasible reasoning, non-monotonic reasoning, belief revision and belief updates, etc. More recently, works around argumentation frameworks have shown great promises for reasoning with inconsistent information. These are based on the construction and the comparison of arguments [8]. Given the defeasibility nature of arguments, it is perhaps not surprising that argumentation frameworks have been used to formalise and capture defeasible (or, default) reasoning [6,18].

Nevertheless, three problems remain to be addressed in these frameworks:

- i. The problem of *multiple extensions*: Under the reasoning framework, more than one consistent sets of beliefs (i.e., extensions) can be derived. Any pair of extensions put together will result in inconsistency. This problem presents the reasoner a *dilemma* as he might have to choose one extension to believe if he wants to maximise his information.

- ii. The problem of *none extension*: Under the reasoning framework, no extensions can be derived. Accepting any piece of information will lead to a contradiction. This problem presents the reasoner a *paradox* as none of the information can be consistently believed.
- iii. The problem of *context-sensitivity of priorities*: One piece of information can take precedence over another and, in case the two are conflicting, the former should be accepted. However, such priority information is context-specific, in the sense that it is only applicable in certain contexts [10].

The approach we introduce in this paper, bearing in mind the above problems, combines the following characteristics:

1. It employs an argumentation framework as the underlying engine for reasoning. This is because the abstract argumentation framework [8] is very generic and flexible. In particular, it is sufficiently powerful to allow most defeasible reasoning frameworks to be captured in an argumentation-theoretic approach [6, 18].
2. The defeasibility of the agent's reasoning and beliefs is rendered by assumptions. That is, all inference rules are applicable (and undefeasible), but the acceptability of a conclusion may be dependent on defeasible assumptions.
3. Context-sensitivity of priorities between defeasible information is rendered by explicit conditional priorities between defeasible assumptions.

The advantage of our approach is in a simple language to allow an agent's beliefs to be easily expressed and reasoning in such a knowledge base can be carried out using well-known techniques. Our handling of context-sensitive priorities based on defeasible assumptions is a novel aspect of our approach. As a consequence, our conditions are defined on the arguments, rather than by classes of logic. In addition, we also identify a class of well-behaving conditional priorities.

2 Background

Consider a propositional language \mathcal{L} over a finite alphabet of atoms, which is closed under the usual connectives \neg , \vee , \wedge , and \rightarrow . We use Cn to denote the consequence operation of classical logic. We often write $K \vdash x$ for $x \in Cn(K)$. Consider also a finite set \mathcal{A} of assumptions, such that $\mathcal{A} \cap \mathcal{L} = \emptyset$. An *inference rule* is of the form: $\alpha \Leftarrow \beta_1, \dots, \beta_m : \delta_1, \dots, \delta_n$ ($m, n \geq 0$), where $\alpha \in \mathcal{L}$ is the conclusion, or the head, of the inference rule, and $\beta_1, \dots, \beta_m \in \mathcal{L}$ are its *pre-requisites* and $\delta_1, \dots, \delta_n \in \mathcal{A}$ are its (*supporting*) *assumptions*. Furthermore, for each assumption $\delta \in \mathcal{A}$, there is a set of propositions, denoted by $\overline{\delta}(\subseteq \mathcal{L})$, that deny the applicability of the assumption δ . Observe that an assumption can be denied on several different grounds.

Several types of assumptions usually encountered in the literature of knowledge representation and reasoning include: *closed-world assumptions* (i.e., what is not currently known to be true is assumed to be false), *taxonomical assumptions* (e.g., birds normally fly), *assumptions about dynamic domains*, which allow an agent to conjecture about certain fluents and features of the world in relation to the actions performed in those domains (see e.g., [17]).

Let \mathcal{R} be a finite set of inference rules, an *argument* (wrt. \mathcal{R}) is a tree defined as follows: If $\alpha \Leftarrow \beta_1, \dots, \beta_m : \delta_1, \dots, \delta_n$ is an inference rule in \mathcal{R} and T_1, \dots, T_m are arguments whose heads are β_1, \dots, β_m , then the tree T whose root is labeled by α and $(m+n)$ subtrees are T_1, \dots, T_m plus the n leafs labeled by the assumptions $\delta_1, \dots, \delta_n$ is an argument; α is the *head* of this argument.

We will refer to a triple $(\mathcal{L}, \mathcal{A}, \mathcal{R})$ as an *assumption-based framework*.

We require that the head of each argument be logically consistent. That is, if h is the head of an argument then $Cn(h) \neq \mathcal{L}$. Let T be an argument, we denote by *head*(T) the head of T . Given an argument T , a subtree T' of T is a *sub-argument* of T . The set of all sub-arguments of an argument T is denoted by *sub*(T). Given an argument tree T , by collecting all propositions appearing on T , we have the set of beliefs supported by T , denoted by *theory*(T). If Θ is a set of arguments, we denote: $\text{theory}(\Theta) \triangleq \bigcup_{\theta \in \Theta} \text{theory}(\theta)$.

Example 1. Let us consider the following set of inference rules, representing an inheritance network, reproduced from [10]:

- $r_1 \neg m \Leftarrow s : \text{TxNorm}\langle s, \neg m \rangle$
- $r_2 m \Leftarrow a : \text{TxNorm}\langle a, m \rangle$
- $r_3 y \Leftarrow s : \text{TxNorm}\langle s, y \rangle$
- $r_4 a \Leftarrow y$
- $r_5 s \Leftarrow$

The normative interpretation of the above inference rules is that

- (r₁) “normally, students (s) are not married ($\neg m$),” where $\text{TxNorm}\langle s, \neg m \rangle$ expresses the taxonomical assumption of the normality of the class of “students” regarding the property “not married;” (We will provide the formal specifications of assumptions in the next section.)
- (r₂) “normally, adults (a) are married (m);”
- (r₃) “normally, students (s) are young adults (y);” and
- (r₄) “all young adults (y) are adults (a),” relying on no defeasible assumptions.

The desirable semantics here is represented by the model $M = \{s, y, a, \neg m\}$. To deliver this semantics, traditional priority-based approaches in the literature (e.g., [16,7]) assign rule r_2 a lower priority than rule r_1 (as r_1 is considered to be more *specific* than r_2). Let us consider now the marital status of another student who is an adult but not a young one. That is, the rule

- (r₆) $\neg y \Leftarrow$

is added to the above set of rules. Now, since y does not hold, rule r_1 can no longer be considered more specific than r_2 . Hence, it is intuitive to expect that neither m nor $\neg m$ should be concluded in this case. This is also the result sanctioned by all semantics of defeasible inheritance networks [15]. In any priority-based system in which the priorities between rules are insensitive to the contextual information (e.g., [16,7]), the model $M_1 = \{s, \neg y, a, \neg m\}$ would be considered to be of higher priority than the model $M_2 = \{s, \neg y, a, m\}$ due to the pre-imposed priority between the two rules r_1 and r_2 . That means priority-based approaches in the literature conclude $\neg m$ given the knowledge base $\{r_1, r_2, r_3, r_4, r_5, r_6\}$, which is not the intuitive result we expect.

We now proceed to elaborate one of the most fundamental notions of argumentation frameworks, namely the *attack relation*, which represents the defeasibility relation

between arguments. In this paper, we will consider two kinds of attack between arguments: An argument can (i) rebuts another argument by contradicting its conclusion, or (ii) undercuts another argument by denying one of its supporting assumptions.

Definition 1. Let T_1 and T_2 be two arguments.

1. T_1 **rebuts** T_2 iff $\text{head}(T_1)$ and $\text{head}(T_2)$ are inconsistent, i.e., $Cn(\{\text{head}(T_1), \text{head}(T_2)\}) = \mathcal{L}$.
2. T_1 **undercuts** T_2 at δ iff δ occurs as one of the supporting assumptions in T_2 and $\exists \gamma \in \overline{\delta} \cdot \text{head}(T_1) \vdash \gamma$.

Observation 1. The rebut relation is symmetric. That is, T_1 rebuts T_2 if and only if T_2 rebuts T_1 .

The notions of one argument attacking another is naturally extended to a set of arguments attacking an argument. For instance, a set of argument T_1, \dots, T_k rebuts an argument T if $\{\text{head}(T_1), \dots, \text{head}(T_k)\}$ is consistent and $\{\text{head}(T_1), \dots, \text{head}(T_k), \text{head}(T)\}$ is not. A set of arguments undercutting another argument can similarly be defined. However, if the inference rules of classical logic are included in \mathcal{R} then an attack of a set of arguments Θ on an argument T can be reduced to an attack on T by an argument T_Θ , which is derived from the arguments in Θ using the classical inference rules. In the rest of this paper, we will assume that \mathcal{R} includes all classical inference rules, e.g. from the Sequent Calculus. Furthermore, as we are concerned not only with any particular conclusion but also with all constructible arguments from a given background theory $K \subseteq \mathcal{L}$ and the appropriate sets of assumptions, we will be interested mainly on the defeasible relation between sets of arguments.

We are not particularly interested in self-conflicting sets of arguments as they are usually considered to be invalid.

Definition 2. A set of arguments Θ is **conflict-free** if there doesn't exist a pair of sub-arguments $T_1, T_2 \in \bigcup_{\theta \in \Theta} \text{sub}(\theta)$ such that T_1 attacks T_2 .

Observation 2. If a set of arguments Θ is conflict-free then $\bigcup_{\theta \in \Theta} \text{theory}(\theta)$ is consistent.

Definition 3. Let a conflict-free set of arguments Θ be given.

1. If T is an argument and Θ rebuts T or Θ undercuts one of the supporting assumptions of T , then Θ is said to **attack** T , denoted by $\text{attack}(\Theta, T)$.
2. If Θ' is a set of arguments and $\text{attack}(\Theta, T)$ for some $T \in \bigcup_{\theta \in \Theta'} \text{sub}(\theta)$ then Θ attacks Θ' , denoted by $\text{attack}(\Theta, \Theta')$.

Before we proceed, we will reproduce the basic argumentation-theoretic notions of acceptability of arguments, also known as the stability semantics and the preferability semantics.

Definition 4. Let a conflict-free set Θ of arguments be given,

1. Θ is **admissible** if it defends itself against all attacks on its members: $\forall \Theta'. (\text{attack}(\Theta', \Theta) \rightarrow \text{attack}(\Theta, \Theta'))$.

2. Θ is **stable** if it attacks every argument outside of Θ ; i.e., $\forall T \notin \Theta. (\text{attack}(\Theta, T))$.
3. Θ is **preferable** if it is a maximal (with respect to set inclusion) admissible set of arguments.

Recently, several semantics for argumentation have been introduced, including the defeasible semantics with ambiguity blocking by Governatori et al [14] and the ideal semantics by Dung et al. [9].

3 A Language of Assumptions and Priorities

Given that different types of assumptions play different roles in common sense reasoning (see Section 2), an assumption can (i) express a presumption about a single predicate (e.g., in the case of the Negation-As-Failure operator *not*), or (ii) express some presumed relationship between several predicates (e.g., in the case of the *normality* of a *bird* with respect to *fly*).

Our aim is to design a simple language to allow rules about the priorities between assumptions to be expressed. The assumptions take the following form:

assumption ::= assumption_name⟨parameter+⟩

where parameter is a member of the underlying language \mathcal{L} .

Examples of assumptions include:

- naf⟨atom⟩, where atom is an atomic proposition from \mathcal{L} , indicating the Negation-As-Failure assumption.
- TxNorm⟨atom, lit⟩, where lit is a literal from \mathcal{L} , indicating the taxonomical normality of a class of objects represented by atom regarding the property indicated by the literal lit.

Subsequently, let SoA1 and SoA2 denote non-empty sets of assumptions, the priority rules take the following form:

priority_rule ::= If C then $\text{SoA1} < \text{SoA2}$

where $C \in \mathcal{L}$ is the context condition in which the priority expression $\text{SoA1} < \text{SoA2}$ is applicable.

Example 1 (cont.) The defeasible knowledge base presented earlier can now be augmented with the following priority rule:

If y then $\{\text{TxNorm}\langle a, m \rangle\} < \{\text{TxNorm}\langle s, \neg m \rangle\}$

While the syntax of priority rules allows for comparison between sets of assumptions, their intended use is to compare between single assumptions. Nevertheless, most arguments involve more than one assumption. Thus we have to determine the ordering between collective sets of assumptions to allow arguments to be compared. We will introduce the relation \prec between sets of assumptions, possibly under some condition. Formally, the relation \prec is a ternary relation, taking three arguments: a formula (from \mathcal{L}) and two sets of assumptions. However, the first argument will be placed in the subscript of the relation \prec for the sake of presentation.

Definition 5. Let a set \mathcal{P} of priority rules be given. A set \prec is said to **satisfies** \mathcal{P} if:

1. $(C, \Delta_1, \Delta_2) \in \prec$ when [If C then $\Delta_1 < \Delta_2$] $\in \mathcal{P}$.

A set of priority rules \mathcal{P} can be inconsistent as the rules might state that under the same condition a set of assumptions is both of less and higher priority than another set of assumptions. We will write $\Delta_1 \prec_K \Delta_2$ instead of $(K, \Delta_1, \Delta_2) \in \prec$.

Definition 6. A set of priority rules \mathcal{P} is **possibly consistent** if there exists a relation \prec that satisfies \mathcal{P} such that $\forall \Delta_1, \Delta_2 \subseteq \mathcal{A}. \forall K \in \mathcal{L}. \Delta_1 \not\prec_K \Delta_1 \cup \Delta_2 \& (\Delta_1 \prec_K \Delta_2 \rightarrow \Delta_2 \not\prec_K \Delta_1)$.

A relation \prec that (i) satisfies a possibly consistent set of priority rules \mathcal{P} , and (ii) is minimal (wrt. set-inclusion), is called a *preliminary conditional priority relation* (or, **PCPR**) wrt. \mathcal{P} . Of course, the above definition of PCPRs is too weak to render a priority relation between sets of assumptions. For instance, if we have $K, K' \in \mathcal{L}$ and $\Delta_1, \Delta_2 \subseteq \mathcal{A}$ such that $\Delta_1 \prec_K \Delta_2$ and $K' \vdash K$ then it is natural to expect that $\Delta_1 \prec_{K'} \Delta_2$. Nevertheless, the above definitions of the PCPRs doesn't include such an extension. In order to allow the defined priority relations to cover such cases, we need to extend PCPRs as shown in the following definition.

A *subset coverage* of a set S is defined to be a set $\{S_1, \dots, S_k\}$ so that $\forall i \in \{1, \dots, k\}. S_i \subseteq S$ and $\bigcup_{i=1}^n S_i = S$.

Definition 7. Let a relation $\prec \subseteq \mathcal{L} \times \mathcal{A} \times \mathcal{A}$ be given, the **extension** of \prec , denoted by \prec^{Ext} , is defined as follows.

1. $\forall K \in \mathcal{L}. \forall \Delta \subseteq \mathcal{A}. \Delta \neq \emptyset \rightarrow \Delta \prec_K^{Ext} \emptyset$; that is, the empty set of assumptions is always of higher priority than any non-empty set of assumptions.
2. If $\Delta_1, \Delta_2 \subseteq \mathcal{A}$ are two non-empty sets of assumptions, and $K \in \mathcal{L}$, $\Delta_1 \prec_K^{Ext} \Delta_2$ iff there exists a subset coverage $\{\Delta_2^1, \dots, \Delta_2^k\}$ of Δ_2 so that

$$\forall j \in \{1, \dots, k\}. \exists \Delta'_1 \subseteq \Delta_1. \exists \kappa \in Cn(K). \Delta'_1 \prec_\kappa \Delta_2^j.$$

3. $\forall K \in \mathcal{L}. \forall \Delta_1, \Delta_2, \Delta_3 \subseteq \mathcal{A}. \Delta_1 \prec_K^{Ext} \Delta_2 \& \Delta_2 \prec_K^{Ext} \Delta_3 \rightarrow \Delta_1 \prec_K^{Ext} \Delta_3$.

Example 2. Consider the following set of inference rules:

$$\begin{aligned} \{p \Leftarrow; & \quad s \Leftarrow; \quad u \Leftarrow s; \quad q \Leftarrow p : \delta_1, \delta_2; \\ r \Leftarrow q : \delta_3; & \quad t \Leftarrow p : \delta_4; \quad \neg r \Leftarrow t, u : \delta_2, \delta_5\}. \end{aligned}$$

Assume that the set \mathcal{P} of priority rules contains the following two rules:

If u then $\{\delta_4\} < \{\delta_2\}$ and If $u \wedge p$ then $\{\delta_5\} < \{\delta_1, \delta_3\}$.

Then we have $\{\delta_4\} \prec_u \{\delta_2\}$ and $\{\delta_5\} \prec_{u \wedge p} \{\delta_1, \delta_3\}$. Moreover, after extending \prec we also have $\{\delta_2, \delta_4, \delta_5\} \prec_{u \wedge p}^{Ext} \{\delta_1, \delta_2, \delta_3\}$. Given the set of assumptions $\{\delta_1, \delta_2, \delta_3\}$, the argument for $[p, q, r]$ can be constructed. On the other hand, given the set of assumptions $\{\delta_2, \delta_4, \delta_5\}$, the argument for $[s, t, \neg r]$ can be constructed. Based on the the extended priority relation between the supporting sets of assumptions, the former argument can be considered to defeat the latter, allowing r to be accepted.

Essentially, while the proposed language allows the knowledge engineer to make simple statements about the priority relationship between assumptions (and thus, arguments), the above definition aims to complete such description as a legitimate priority relation. The first condition allows strict arguments (i.e., those that don't require the support of defeasible assumptions) to defeat non-strict arguments. The second condition allows a collection of priority rules to be put together so that sets of assumptions can be compared. The third condition enables the transitivity of the extended priority relation. Some properties of the extension of a PCPR can be established:

Proposition 1. *Let \prec be a PCPR, the following holds:*

1. $\prec^{\text{Ext}} = (\prec^{\text{Ext}})^{\text{Ext}}$.
2. $\forall K, K' \in \mathcal{L}. \forall \Delta_1, \Delta_2 \subseteq \mathcal{A}. (K \in Cn(K') \& \Delta_1 \prec_K^{\text{Ext}} \Delta_2) \rightarrow \Delta_1 \prec_{K'}^{\text{Ext}} \Delta_2$.

However, there is another disturbing observation about the extension of a PCPR:

Observation 3. *Assume that a possibly consistent set of priority rules \mathcal{P} is given. Let \prec be the PCPR wrt. \mathcal{P} , the relation \prec^{Ext} satisfying \mathcal{P} may not exist.*

The above Observation actually reveals a much deeper problem to the priority rules. Essentially, a possibly consistent set of priority rules can still be inconsistent.

Definition 8. *Let \prec be the PCPR wrt. a possibly consistent set of priority rules \mathcal{P} , the set \mathcal{P} is **consistent** if the extension \prec^{Ext} exists.*

If \mathcal{P} is consistent then the relation \prec^{Ext} is called the conditional priority relation wrt. \mathcal{P} and denoted by $\prec^{\mathcal{P}}$.

From Definition 6 and Definition 7, it's trivial to show that:

Lemma 1. *Given a consistent set of priority rules \mathcal{P} , the relation $\prec^{\mathcal{P}}$ is a strict pre-order.*

4 Acceptability of Arguments

The basic notions of acceptability of arguments in various argumentation frameworks have mostly been defined purely on the basis of other constructible and interacting arguments [12,11,8]. A number of recent works (e.g., [7,10,1,5]) have shown that preference and priority relations allow for more sophisticated and appropriate handling of conflicts and uncertain information. In this paper, we aim at taking these ideas to another level by combining (i) the internal construction of arguments (via defeasible assumptions and inference rules), and (ii) the interaction between arguments (via a defeasibility relation such as the attack relation between arguments), with (iii) other criteria of reasoning (possibly domain-specific) that are expressible in the proposed language for assumptions and priority relations between arguments.

Given that (i) the only defeasible part in an argument θ constructible in our framework is the supporting assumptions used in the construction of θ , and (ii) the priority rules allow the agent to compare arguments, conflicts can be resolved based on the relative strength of the conflicting arguments. However, there are two issues that need to be addressed before we proceed to define a notion of acceptability.

4.1 Priority Relations Can Interfere with Argumentation-Theoretic Semantics

Priority relations can help resolving conflicts. For instance, let A and B be two arguments such that A attacks B , and B attacks A . If we further know that A is of higher priority than B then this dilemma can be easily resolved as the attack of B on A can be deemed incredible, allowing A to be accepted. Subsequently, B should be rejected. On the other hand, priority relations can also interfere with an accepted resolution produced by argumentation-theoretic semantics. For instance, if we have three arguments A , B , and C so that A attacks B and B attacks C , most argumentation-theoretic semantics (e.g., stability semantics and preferability semantics) accept arguments A and C while rejecting B . But if according to some priority relation B is considered to be of higher priority than A then most preference-theoretic approaches would accept A (on the basis that it is not attacked by any known arguments) and B (on the basis that its only attacker, A , is weaker than itself and thus giving B an automatic self-defence against this attacker) while rejecting C . Argumentation-theoretically, this outcome (i.e., the set of accepted arguments $\{A, B\}$) is self-conflicting as A attacks B .

In general, the important requirement is that the agent's beliefs have to be always consistent. Note that, earlier in this paper, we distinguish between two types of attack, namely *undercutting* and *rebutting*, between arguments. When an argument T_1 undercuts another argument T_2 , T_1 essentially denies one of the supporting assumptions used in the construction of T_2 . And that doesn't necessarily result in an inconsistency. On the other hand, when two arguments T_1 and T_2 rebut each other, the conclusion of T_1 is essentially the negation of the conclusion of T_2 . Thus, it's not possible to accept both T_1 and T_2 without having an inconsistency. Fortunately it can be shown that, under any priority relation, if the arguments T_1 and T_2 rebut each other, they can not be both accepted under the standard argumentation-theoretic semantics. Before we proceed, we need to clarify what it means for an argument to attack another one given a priority relation.

Definition 9. Let \mathcal{P} be a consistent set of priority rules. Let $Asmptn(\theta)$ denote the set of assumptions occurring in the argument θ and $K \in \mathcal{L}$,

- Argument $T_1 \prec_K^{\mathcal{P}}$ -**rebuts** argument T_2 iff T_1 rebuts T_2 and $Asmptn(T_1) \not\prec_K^{\mathcal{P}} Asmptn(T_2)$.
- Argument $T_1 \prec_K^{\mathcal{P}}$ -**undercuts** argument T_2 at δ iff T_1 undercuts T_2 at δ and $Asmptn(T_1) \not\prec_K^{\mathcal{P}} \{\delta\}$.

Lemma 2. The $\prec^{\mathcal{P}}$ -rebut relation is asymmetric, i.e., it may be the case that $T_1 \prec_K^{\mathcal{P}}$ -rebuts T_2 and T_2 does not $\prec_K^{\mathcal{P}}$ -rebut T_1 .

Proposition 2. Let \mathcal{P} be a consistent set of priority rules and $K \in \mathcal{L}$, if an argument T_1 rebuts another argument T_2 then at most one of them is accepted under the stability semantics or the preferability semantics.

To summarise, priority-based defeasibility relation allows for the agent to accept arguments that may undercut each other while still maintaining consistency within her beliefs. We are now considering the second issue that is more native to our approach.

4.2 Context Conditions Are Interdependent with the Involved Arguments

As a reminder, conditional priority relations essentially assert that an argument is stronger (or more acceptable) than another argument under some context condition. For an agent to decide whether she should use this assertion, the context condition needs to be checked against the current beliefs held by the agent. However, these beliefs are derived from the constructible (and possibly conflicting) arguments that also include the arguments under consideration. We have a circularity problem!

These problems come in two critical forms, and we will consider them one after the other.

Cycles of Self-Defeating Arguments. As an example, let a consistent set of priority rules \mathcal{P} , a context condition $K \in \mathcal{L}$, and three arguments T_1, T_2 , and T_3 be such that (i) T_1 attacks T_2 , (ii) $Asmptn(T_1) \prec_K^{\mathcal{P}} Asmptn(T_2)$, (iii) T_2 attacks T_3 , and (iv) K is not derivable from the current beliefs, unless $head(T_3)$ is added to the current beliefs.

The paradox here is that, without $head(T_3)$ the conditional priority $\prec_K^{\mathcal{P}}$ is not applicable, allowing T_1 to defeat T_2 , which in turn stops the attack of T_2 on T_3 , and subsequently allowing T_3 and $head(T_3)$ to be accepted. That would make the conditional priority $\prec_K^{\mathcal{P}}$ applicable, which blocks the attack of T_1 on T_2 .

Cycles of Self-Supporting Arguments. Again, we consider an example with a consistent set of priority rules \mathcal{P} , a context condition $K \in \mathcal{L}$, and two arguments T_1 and T_2 so that (i) T_1 undercuts T_2 at $\delta \in Asmptn(T_2)$, (ii) $Asmptn(T_1) \prec_K^{\mathcal{P}} \delta$, and (iii) K is not derivable from the current beliefs, unless $head(T_2)$ is added to the current beliefs.

The dilemma here is that, without $head(T_2)$ the conditional priority $\prec_K^{\mathcal{P}}$ is not applicable, allowing T_1 to defeat T_2 . On the other hand, by allowing T_2 and $head(T_2)$ to be accepted, we can also block the attack of T_1 on T_2 , resulting in a consistent extension containing both T_1 and T_2 .

REMARK. The problem discussed here is not unique to our approach. In fact, it could be shown that this problem is inherent to any approach that is based on conditional priorities, e.g., Dung and Son's [10] argument-based approach to reasoning with specificity.

4.3 Stratified Argumentation

We have also examined various cases of fallacious argumentation in which argumentation frameworks with conditional priorities could result in undesirable reasoning outcomes. To prevent such fallacious reasoning, we will look for some well-formed classes of argument systems for which acceptability of arguments is well defined.

The following definition extends the notion of stratification for logic programs [4,13,3] to set of arguments with conditional priorities. Note that we write $attack^*(\Theta, \theta)$ when the set of arguments Θ attacks the argument θ and no strict subset of Θ attacks θ .

Definition 10. Let a set of arguments Θ be such that Θ contains the subarguments of every argument in Θ (i.e., $\Theta = \bigcup_{\theta \in \Theta} sub(\theta)$).

1. A **stratification** for Θ is a function ρ from Θ to the countable ordinals.
2. An argument $\theta \in \Theta$, whose subarguments are $\theta_1, \dots, \theta_k$, ($k \geq 1$) is **stratified wrt. ρ** if

- (i) $\rho(\theta) \geq \rho(\theta_i)$, for $i = 1 \dots k$; and
 - (ii) $\forall \Psi \subseteq \Theta. \text{attack}^*(\Psi, \theta) \rightarrow \rho(\theta) > \max_{\psi \in \Psi} \{\rho(\psi)\}$.
3. Θ is **stratified** wrt. ρ if all of its arguments are stratified wrt. ρ . And, Θ is stratified if it is stratified wrt. some stratification.

Proposition 3. Let Θ be a set of arguments, if Θ is stratified then it has a unique stable extension.

Essentially, being stratified prevents a set of arguments from having any cycle of attack. Nevertheless, the above definition and property are only applicable to non-prioritised sets of arguments. As priority relations can help break cycles of attack, they can also help turn non-stratified sets of arguments to stratified ones. Observe that when there is an unconditional priority relation \prec between arguments we can simply replace the relation attack^* in the above definition by the relation \prec - attack^* , and a notion of priority-based stratification can be straightforwardly produced. Furthermore, \prec - $\text{attack}^* \subseteq \text{attack}^*$ as the priority relation may block some attacks between arguments due to the attacked arguments being more prioritised than their respective attackers. By allowing the priority relation to block attacks between arguments, we effectively provide a resolution for the first issue discussed above. That is, the priority relation takes precedence over the argumentation structures.

Nevertheless, it's getting more complicated when the priority relations are conditional. This is because of the second issue. That is, in addition to the cycles of attack between arguments, a stratified argumentation also needs to rule out cycles created by the interaction between arguments and the context conditions of priority rules.

- Definition 11.**
1. Let T be an argument and $K \in \mathcal{L}$ a context condition. T **contributes in a derivation of K** if there exists $K' \in \mathcal{L}$ s.t. $K' \not\vdash K$ and $K' \cup \{\text{head}(T)\} \vdash K$.
 2. Argument T_1 **blocks an attack** by argument T_2 , denoted by $\text{block}(T_1, T_2)$, if either (i) T_2 rebuts an argument T_3 and $\text{Asmptn}(T_2) \prec_K^P \text{Asmptn}(T_3)$ and T_1 contributes in a derivation of K ; or (ii) T_2 undercuts an argument T_3 at δ and $\text{Asmptn}(T_2) \prec_K^P \{\delta\}$ and T_1 contributes in a derivation of K .¹
 3. Let a consistent set of priority rules \mathcal{P} be given, argument T_1 **unconditionally attacks** argument T_2 wrt. \mathcal{P} , denoted by $\text{attack}_{\mathcal{P}}(T_1, T_2)$, if either $T_1 \prec_{\top}^P$ -rebuts T_2 , or $T_1 \prec_{\top}^P$ -undercuts T_2 at $\delta \in \text{Asmptn}(T_2)$, where $\top \equiv \neg A \vee A$.

Definition 12. Let \mathcal{P} be a consistent set of priority rules. Assume that Θ is a set of arguments Θ that contains the subarguments of every argument in Θ (i.e., $\Theta = \bigcup_{\theta \in \Theta} \text{sub}(\theta)$).

1. A **\mathcal{P} -stratification** for Θ is a function ρ from Θ to the countable ordinals.
2. An argument $\theta \in \Theta$, whose subarguments are $\theta_1, \dots, \theta_k$, ($k \geq 1$) is \mathcal{P} -stratified wrt. ρ if
 - (i) $\rho(\theta) \geq \rho(\theta_i)$, for $i = 1 \dots k$;
 - (ii) $\forall \theta' \in \Theta. \text{attack}_{\mathcal{P}}(\theta', \theta) \rightarrow \rho(\theta) > \rho(\theta')$; and
 - (iii) $\forall \theta' \in \Theta. \text{block}(\theta', \theta) \rightarrow \rho(\theta) > \rho(\theta')$.

¹ Note that it's perfectly legitimate for T_3 and T_1 to be the same argument.

3. Θ is **\mathcal{P} -stratified** wrt. ρ if all of its arguments are \mathcal{P} -stratified wrt. ρ . And, Θ is \mathcal{P} -stratified if it is \mathcal{P} -stratified wrt. some \mathcal{P} -stratification.

Proposition 4. Let \mathcal{P} be a consistent set of priority rules and Θ a set of arguments, if Θ is \mathcal{P} -stratified then it has a unique stable extension.

5 Related Work

The work that is most relevant to our approach presented in this paper is by Prakken and Sartor [16]. In their approach, a priority relation between inference rules is introduced. Moreover, statements expressing this priority relation is part of the knowledge base. That allows their approach to reason about this priority relation. Prakken and Sartor's approach also allows dynamic priorities to be defined, enabling context-sensitive inferences. Antoniou [2] also introduces dynamic priorities to defeasible logic.

Unlike the approaches introduced by Prakken and Sartor and by Antoniou as well as many other priority-based approaches in the literature (e.g., [7,1]) in which the priority relation is defined between inference rules, in our approach the priority relation is defined between assumptions. This is important because in many situations, the applicability of a defeasible inference rule needs to be examined in the presence of other conflicting rules (see e.g., [17]). Furthermore, while most of the conditions in other approaches are defined by classes of logic, our conditions are defined on arguments.

Amgoud and Cayrol ([1]) also systematically study preference-based argumentation frameworks. However, their study is based on the assumption that a priority relation between defeasible beliefs of the knowledge base is in place to allow arguments (constructed from these beliefs and classical logics) to be compared. Furthermore, their priority relation is also context-insensitive.

6 Conclusion

In this paper we re-visit an argumentation-theoretic approach to defeasible reasoning. Based on the premises that (i) the defeasibility in an agent' beliefs and reasoning is due to the (context-specific) assumptions made by the agent to enrich her knowledge about the world; and (ii) argumentation frameworks capture the essence of practical reasoning in many application and problem domains as explored throughout the literature, we investigate the feasibility of assumption-based argumentation frameworks.

We introduced a common language for expressing assumptions and conditional priorities. We then came back to the notion of argument acceptability which is arguably the central notion of all argumentation-based approaches. We identified the major issues that could intrinsically impede the determination of whether an argument, or a set of arguments, should be accepted. We then introduced the conditions under which acceptability of arguments can be cleanly determined by inhibiting the above issues.

For future work, we will investigate how the proposed argumentation framework with conditional priorities can be employed to render not only an agent's beliefs, but also her intentions, goals, and plans. We would like to see how the additional constructs interact and the effects they will have on the agent's arguments as well as on the acceptability of arguments.

References

1. Amgoud, L., Cayrol, C.: Inferring from inconsistency in preference-based argumentation frameworks. *J. Autom. Reason.* 29(2), 125–169 (2002)
2. Antoniou, G.: Defeasible logic with dynamic priorities. *Int. J. Intell. Syst.* 19(5), 463–472 (2004)
3. Apt, K.R., Blair, H.A.: Arithmetic classification of perfect models of stratified programs. *Fundam. Inform.* 14(3), 339–343 (1991)
4. Apt, K.R., Blair, H.A., Walker, A.: Towards a theory of declarative knowledge. In: Minker, J. (ed.) *Foundations of deductive databases and logic programming*, pp. 89–148. Morgan Kaufmann Publishers Inc., San Francisco (1988)
5. Bench-Capon, T.J.M.: Persuasion in Practical Argument Using Value-based Argumentation Frameworks. *J Logic Computation* 13(3), 429–448 (2003)
6. Bondarenko, A., Dung, P.M., Kowalski, R.A., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence Journal* 93, 63–101 (1997)
7. Brewka, G.: Reasoning about priorities in default logic. In: AAAI, pp. 940–945 (1994)
8. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence Journal* 77, 321–357 (1995)
9. Dung, P.M., Mancarella, P., Toni, F.: Computing ideal sceptical argumentation. *Artif. Intell.* 171(10-15), 642–674 (2007)
10. Dung, P.M., Son, T.C.: An argument-based approach to reasoning with specificity. *Artif. Intell.* 133(1-2), 35–85 (2001)
11. Elvang-Göransson, M., Hunter, A.: Argumentative logics: Reasoning with classically inconsistent information. *Data Knowl. Eng.* 16(2), 125–145 (1995)
12. Elvang-Göransson, M., Krause, P., Fox, J.: Acceptability of arguments as ‘logical uncertainty’. In: ECSQARU, pp. 85–90 (1993)
13. Gelfond, M.: On stratified autoepistemic theories. In: AAAI, pp. 207–211 (1987)
14. Governatori, G., Maher, M.J., Antoniou, G., Billington, D.: Argumentation semantics for defeasible logic. *J. Log. Comput.* 14(5), 675–702 (2004)
15. Horty, J.F., Thomason, R.H., Touretzky, D.S.: A skeptical theory of inheritance in nonmonotonic semantic networks. *Artif. Intell.* 42(2-3), 311–348 (1990)
16. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics* 7, 25–75 (1997)
17. Vo, Q.B., Foo, N.Y.: Reasoning about action: An argumentation-theoretic approach. *Journal of Artificial Intelligence Research* 24, 465–518 (2005)
18. Vo, Q.B., Foo, N.Y., Thurbon, J.: Semantics for a theory of defeasible reasoning. *Annals of Mathematics and Artificial Intelligence* 44(1-2), 87–119 (2005)

Knowledge Supervised Text Classification with No Labeled Documents

Congle Zhang^{1,2}, Gui-Rong Xue^{1,2}, and Yong Yu

¹ Apex Lab, Shanghai Jiaotong University, Shanghai, 200240

² State Key Lab of CAD & CG, Zhejiang University, Hangzhou, 310058

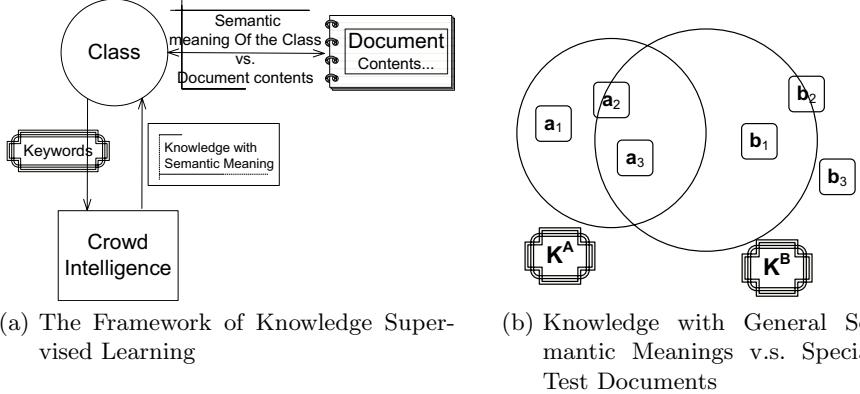
{zhangcongle,grxue,yyu}@apex.sjtu.edu.cn

Abstract. In traditional text classification approaches, the semantic meanings of the classes are described by the labeled documents. Since labeling documents is often time consuming and expensive, it is a promising idea that asking users to provide some keywords to depict the classes, instead of labeling any documents. However, short pieces of keywords may not contain enough information and therefore may lead to unreliable classifier. Fortunately, there are large amount of public data easily available in web directories, such as ODP, Wikipedia, etc. We are interested in exploring the enormous crowd intelligence contained in such public data to enhance text classification. In this paper, we propose a novel text classification framework called “*Knowledge Supervised Learning*” (KSL), which utilizes the knowledge in keywords and the crowd intelligence to learn the classifier without any labeled documents. We design a two-stage risk minimization (TSRM) approach for the KSL problem. It can optimize the expected prediction risk and build the high quality classifier. Empirical results verify our claim: our algorithm can achieve above 0.9 on Micro-F1 on average, which is much better than baselines and even comparable against SVM classifier supervised by labeled documents.

1 Introduction

Supervised learning [2,5] and semi-supervised learning [18,19] algorithms are widely used in text classification. They require high quality labeled documents for each class to build the classifier. Some studies [13] propose another strategy that users can provide representative keywords to depict the classes. Such alternative is very promising because labeling documents is often much more expensive than giving some keywords. For example, when internet service providers want to classify web pages according to users’ interests, it is much easier to ask them issue some words to describe their interest than to collect web pages they are really interested in.

However, one can hardly expect users to provide long and detailed keywords. It means that the information of the keywords are not enough to learn a reliable classifier. The gap between the keywords and the corresponding semantic meaning should be filled up. We notice that many web directories grows enormous and easily available, e.g. Open directory project (ODP), Wikipedia, Yahoo! Directory. Such public data sets contain huge amount of articles together

**Fig. 1.** Idea Illustration

with annotated category attributes. We are interested in exploring such *crowd intelligence* in large public data to enrich the short keywords, and obtain the knowledge that reveals the semantic meaning of the classes. Such knowledge can further supervise the text classification without any labeled documents. We call this learning framework *knowledge supervised learning*(KSL) shown in Fig. 1(a).

There exists a challenge in above framework: knowledge for each class may be general in semantic meanings and cover a mixture of topics because the corresponding keywords are short. Meanwhile, test documents are often special and focus on few topics. For example, knowledge for the class with keyword “computer” may contain thousands of topics, including software, hardware, companies and people, while test documents in “computer” class may be just about computer courses. Figure 1(b) is an illustration for a binary classification task. Knowledge K^A and K^B are that of the classes A and B . We use big circle for them because they are general and cover many topics. They also intersect because some topics may be shared. $\{a_i, b_i\}$ are test documents, which are represented by small squares because they are special and cover few topics. Some documents (e.g. a_1, b_1) are related to topics of a single class; some others (e.g. a_2, a_3), may be relevant to topics of both; others (e.g. b_2, b_3), may cover topics irrelevant to the knowledge of the keywords.

In this paper, we propose a two-stage risk minimization approach (TSRM) to deal with the above challenge of the KSL problem. TSRM incorporates the keywords and the crowd intelligence into the learning process under the risk minimization framework [11] to learn a high quality classifier. TSRM is based on two intuitions. Firstly, it is reasonable to predict a_1 (or b_1) to class A (or B) in Figure 1(b) because it is related to topics only relevant to K^A or K^B . Secondly, it is also reasonable for us to predict a_2, a_3 to class A and b_2, b_3 to class B , by considering the fact that documents in one class are mutually close to each other. Our TSRM algorithm contains two stages to fulfill these two intuitions: (1) the first stage in TSRM is to minimize the independent risk, i.e. the prediction risk of one test document is decided by taking into account the knowledge of the classes

and the test document itself, but is irrelevant to the predictions of the other test documents. (2) the second stage in TSRM is to update the independent risk into the dependent risk, by considering the relationship among the test documents.

Experiments on 20-Newsgroup and Reuters-21578 show our TSRM approach is effective and reliable. In order to demonstrate that TSRM deals well with short keywords, we just use class name as the keywords for each class. TSRM achieves above 0.9 on Micro-F1, which significantly outperform baselines. We have also conducted SVM with 10% labeled documents to prove that our algorithm has comparable performance against supervised learning, though we use no labeled documents.

2 Related Work

Studies [13,3] propose that giving words is much more convenient than labeling documents. Work in [3] propose active learning while work in [13] modifies naive Bayes to bootstrap one and make it possible to classify by words. Compared with them, we incorporate knowledge into learning process. In experiment, we have compared with bootstrap and shown that KSL achieves better and more reliable results than bootstrap, especially when class-description is short and general.

Several recent papers have modified learning approach to use extra information in text classification. Some work use a set of keywords for each task class as domain knowledge. Work in [1] combines key words with labeled documents in a Bayesian logistic regression model to improve classification. Learning algorithm SVM and logistic regression are modified in [15,22], which convert domain knowledge into weighted training examples. A study [10] proposes interactive approach to up-weight human selected terms in SVM learning. Out-task labeled examples in logistic regression can be used to train in-task examples[4]. Work in [9] applies a generative model with explicit aspect layer which are relevant topics to task classes. Non-textual uses of Bayesian priors is employed in [6]. These works show extra information can help labeled documents in learning tasks. Compared with them, our work focus on using knowledge to supervise learning process without any labeled documents.

Some papers explores relationship among features in out-task documents and then generate or select features to improve text classification task. Work in [14] uses transfer learning to build covariance matrix for feature. Gabrilovich[7] propose the way to use world knowledge in large web directory for feature generation and feature selection. This sort of techniques can be used to enhance text classification with labeled documents [8]. For our problem, since there is no labeled documents, it will introduce noise if we extend general keywords in enormous knowledge. We have conducted experiment comparing against this sort of strategy to show this fact.

3 Two-Stage Risk Minimization Approach for Knowledge Supervised Learning

In this section, We first formulate the knowledge supervised learning problem. Secondly, we model the KSL problem in a risk minimization framework. This

convert the problem to find a prediction having the least risk. Finally, we design the two-stage risk minimization algorithm (TSRM) that can predict the test documents with least risk by considering knowledge from the crowd intelligence, and mutual relationship among test documents as well.

3.1 Problem Formulation

Our learning task in this paper is to classify the test documents into the classes represented by the keywords, with the help of the crowd intelligence in the public data. We denote them as follows. We have keywords for the classes as $\mathcal{C} = \{\mathbf{c}_\alpha\}_{\alpha=1}^\ell$, where \mathbf{c}_α is the representative keywords for one class α . The test document set is $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$, where \mathbf{x}_i is a document in vector form with words belonging to word set $\mathcal{W} = \{w_j\}_{j=1}^m$. For the crowd intelligence, we denote them as tuple set $\mathcal{K} = \{\kappa_k = (\mathbf{d}_k, t_k)\}$, where \mathbf{d}_k is an article and t_k is the topic with respect to \mathbf{d}_k . We can formulate crowd intelligence in this way because documents in public data always annotated with tags or categories, which can be viewed as the topic related to that document. The output of our learning task is the prediction $\mathbf{x} \rightarrow y$. We denote $\mathcal{Y} = \{y_i\}_{i=1}^n$, where $y_i \in \{1, 2, \dots, \ell\}$ is a class.

3.2 Modeling KSL under the Risk Minimization Framework

In this subsection we model the knowledge supervised learning process under the risk minimization framework, incorporating keywords, test documents and crowd intelligence as well. This risk minimization framework is based on Bayesian decision theory, which can further derive our TSRM algorithm.

We notice there are three roles in our learning process: users \mathcal{U} who provide the keywords; document source \mathcal{S} that provides the test documents; public data \mathcal{P} that provides the crowd intelligence. We assume that $\mathcal{U}, \mathcal{S}, \mathcal{P}$ generates $\mathcal{C}, \mathcal{X}, \mathcal{K}$ with respect to the Markov chains. For keywords, \mathcal{U} generates \mathbf{c}_α according to $\mathcal{U} \rightarrow \theta_{\mathcal{C}}^\alpha \rightarrow \mathbf{c}_\alpha$. That is, \mathcal{U} selects a model with parameter $\theta_{\mathcal{C}}^\alpha$, according to distribution $p(\theta_{\mathcal{C}}^\alpha | \mathcal{U})$. Then, using this model, the keywords \mathbf{c}_α is generated by probability $p(\mathbf{c}_\alpha | \theta_{\mathcal{C}}^\alpha)$. In the same way, we have $\mathcal{S} \rightarrow \theta_{\mathcal{X}}^i \rightarrow \mathbf{x}_i$ with respect to $p(\theta_{\mathcal{X}}^i | \mathcal{S})$ and $p(\mathbf{x}_i | \theta_{\mathcal{X}}^i)$; $\mathcal{P} \rightarrow \theta_{\mathcal{K}}^k \rightarrow \kappa_k$ with respect to $p(\theta_{\mathcal{K}}^k | \mathcal{P})$ and $p(\kappa_k | \theta_{\mathcal{K}}^k)$. This process is similar to language model for document retrieval [11].

We use a hidden binary variable $B_{i,\alpha}$ to model whether the test document \mathbf{x}_i belong to the class α represented by the keywords \mathbf{c}_α or not. Its value follows the probability $p(B_{i,\alpha} | \theta_{\mathcal{X}}^i, \theta_{\mathcal{C}}^\alpha, \theta_{\mathcal{K}})$, where $\theta_{\mathcal{K}} = \{\theta_{\mathcal{K}}^k | \kappa_k \in \mathcal{K}\}$.

With the modeling process above, we can optimize the prediction under risk minimization framework. Each prediction \mathcal{Y} is associated with a *loss function*, whose value is related to outside parameters, i.e. $\theta = \{\theta_{\mathcal{K}}, \{\theta_{\mathcal{X}}^i\}_{i=1}^n, \{\theta_{\mathcal{C}}^\alpha\}_{\alpha=1}^\ell, \{B_{i,\alpha}\}_{i,\alpha}\}$. We denote the loss as $L(\mathcal{Y}, \theta)$. With Bayesian decision theory, the expect risk for prediction \mathcal{Y} is given by

$$R(\mathcal{Y} | \mathcal{U}, \mathcal{C}, \mathcal{S}, \mathcal{X}, \mathcal{K}) = \int_{\Theta} L(\mathcal{Y}, \theta) p(\theta | \mathcal{U}, \mathcal{C}, \mathcal{S}, \mathcal{X}, \mathcal{K}) d\theta \quad (1)$$

where the posterior distribution can be written as:

$$p(\theta|\mathcal{U}, \mathcal{C}, \mathcal{S}, \mathcal{X}, \mathcal{K}) \propto p(\theta_{\mathcal{K}}|\mathcal{K}) \prod_{i,\alpha} p(B_{i,\alpha}|\theta_{\mathcal{X}}^i, \theta_{\mathcal{C}}^\alpha, \theta_{\mathcal{K}}) p(\theta_{\mathcal{X}}^i|\mathbf{x}_i, \mathcal{S}) p(\theta_{\mathcal{C}}^\alpha|\mathbf{c}_\alpha, \mathcal{U})$$

The rest of the paper will focus on designing two-stage algorithm to predict \mathcal{Y} with least expected risk for $R(\mathcal{Y}|\mathcal{U}, \mathcal{C}, \mathcal{S}, \mathcal{X}, \mathcal{K})$.

First Stage: Minimizing Independent Risk. Now we consider the risk to classify one single document with document independence assumption, which indicates the risk to predict y_i for \mathbf{x}_i is merely dependent on θ discussed above, and independent to other predictions $y_{i'}$, where $i' \neq i$. Thus, the risk $R(\mathcal{Y}|\mathcal{U}, \mathcal{C}, \mathcal{S}, \mathcal{X}, \mathcal{K})$ can be written as $R = \sum_i R^I(y_i, \mathbf{x}_i)$. By some standard calculation, we derive:

$$R^I(y_i, \mathbf{x}_i) = \sum_{\alpha=1}^{\ell} \sum_{B_{i,\alpha}=0}^1 \int_{\Theta_{\mathcal{X}}} \int_{\Theta_{\mathcal{C}}} \int_{\Theta_{\mathcal{K}}} L(y_i = \alpha | B_{i,\alpha}, \theta_{\mathcal{K}}, \theta_{\mathcal{X}}^i, \theta_{\mathcal{C}}^\alpha) \times \\ p(\theta_{\mathcal{K}}|\mathcal{K}) p(B_{i,\alpha}|\theta_{\mathcal{X}}^i, \theta_{\mathcal{C}}^\alpha, \theta_{\mathcal{K}}) p(\theta_{\mathcal{C}}^\alpha|\mathbf{c}_\alpha, \mathcal{S}) p(\theta_{\mathcal{X}}^i|\mathbf{x}_i, \mathcal{S}) d\theta_{\mathcal{X}} d\theta_{\mathcal{K}} d\theta_{\mathcal{C}} \quad (2)$$

In Equation (2), $L(y_i = \alpha|\theta)$ means with decision $y_i = \alpha$ and environment condition θ , the loss attached to classify \mathbf{x}_i to class α . We can combine the items in left side of Equation (2) related to \mathcal{K} and define L^K as:

$$L^K(y_i = \alpha | \theta_{\mathcal{X}}^i, \theta_{\mathcal{C}}^\alpha, \theta_{\mathcal{K}}) \triangleq \sum_{B_{i,\alpha} \in \{0,1\}} \int_{\Theta_{\mathcal{K}}} L(y_i | B_{i,\alpha}, \theta_{\mathcal{K}}, \theta_{\mathcal{X}}^i, \theta_{\mathcal{C}}^\alpha) p(B_{i,\alpha} | \theta_{\mathcal{X}}^i, \theta_{\mathcal{C}}^\alpha, \theta_{\mathcal{K}}) d\theta_{\mathcal{K}} \quad (3)$$

Notably, $B_{i,\alpha}$ completely depends on $\theta_{\mathcal{X}}^i$, $\theta_{\mathcal{C}}^\alpha$ and $\theta_{\mathcal{K}}$ because \mathbf{x}_i , \mathbf{c}_α and \mathcal{K} supervises the learning and determines whether \mathbf{x}_i should be classify to class α . Thus, we have

$$L^K(y_i = \alpha | \theta_{\mathcal{X}}^i, \theta_{\mathcal{C}}^\alpha, \theta_{\mathcal{K}}) \propto h(y_i, \alpha) \sum_{\kappa} L(y_i = \alpha | \kappa, \theta_{\mathcal{X}}^i, \theta_{\mathcal{C}}^\alpha) \quad (4)$$

$$\text{where } h(y_i, \alpha) = \begin{cases} 1 & \text{if } y_i = \alpha \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Up to now, $L(y_i|\theta)$ is unspecified for generality. Next, we will take advantage of knowledge \mathcal{K} to specify a effective loss function. If \mathbf{x}_i is about one topic t while keywords for \mathbf{c}_α is also very relevant to that topic, we can reduce the risk if we predict \mathbf{x}_i as class α , even when \mathbf{x}_i seems to be unrelated to \mathbf{c}_α . We can map both keywords and test documents into topic space of crowd intelligence and then judge their relationship. Such mapping largely alleviate the sparseness problem of keywords since topics in \mathcal{K} covers a huge range. To formulate this idea, we calculate L^K as

$$L^K(y_i = \alpha | \theta_{\mathcal{X}}^i, \theta_{\mathcal{C}}^\alpha, \theta_{\mathcal{K}}) = h(y_i, \alpha) \sum_{\kappa} L(y_i = \alpha | \kappa, \theta_{\mathcal{X}}^i, \theta_{\mathcal{C}}^\alpha) \quad (6)$$

$$= h(y_i, \alpha) \sum_{\kappa} p(\kappa | \theta_{\mathcal{X}}^i) \log \frac{p(\kappa | \theta_{\mathcal{X}}^i)}{p(\kappa | \theta_{\mathcal{C}}^\alpha)} \quad (7)$$

In Equation (7), the loss to predict \mathbf{x}_i as class α is evaluated in the Kullback-Leibler (KL) divergence between their distributions on topic space.

With $L^{\mathcal{K}}$ in Equation (7), one can rewrite Equation (2) as:

$$R^I(y_i, \mathbf{x}_i) = \sum_{\alpha=1}^{\ell} \int_{\Theta_{\mathcal{X}}} \int_{\Theta_{\mathcal{C}}} L^{\mathcal{K}}(y_i | \theta_{\mathcal{X}}^i, \theta_{\mathcal{C}}^{\alpha}, \theta_{\mathcal{K}}) p(\theta_{\mathcal{C}}^{\alpha} | \mathbf{c}_{\alpha}, \mathcal{S}) p(\theta_{\mathcal{X}}^i | \mathbf{x}_i, \mathcal{S}) d\theta_{\mathcal{X}} d\theta_{\mathcal{C}} \quad (8)$$

Equation (8) means the risk minimization criterion is equivalent to the sum of expected loss value. Since explicitly computing the risk is difficult, like [11], we approximate $R^I(y_i, \mathbf{x}_i)$ at the posterior mode:

$$R^I(y_i, \mathbf{x}_i) \propto \sum_{\alpha=1}^{\ell} L^{\mathcal{K}}(y_i | \hat{\theta}_{\mathcal{X}}^i, \hat{\theta}_{\mathcal{C}}^{\alpha}, \theta_{\mathcal{K}}) \quad (9)$$

where $\hat{\theta}_{\mathcal{X}}^i$ and $\hat{\theta}_{\mathcal{C}}^{\alpha}$ are parameters in expected value. Replace $L^{\mathcal{K}}(y_i | \cdot)$ with the right side of Equation (7), we obtain the independent risk for predicting one document:

$$R^I(y_i, \mathbf{x}_i) \propto \sum_{\alpha=1}^{\ell} h(y_i, \alpha) \sum_{\kappa} p(\kappa | \hat{\theta}_{\mathcal{X}}^i) \log \frac{p(\kappa | \hat{\theta}_{\mathcal{X}}^i)}{p(\kappa | \hat{\theta}_{\mathcal{C}}^{\alpha})} \quad (10)$$

Let us see how independent risk in Equation (10) works and what its limitation is. Back to Figure 1(b), we can verify that for the test document \mathbf{a}_1 , $R^I(y_{a_1} = A, \mathbf{a}_1)$ is sure to have less value than $R^I(y_{a_1} = B, \mathbf{a}_1)$. It is because \mathbf{a}_1 mostly relate to topics of class A as shown in the figure, which leads to the result that $\sum_{\kappa} p(\kappa | \hat{\theta}_{\mathcal{X}}^i) \log \frac{p(\kappa | \hat{\theta}_{\mathcal{X}}^i)}{p(\kappa | \hat{\theta}_{\mathcal{C}}^{\alpha})}$ varies big for $\alpha = A$ and $\alpha = B$, and further leads to the difference in independent risk. That is, independent risk is good to predict the test documents that are distinguishable in topic space. However, for documents like \mathbf{a}_3 or \mathbf{b}_3 , $\sum_{\kappa} p(\kappa | \hat{\theta}_{\mathcal{X}}^i) \log \frac{p(\kappa | \hat{\theta}_{\mathcal{X}}^i)}{p(\kappa | \hat{\theta}_{\mathcal{C}}^{\alpha})}$ varies little for $\alpha = A$ and $\alpha = B$, which makes independent risk not enough to predict them. That is to say, it is necessary to further utilize the structure in test documents to overcome the limitation of the independent risk.

Second Stage: Minimizing Dependent Risk. Dependent risk in this section cancels the independent assumption before. The key intuition is, since documents in one classes are always mutually close to each other, prediction risk for one document should be influenced by others. We explore such dependence by utilizing the word-document relationship among the test documents. We apply a probabilistic approach to incorporate the observed document-word co-occurrence data $\mathcal{X} \times \mathcal{W}$ into risk minimization framework. For document $\mathbf{x}_i \in \mathcal{X}$ and word $w_j \in \mathcal{W}$, they are attached class-specific distribution for class α . We denote them as $p_{i|\alpha} \triangleq p(x_i | \alpha)$ and $q_{j|\alpha} \triangleq p(w_j | \alpha)$. Meanwhile, each class is attached with a prior distribution $\pi_{\alpha} \triangleq p(\alpha)$. We can assume \mathbf{x}_i and w_j are conditional independent given α . Formally, $p(x_i, w_j | \alpha) = p(x_i | \alpha) \times p(w_j | \alpha)$. The joint probability

distribution for the co-occurrence is a mixture of separable conditional distributions: $p(x_i, w_j) = \sum_{\alpha=1}^{\ell} p(\alpha)p(x_i, w_j|\alpha) = \sum_{\alpha=1}^{\ell} \pi_{\alpha} p_{i|\alpha} q_{j|\alpha}$. Our purpose is to identify $p_{i|\alpha}$ that can further reveals the predictions for the test documents.

As discussed in the last sections, independent risk brings some reliable predictions. We want to propagate likelihood from such reliable predictions to other predictions with the probability distributions defined above. For example, we can identify the relationship between \mathbf{a}_3 and $\alpha = A$ by $p(\mathbf{a}_3|\alpha = A)$ referring $p(\mathbf{a}_1|\alpha = A)$ in Figure 1(b). To this end, the probabilities π_{α} , $p_{i|\alpha}$ and $q_{j|\alpha}$ can be decided by considering predictions obtained by independent risk. Meanwhile, we can also utilize representative keywords given by users. Concretely, (1)we set $p(\mathbf{c}_{\alpha}|w_j) = 1$ when word w_j occur in class-description \mathbf{c}_{α} , since this word is the representative word for that class identified by user. (2)we set $p(\mathbf{c}_{\alpha}|\mathbf{x}_j) = 1$ when prediction $y_j = \alpha$ for \mathbf{x}_j has a very low independent risk in Equation 10. It is reasonable to set in this way because low independent risk means the test document only relate to topics of a single class.

We define $V = \{w_j | \exists \alpha, s.t. w_j \in c_{\alpha}\}$ as the set of all the words that appear in keywords. We denote the projection between $w_j \in V$ and \mathbf{c}_{α} as $g : w_j \rightarrow \alpha$. Similarly, we use set U to denote those documents having predictions with low independent risk in Equation (10). These prediction can be defined as the projection $f : x_i \rightarrow \alpha$.

Based on above discussion, we can identify the probability distributions by maximizing log-likelihood \mathcal{L} fitting word-document co-occurrence, keyword and independent prediction risk as:

$$\begin{aligned} \mathcal{L} = & \sum_{i,j} n_{ij} \log p(\mathbf{x}_i, w_j) = \sum_{x_i \in U} \sum_{w_j \in V} n_{ij} \log \left(\sum_{\alpha} \pi_{\alpha} p_{i|\alpha} q_{j|\alpha} \right) \\ & + \sum_{x_i \in U} \sum_{w_j \in V} n_{ij} \log (q_{j|f(x_i)}) + \sum_{x_i \in U} \sum_{w_j \in V} n_{ij} \log (p_{i|g(w_j)}) \end{aligned} \quad (11)$$

where n_{ij} indicates the times that word w_j occur in document \mathbf{x}_i . Equation (11) comes from following facts, derived easily from the definitions of U, V :

$$\forall w_j \in V: q_{j|\alpha} = \frac{p(\mathbf{c}_{\alpha}|w_j)p(w_j)}{p(\alpha)} \propto \begin{cases} 1/\pi_{\alpha} & \text{if } \alpha = g(w_j) \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

$$\forall \mathbf{x}_i \in U: p_{i|\alpha} = \frac{p(\mathbf{c}_{\alpha}|\mathbf{x}_i)p(\mathbf{x}_i)}{p(\alpha)} \propto \begin{cases} 1/\pi_{\alpha} & \text{if } \alpha = f(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

The EM algorithm is known to increase the likelihood in every step. We introduce the posterior probabilities $\phi_{\alpha|i,j} = p(\alpha|x_i, w_j)$, which is calculated in E-step based on current estimates of parameters through Bayesian rule:

$$\phi_{\alpha|i,j} = \frac{\pi_{\alpha} p_{i|\alpha} q_{j|\alpha}}{\sum_{\alpha'} \pi_{\alpha'} p_{i|\alpha'} q_{j|\alpha'}} \quad (14)$$

In M-step, $p_{i|\alpha}$, $q_{j|\alpha}$ and π_{α} are derived by:

$$p_{i|\alpha} \propto \sum_j n_{ij} \phi_{\alpha|i,j} \quad ; \quad q_{j|\alpha} \propto \sum_i n_{ij} \phi_{\alpha|i,j} \quad (15)$$

$$\pi_\alpha \propto \sum_i \sum_j n_{ij} \phi_{\alpha|i,j} \quad (16)$$

Back to risk minimization framework, the loss should be high if we classify \mathbf{x}_i to class α when $p(\mathbf{c}_\alpha | \mathbf{x}_i) \propto p_{i|\alpha} \pi_\alpha$ is high. With the distribution calculated in equation 15, we can rewrite the loss function as: $L(y_i | \cdot) = 1/\pi_{y_i} p_{i|y_i}$. Following the same way we yield equation 10, we can updated independent risk to dependent risk $R^D(y_i = \alpha, \mathbf{x}_i)$:

$$R^D(y_i = \alpha, \mathbf{x}_i) \propto \sum_{j=1}^{\ell} h(y_i, \alpha) p_{i|\alpha} \quad (17)$$

Let us see how Equation (17) overcomes the limitation of the independent risk. Test documents in the same class share more common words and therefore conditional probabilities of that class are propagated among them more strongly. For test documents that cannot be distinguished by independent risk, e.g. \mathbf{a}_3 or \mathbf{b}_3 in Figure 10, dependent risk can separate them apart with the help of $p_{i|\alpha}$ decided by the entire test document set.

4 Implementation Details

In order to estimate $p(\kappa | \hat{\theta}_{\mathcal{X}}^i)$, $p(\kappa | \hat{\theta}_{\mathcal{C}}^\alpha)$ in Equation (10), we view $\hat{\theta}_{\mathcal{X}}^i$ and $\hat{\theta}_{\mathcal{C}}^\alpha$ as empirical distribution of document $x_i = w_i^1 w_i^2 \dots w_i^{|x_i|}$ and $c_\alpha = w_\alpha^1 w_\alpha^2 \dots w_\alpha^{|c_\alpha|}$. For practical implementation, we count the number of word w occur in topic attached document set, sum the number and normalize it to estimate the value. Since there are huge amount of knowledge topics and it is time consuming to enumerate them all, we select some κ to use in Equation (10). For each class α , we prefer topics that are able to distinguish it from other classes. We choose the knowledge topics κ that differ greatly in $p(\kappa | \hat{\theta}_{\mathcal{C}}^\alpha)$ to other $\alpha' \neq \alpha$. We balanced select reliable prediction based on independent risk for each class and add constraints according to these predictions. Adopting EM-algorithm, after coverage, TSRM makes prediction for \mathcal{Y} as $\mathcal{Y} = \{y_i | y_i = \arg \min_\alpha R^D(y_i = \alpha, \mathbf{x}_i)\}$.

5 Experiments

5.1 Experiment Design

We have evaluated the performance of our TSRM approach using two data corpus: the Reuters-21578 and the 20 Newsgroups document corpora. These two data corpus have been widely used for classification experiments. For Reuters, we discard documents with multiple class labels and removed the classes with less than 60 documents. This lead to 14 classes with 8526 documents. The Newsgroup corpora contains about 20,000 documents that were collected from 20 news groups in the public domains. We focus on binary classification in this paper, which means we can generate $190(C_{20}^2)$ datasets for 20Ng and 91 datasets(C_{14}^2) for Reuters.

We use “Open Directory Project” as the crowd intelligence. We crawl web pages from ODP and remove html tags. They are articles \mathbf{b} of \mathcal{K} . We use the leaf categories in ODP category tree to serve as topics t of \mathcal{K} . In this paper’s experiment, \mathcal{K} contains 1.3 million articles and 156 thousand categories. In order to prove our TSRM approach can deal well with short, general keywords, we just use the class names as the representative keywords for the classes whose class names are not abbreviation. For other abbreviation cases, we extend it to readable phrase in less than five words.

To study the performance of our proposed knowledge supervised learning algorithm, we compare the performance of TSRM against that of three strategies. (1)Bootstrap: follow [13], use keywords as the seed information, extend it in task documents and execute Naive Bayes EM to train a model to classify documents; (2)train from relevant knowledge articles (TRKA): find articles most relevant to keywords in knowledge \mathcal{K} by cosine similarity, and use these articles as labeled set to train the classifier; (3)supervised learning (SL): we randomly sample a portion of documents from \mathcal{X} as the labeled documents and train a supervised learning model. Notable, the third strategy SL utilizes labeled documents, which is not available in TSRM settings. We conduct this comparison to prove that TSRM can reach the comparable performance against supervised learning approach with labeled documents, though TSRM needs none of them. We use SVM-Light [17] with carefully tuned parameters in strategy 2 and 3 to build the classifier.

We employ Micro-F1 defined as $\frac{2pr}{p+r}$ as performance metrics, where p and r are precision and recall respectively. Besides overall performance comparison against other strategies, we conducts a series of experiments to investigate our approach. We check the precision-recall curve by predicting merely on independent risk, which verify the necessarily and importance of dependent risk. We also evaluate how parameters effect the final performance.

5.2 Experiment Results

Overall Performance. We have tested the TSRM approach for both the Reuters and the Newsgroup document corpora on all datasets of the classes. For limited space, we cannot list all 190 datasets for 20Ng and 91 datasets for Reuters. We choose 7 and 10 datasets for 20ng/Reuters respectively, following the rule that every class is in one dataset. We also give the average case (average on 190/91 datasets).

For each class, top 200 relevant articles in \mathcal{K} are extracted to train the classifier for TRKA; we choose top twenty percent documents with smallest independent risk for U in Equation (11). 10% documents are sampled as training samples for strategy SL.

Table 1 is the performance comparison on Reuters and 20Ng. TRKA arrives at a poor performance, because many articles relevant to keywords are noise for the test documents. Bootstrap is very unreliable because representative keywords are short and very general. It can obtain good performance in some cases but may totally fail in others. Results of the baselines show that it is necessary to design some delicate algorithm for our problem. TSRM approach greatly outperforms

Table 1. Performance Comparison by 4 strategies on Reuters and 20ng

Class Pair	TRKA	Bootstrap	TSRM	SL
Average for 190 pairs in 20ng	0.543	0.841	0.919	0.936
Average for 91 pairs in Reuters	0.498	0.711	0.906	0.924
alt.atheism / talk.politics.mideast	0.595	0.954	0.911	0.961
comp.sys.ibm.pc.hardware / rec.autos	0.422	0.950	0.926	0.935
comp.graphics / misc.forsale	0.516	0.518	0.882	0.957
comp.os.ms-windows / comp.sys.mac.hardware	0.672	0.477	0.851	0.890
comp.windows.x / rec.sport.baseball	0.622	0.973	0.980	0.984
rec.motorcycles / sci.space	0.385	0.785	0.974	0.964
rec.sport.hockey / soc.religion.christian	0.559	0.993	0.981	0.954
sci.electronics / sci.med	0.261	0.673	0.959	0.958
talk.politics.guns / talk.religion.misc	0.405	0.432	0.920	0.931
talk.politics.misc / talk.religion.misc	0.332	0.609	0.895	0.939
earn / acq	0.546	0.702	0.888	0.943
crude / trade	0.362	0.608	0.943	0.887
money-fx / interest	0.377	0.571	0.834	0.769
ship / money-supply	0.883	0.917	0.987	0.986
sugar / coffee	0.767	0.530	0.972	0.993
gold / gnp	0.650	0.956	0.862	0.912
cpi / cocoa	0.031	0.514	0.934	0.981

TRKA and Bootstrap. Meanwhile, its performance is comparable against supervised learning methods SL with labeled documents. For some classes, TSRM performs even better. TSRM is also reliable, because it effectively exploits the valuable knowledge to supervise the learning process.

Performance of Independent Risk. In this part, we conduct experiments to show the performance with independent risk only. In Figure 2(a), we make prediction merely by independent risk in Equation (10), i.e. $y_i = \arg \min_{\alpha} R^I(\alpha, \mathbf{x}_i)$. For three datasets and the average case on 20NG, we change the recall value and record how precision changes. It can be seen that precision is satisfactory with low recall. That is to say, it is reliable to predict those documents whose independent risk is small. Figure 2(b) show average P/R curves for prediction on 20ng by independent risk and dependent risk respectively. From this figure, dependent risk outperforms independent risk greatly because it effectively improves recall. These experiments verify our claim that (1)minimizing independent risk can lead to high precision on test documents with small R^I value. (2)it is necessary to consider dependency among test documents to refine the predictions.

Parameters. There are two parameters in our algorithm. One is the number of topics we used in equation 10, i.e. size of κ set. Another is the number of documents we adopt independent risk to make prediction, i.e. size of set U . Left side of figure 3 show how Micro-F1 changes with $|\kappa|$. We see big $|\kappa|$ helps the performance generally, which means TSRM successfully exploits the crowd intelligence. Right side show how performance changes with $|U|$. As expected, performance

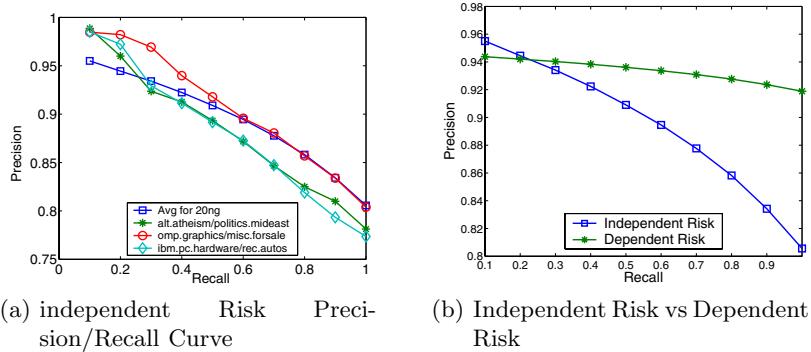


Fig. 2. Performance of Independent Risk

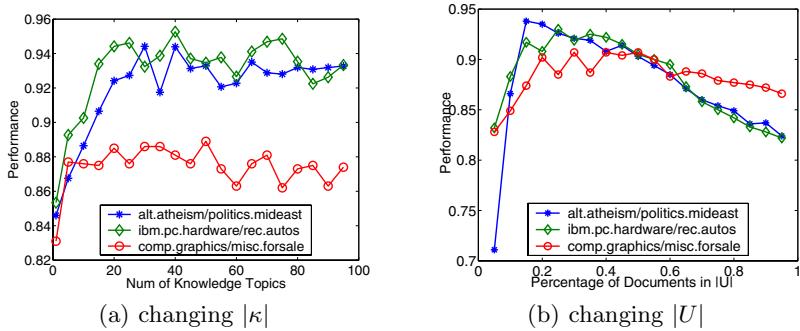


Fig. 3. Performance when changing parameters

increases at first and decrease later. The reason is that some documents should be predicted by minimizing dependent risk rather than independent one along.

6 Conclusion

This paper introduce a novel text classification framework named Knowledge Supervised Learning, which utilize the representative keywords for the classes and the crowd intelligence to learn the high quality classifier without any labeled documents. We propose a two-stage risk minimization algorithm to effectively solve the KSL problem. The experiments show that TSRM can greatly improve the baselines and achieve 0.91 in Reuters and 0.92 in 20 Newsgroup on Micro-F1. The performance of TSRM is also comparable to that of supervised learning techniques like SVM with labeled documents.

Acknowledgements

Supported by the Open Project Program of the State Key Lab of CAD&CG Grant No., Zhejiang University.

References

1. Dayanik, A., Lewis, D.: Constructing informative prior distributions from domain knowledge in text classification. In: SIGIR 2006, pp. 493–500 (1995)
2. Genkin, A., Lewis, D., Madigan, D.: Large-scale bayesian logistic regression for text categorization. Technical report, DIMACS (2004)
3. Liu, B., Li, X., Lee, W.S.: Text Classification by Labeling Words. In: AAAI 2004, pp. 425–430 (2004)
4. Chelba, C., Acero, A.: Adaptation of maximum entropy capitalizer: Little data can help a lot. In: EMNLP 2004 (2004)
5. Lewis, D., Gale, W.: A sequential algorithm for training text classifiers. In: SIGIR 1994 (1994)
6. Madigan, D., Gavrin, J., Raftery, A.: Eliciting prior information to enhance the predictive performance of bayesian graphical models. Communications in Statistics-Theory and Methods, pp. 2271–2292 (1995)
7. Gabrilovich, E., Markovitch, S.: Feature Generation for Text Categorization Using World Knowledge. In: IJCAI 2005 (2005)
8. Gabrilovich, E., Markovitch, S.: Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. In: AAAI 2006 (2006)
9. Ifrim, G., Weikum, G.: Transductive Learning for Text Classification Using Explicit Knowledge Models. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 223–234. Springer, Heidelberg (2006)
10. Raghavan, H., Madani, O., Jones, R.: Interactive feature selection. In: IJCAI 2005, pp. 841–846 (2005)
11. Lafferty, J., Zhai, C.: Document language models, query models, and risk minimization for information retrieval. In: Proceedings of SIGIR 2001 (2001)
12. Nigam, K., Ghani, R.: Analyzing the Effectiveness and Applicability of Co-training. In: CIKM 2000, pp. 86–93 (2000)
13. Jones, R., McCallum, A., Nigam, K., Riloff, E.: Bootstrapping for text learning tasks. In: IJCAI 1999 Workshop on Text Mining (1999)
14. Raina, R., Ng, A.Y., Koller, D.: Constructing informative priors using transfer learning. In: ICML 2006, pp. 713–720 (2006)
15. Schapire, R., Rochery, M., Rahim, M., Gupta, N.: Incorporating prior knowledge into boosting. In: ICML 2002 (2002)
16. Hofmann, T., Puzicha, J.: Statistical Models for Co-occurrence Data. Technical Report 1999 (1999)
17. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Nédellec, C., Rouveiro, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
18. Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines. In: International Conference on Machine Learning, ICML 1999 (1999)
19. T. Joachims, Transductive Learning via Spectral Graph Partitioning. In: Proceedings of the International Conference on Machine Learning (ICML) (2003)
20. Mitchell, T.: The role of unlabeled data in supervised learning. In: Proceedings of the Sixth International Colloquium on Cognitive Science (1999)
21. Ji, X., Xu, W.: Document clustering with prior knowledge. In: SIGIR 2006, pp. 405–412 (2006)
22. Wu, X., Srihari, R.: Incorporating prior knowledge with weighted margin support vector machines. In: KDD 2004, pp. 326–333 (2004)

Constrained Local Regularized Transducer for Multi-Component Category Classification

Congle Zhang and Yong Yu

Department of Computer Science and Engineering,
Apex Lab, Shanghai Jiaotong University, Shanghai, 200240
{zhangcongle,yyu}@apex.sjtu.edu.cn

Abstract. Transductive learning is proposed to incorporate both labeled and unlabeled examples into the learning process. Several methods have been developed and show encouraging performance. However, people may meet complicated classification tasks in real world applications, where one category contains multiple components. Traditional transductive learning algorithms are not very effective in such settings. In this paper, we propose a novel transductive learning approach called *constrained local regularized transducer*(CLRT) for multi-component category classification. CLRT is based on the local separable assumption that it is possible to build a linear predictor in one small area. We implement the assumption by minimizing a unified objective function, which can be optimized globally. Experiment results validate that CLRT can achieve satisfied performance robustly and efficiently.

1 Introduction

Transductive learning is widely used in many classification tasks [15]. Besides labeled examples for each category, unlabeled examples are already known when training the classifier, which can be called *transducer* in transductive learning. Some approaches, such as TSVM [15,6], use unlabeled examples to adjust the global hypothesis. Others like spectral graph transducer [7], normalized cut [14] follows the local consistency assumption that near examples tend to have the same assignment.

We notice in real world applications, people often meet complicated problems where one category contains more than one component. For example, in user interest classification, a user may have several interests that differ a lot from each other. In automatic diagnosis, patients suffering from the same disease often show quite different symptoms. We call such problem *multi-component category classification*. Figure 1 is a simple example. There are two challenges for transductive learning in such settings. Firstly, examples are not linearly separable globally. Secondly, components of the same category can be far in distance, such as components *A* and *C* in Figure 1. Comparatively, components of different categories will be very close (like *A* and *B*), which makes near examples have different labels. Because of these challenges, traditional approaches relying on global linear predictor or local consistency assumption will not be very effective in multi-component category setting.

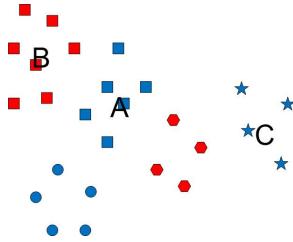


Fig. 1. An multi-component category classification example, red/blue examples belong to two categories

To overcome these challenges, we propose a novel *constrained local regularized transducer* (CLRT) for multi-component category classification. CLRT is based on *local separable assumption*, which means examples in a small area belong to at most two components and can be separated by a linear predictor if they have different labels. With this assumption in mind, besides keeping consistent with pre-given labels on training set, the generalized assignments should follow the principle that when examples of different categories are in one small area, their assignments should meet the result of the local linear predictor. On one hand, CLRT based on such principles enforces assignments fit multiple local predictors instead of a global linear predictor. On the other hand, even when the unlabeled example has neighbors on the components of different categories, local linear predictor can separate them apart and prevent it having the same assignment with its neighbors. Therefore, CLRT can deal well with challenges that global linearly separable and local consistency properties do not hold in multi-category setting.

In this paper, constrained local regularized transducer implements local separable assumption by minimizing a unified objective function, which is a trade-off between the constraint part and the local regularized part. The constrained part gives square loss as the training error when transducer assigns the training examples with wrong labels. The local regularized part considers every example on its neighborhood and enforce its assignment to keep consistent with the local regularized linear predictor. The unified objective function can be globally optimized by an extended spectral methods.

We conduct experiments on 20-Newsgroup / Reuters-21578 to provide empirical evidence for our approach. Randomly selected components are mixed together to form both the positive and the negative categories. We apply the proposed CLRT in such settings and compare it with several state-of-the-art classification algorithms, including kNN, SGT, SVM and TSVM with radial basis kernel. For categories containing two components, CLRT achieves 0.070/0.024 on error rate averagely, and 0.091/0.041 averagely for categories containing three components. Such results significantly outperform comparison methods. Besides, CLRT also achieves satisfied performance on traditional settings (i.e. each category contains a single component).

2 Related Work

Transductive learning is proposed by Vapnik in [15]. In this work, transductive support vector machine (TSVM) is designed with the goal of using unlabeled data to adjust the classifier, which is later refined in [6]. TSVM separates data into two sides by maximizing margin for both labeled and unlabeled examples. Some transductive approaches tend to classify near data into the same category according to the local consistency property. Works (such as Mincut [1], multi-way cuts [8], normalized cut [14], and ratio-cut [2]) present the data relationship in an undirected graph and put cut operations on that graph. Spectral graph partitioning (SGT) in [7] added some additional heuristics to improve graph cut performance.

In practical, people will meet complicated classification problems that one category contains multiple components. Work in [11] analyzes user behavior in Internet and points out that users usually have multiple interests. Weblogs are data sources with mixture topics for each blogger [10]. Our CLRT algorithm is designed for such multi-component category setting, where local consistency property may not hold well. We make empirical comparisons to prove this claim.

When categories contain multiple components, examples are not linear separable globally. Kernel functions [13] can map the data into a higher dimensional space and may make it possible to perform the linear separation. Using kernel functions, support vector machines turn to alternative training methods for polynomial, radial basis function (rbf) and multi-layer perceptron classifiers. In our experiment, we test SVM and TSVM with rbf kernel to show that CLRT can reach satisfied performance.

We are inspired by the idea that examples are linearly separable in a small area. Work in [9] proposes the idea of training classifier locally. Locally linear embedding [12,?] attempts to discover nonlinear structure in high dimensional data by exploiting the local symmetries of linear reconstructions. Zhou [19] proposes a learner keeping both local and global consistency. Works in [16,17] provide novel clustering methods with local information. In this paper, we put the local linearly separable idea in transductive learning and design our CLRT for multi-component category classification.

3 Constrained Local Regularized Transducer

The learning task is defined on data array X of n examples $X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$. \mathbf{x}_i is an m dimension feature array. For training, examples with ids in $S_l \subset \{1, 2, \dots, n\}$ are labeled. For simplicity, let's assume labels are binary. We denote $Y^l = (y_1^l, y_2^l, \dots, y_n^l)$, with $y_i^l \in \{-1, 0, 1\}$ representing the label of \mathbf{x}_i . We use $y_i^l = 0$ to denote that \mathbf{x}_i is unlabeled. With X and Y^l , the input for transducer is $\mathcal{D} = \{(\mathbf{x}_i, y_i^l) | i = 1, 2, \dots, n\}$. Transducer estimates the function $f_{\mathcal{D}} : \mathbf{x}_i \rightarrow z_i$, where $z_i = f_{\mathcal{D}}(\mathbf{x}_i)$ is the assignment for x_i . Notice here X is an $m \times n$ feature matrix while Y^l is an n length vector.

Our idea is that for every example, its assignment should meet the result of the linear predictor decided by its neighborhood. We extend regularized linear classifier as our local linear predictor, which will be introduced first. Then we

combine these local linear predictors together according to the rule of minimizing leave-one-out error. After adding constraint to keep assignment consistent to training examples and taking the normalizing problem into account, we arrives at the objective function for CLRT.

3.1 Regularized Linear Classification

Regularized linear classification [18] is a supervised learning approach. It takes labeled examples (i.e. $\{\mathbf{x}_i | i \in S_l\}$) as input and generates a unifier linear predictor \mathbf{w}^* for test examples. If the square loss is applied to estimate the linear predictor, the objective function can be written as:

$$\mathcal{L}_{RLC} = \frac{1}{n} \sum_{i \in S_l} (\mathbf{w}^T \mathbf{x}_i - y_i^l)^2 + \mu \|\mathbf{w}\|^2 \quad (1)$$

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \mathcal{L}_{RLC} \quad (2)$$

$$z_i = f_{\mathcal{D}}(\mathbf{x}_i) = \text{sign}(\mathbf{w}^{*T} \mathbf{x}_i) \quad (3)$$

The first term of Equation (1) is the loss on the training set, the second term is a regularized term to prevent the predictor itself from overfitting. To solve above equations, we can take $\frac{\partial \mathcal{L}_{RLC}}{\partial \mathbf{w}} = 0$ and get

$$\mathbf{w}^* = \left(\sum_{i \in S_l} \mathbf{x}_i \mathbf{x}_i^T + n\mu I \right)^{-1} \left(\sum_{i \in S_l} \mathbf{x}_i^T y_i^l \right) \quad (4)$$

For convenient discussion, we rewrite the loss using feature matrix X and label Y^l to

$$\mathbf{w}^* = (X X^T + n\mu I)^{-1} X Y^l \quad (5)$$

X is an $m \times n$ matrix. When $m \gg n$, calculating the inverse of $X X^T + n\mu I$ is very expensive, an equivalent representation is proved in [16] that:

$$\mathbf{w}^* = X (X^T X + n\mu I)^{-1} Y^l \quad (6)$$

this makes it possible to obtain linear predictor by calculating the inverse of an $n \times n$ matrix, which is much easier when $m \gg n$. Linear regularized classifier here is a global predictor. Its empirical success relies on the fitness of the assumption that examples are linearly separable globally. In multi-component category setting, we want to build linear classifier for every example in its neighborhood according to locally separable assumption. However, labeled examples are scattered among components, which makes labeled examples not enough for training. To avoid this limitation, we utilize unlabeled examples to provide structure information to help build local predictors.

3.2 Transductive Learning with Regularized Local Predictors

Local predictors are built for every example by mining the structure in their neighborhoods. For one example, its assignment should keep consistent with the

local linear predictor. In multi-component setting, when neighbors of \mathbf{x}_i belong to different categories, local predictor helps us to tell \mathbf{x}_i from them. Formally, we denote \mathbf{x}_i 's neighborhood as $X_i = (\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{ik})$, where \mathbf{x}_{ij} is the j-nearest example to \mathbf{x}_i in X , i.e. X_i is an $m \times k$ matrix. Similar to Equation (1), a local predictor \mathbf{w}_i for \mathbf{x}_i is to minimize:

$$\mathcal{L}_i = \frac{1}{k} \sum_{\mathbf{x}_j \in X_i} (\mathbf{w}_i^T \mathbf{x}_j - z_j)^2 + \mu \|\mathbf{w}_i\|^2 \quad (7)$$

where $z_j = f_D(\mathbf{x}_j)$ is the assignment for \mathbf{x}_j . We will carry these assignments in local predictors until we can optimize them globally. We can apply the same way in Equation (5) to find \mathbf{w}_i^* minimizing \mathcal{L}_i that:

$$\mathbf{w}_i^* = (X_i X_i^T + k\mu_i I_i)^{-1} X \mathbf{z}_i \quad (8)$$

where $\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{ik})$, with z_{ij} the assignment for \mathbf{x}_{ij} . We convert \mathbf{w}_i^* to the form as equation (6) to reduce computation cost:

$$\mathbf{w}_i^* = X_i (X_i^T X_i + k\mu_i I_i)^{-1} \mathbf{z}_i \quad (9)$$

Now the problem is to combine local predictors together and to optimize assignments $\{z_i\}$ in $\{\mathbf{w}_i^*\}$. Notice that each \mathbf{w}_i^* is obtained without the value of \mathbf{x}_i . Thus, local regularized predictor is the result of “leave-one-out” learning. By applying the rule of minimizing leave-one-out error [4], we can minimize:

$$\begin{aligned} \mathcal{L}^{loo} &= \frac{1}{n} \ell(f_D^{loo}(\mathbf{x}_i) - z_i)^2 = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i^T \mathbf{w}_i^* - z_i)^2 \\ &= \frac{1}{n} (\mathbf{x}_i^T X_i (X_i^T X_i + k\mu_i I_i)^{-1} \mathbf{z}_i - z_i)^2 \end{aligned} \quad (10)$$

where $f_D^{loo}(\mathbf{x}_i)$ is the assignment of the local predictor for \mathbf{x}_i . We apply the square loss for each local learner. In equation 10, only $\{z_i\}$ are unknown values. If we define $\beta_i = e_i$ (the i'th column of $n \times n$ identity matrix) and α_i with j-th entry as

$$\alpha_{ij} = \begin{cases} (\mathbf{x}_i^T X_i (X_i^T X_i + k\mu_i I_i)^{-1})_j & j \in \mathcal{N}_i \\ 0 & o.w \end{cases} \quad (11)$$

i.e. α_i is obtained by extending $\mathbf{x}_i^T X_i (X_i^T X_i + k\mu_i I_i)^{-1}$ to an n -length vector by adding some zero entries. Then the leave-one-out loss is

$$\begin{aligned} \mathcal{L}^{loo} &= \frac{1}{n} \sum_{i=1}^n ((\alpha_i - \beta_i) \mathbf{z})^2 \\ &= \frac{1}{n} \sum_{i=1}^n \mathbf{z}^T (\alpha_i - \beta_i) (\alpha_i - \beta_i)^T \mathbf{z} \end{aligned} \quad (12)$$

We denote an $n \times n$ matrix M as $M = \frac{1}{n} \sum_{i=1}^n (\alpha_i - \beta_i)(\alpha_i - \beta_i)^T$, then $\mathcal{L}^{loo} = \mathbf{z}^T M \mathbf{z}$.

3.3 Normalized Local Predictors with Constraint

A minimized $\mathbf{z}^T M \mathbf{z}$ enforces the assignment for every example \mathbf{x}_i to keep consistent with its local predictor decided by neighborhood X_i . M provides a good foundation to design the transducer for multi-component category classification. We first discuss some properties of M and then we can arrive at the objective function for the transducer.

Lemma 1. *With M 's Laplacian $L_M = M' - M$ where M' is a diagonal matrix with $M'_{ii} = \sum_j M_{ij}$, the leave-one-out loss can be written as:*

$$\mathcal{L}^{loo} = \mathbf{1}^T M \mathbf{1} - \mathbf{z}^T L_M \mathbf{z} \quad (13)$$

Proof. $\mathcal{L}^{loo} = \mathbf{1}^T M \mathbf{1} - \mathbf{z}^T L_M \mathbf{z}$

$$\begin{aligned} &= \mathbf{1}^T M \mathbf{1} - (\mathbf{z}^T M' \mathbf{z} - \mathbf{z}^T M \mathbf{z}) \\ &= \sum_{i,j} M_{ij} - \left(\sum_i z_i^2 \sum_j M_{ij} - \mathbf{z}^T M \mathbf{z} \right) \\ &= \mathbf{z}^T M \mathbf{z} \end{aligned} \quad (14)$$

Lemma 2. *For an undirected weighted graph $G = (V, E)$, $V = \{v_i\}$ is the node set with n nodes. Each node v_i has the assignment $z_i \in \{+1, -1\}$. E is the edge set $\{(v_i, v_j, M_{ij}) | v_i, v_j \in V\}$, M_{ij} is the weight between node v_i and node v_j . The size of the cut which split G into positive nodes and negative nodes is $\mathbf{z}^T L_M \mathbf{z}$, where L_M is defined in lemma 1.*

Lemma (2) can be proved easily. Lemma (1) rewrites the leave-one-out loss and lemma (2) tells us its graph meaning. Combining lemma (1) and (2), we can arrive at following conclusion:

Theorem 1. *Finding \mathbf{z} that minimize the leave-one-out error for regularized local predictors in equation 12 can be converted to finding minimum cut in undirected weighted graph G with adjacent matrix M .*

Now we convert the multi-component category classification problem to minimizing graph cut. For practical problems, it is necessary to balance the number of nodes in each cut when minimizing the sum of the edge weights cut through. It is because simply minimizing the cut size leads to degenerate cuts easily. For example, one cut is very small while the other cut contains almost all nodes.

Fortunately, transductive learning problems have labeled data to guide graph split. [7] proposes a reasonable principle to put on the transductive solution that averages over examples (e.g. average margin, pos/neg ratio) should have the same expected value in the training and test sets.

Suppose we are given n_+ positive labeled data and n_- negative labeled examples. Following [3], the normalized unsupervised optimization problem can equivalently be written as

$$\min_{\mathbf{z}} \frac{\mathbf{z}^T M \mathbf{z}}{\mathbf{z}^T \mathbf{z}} \quad \text{with} \quad z_i \in \{\gamma_+, \gamma_-\} \quad (15)$$

where $\gamma_+ = \sqrt{\frac{n_-}{n_+}}$ and $\gamma_- = \sqrt{\frac{n_+}{n_-}}$. When “same expected value” principle is met, we have $\mathbf{z}^T \mathbf{z} = n$ and $\mathbf{z}^T \mathbf{1} = 0$, which balance the positive and negative examples. Since solving (15) is NP hard, we can minimize its real relaxation as:

$$\min_{\mathbf{z}} \quad \mathbf{z}^T M \mathbf{z} \quad (16)$$

$$s.t. \quad \mathbf{z}^T \mathbf{1} = 0 \text{ and } \mathbf{z}^T \mathbf{z} = n \quad (17)$$

Taking the labeled examples in Y^l into account, we employ the quadric loss to constrain the transducer as:

$$\sum_{i:y_i^l=+1} c_i(z_i - \gamma_+)^2 + \sum_{i:y_i^l=-1} c_i(z_i - \gamma_-)^2 \quad (18)$$

c_i allows various cost for different labeled examples. The meaning of (18) is clear: when z_i is assigned the desired label in training set, the loss is zero; when they are different, the above-zero value is added to the loss. Adding the quadric loss to Equation (16) and then turning it to matrix form, we arrive at the objective function for CLRT as:

$$\min_{\mathbf{z}} \quad \mathbf{z}^T M \mathbf{z} + c(\mathbf{z} - \boldsymbol{\gamma}) C (\mathbf{z} - \boldsymbol{\gamma}) \quad (19)$$

$$s.t. \quad \mathbf{z}^T \mathbf{1} = 0 \text{ and } \mathbf{z}^T \mathbf{z} = n \quad (20)$$

where the elements of $\boldsymbol{\gamma}$ is equal to γ_+ for positive labeled examples, γ_- for negative labeled examples and zero for other unlabeled examples. c is the trade-off parameter balance the effect of local regularized predictors and the constraint on training set. C is a diagonal matrix with $C_{ii} = c_i$ for labeled examples and zero for unlabeled ones. In order to balance the effect of positive and negative labeled examples, we can set $C_{ii} = \frac{n_-}{n_+ + n_-}$ for positive labeled examples and $C_{ii} = \frac{n_+}{n_- + n_+}$ for negative ones.

Minimizing (19) can well reach multi-component category classification purpose. For every example, its assignment keeps consistent with examples in neighborhood following local separable rule by minimizing local regularized part. Meanwhile, classification results for all examples are consistent to training set by minimizing constraint part. Consistency on the training set and local linear predictors make each example correctly influences its neighbors when they are in same category and distinguish from its neighbors when they have different labels.

Equation (19),(20) can be solved by spectral methods introduced in [5]. For M , we take its Laplacian L_M defined in lemma 1. Then, we make eigen-decomposition for L_M and get $L_M = U \Sigma U^T$. Each column of U is one eigenvector, whose corresponding eigenvalue is in diagonal matrix Σ . For \mathbf{z} , we can find a \mathbf{u} and write \mathbf{z} as $\mathbf{z} = U \mathbf{u}$ because U is invertible. There is an easily verified property for L_M that $L_M \cdot \mathbf{1} = 0 \cdot \mathbf{1}$. This means L_M has an eigenvector $\mathbf{1}$ corresponding to eigenvalue 0. It is well known that for a real symmetric matrix, eigenvectors corresponding to different eigenvalues are orthogonal to each other. Since eigenvalue 0 corresponds to a single eigenvector $\mathbf{1}$, all vectors except $\mathbf{1}$ in U are orthogonal to $\mathbf{1}$. Let $\Sigma_{11} = 0$,

in order to make $\mathbf{z}^T \mathbf{1} = 0$, the first element of \mathbf{u} have to be zero. We denote U' as the matrix obtained by removing eigenvector $\mathbf{1}$ from U . Then we have $\mathbf{z} = U' \mathbf{u}'$, where \mathbf{u}' is obtained by removing the first element in \mathbf{u} . This guarantees $\mathbf{z}^T \cdot \mathbf{1} = 0$. With U' , \mathbf{u}' , (19),(20) can be changed to the following equivalent optimization problem:

$$\begin{aligned} & \min_{\mathbf{u}'} \mathbf{u}'^T U'^T M U' \mathbf{u}' + c(U' \mathbf{u}' - \boldsymbol{\gamma})^T C(U' \mathbf{u}' - \boldsymbol{\gamma}) \\ & \text{s.t. } \mathbf{u}'^T \mathbf{u}' = n \end{aligned} \quad (21)$$

(21) comes from $n = \mathbf{z}^T \mathbf{z} = \mathbf{u}'^T U'^T U' \mathbf{u}' = \mathbf{u}'^T \mathbf{u}'$. By merging square terms, linear terms and constant terms respectively, we have the objective function in following form:

$$\min_{\mathbf{u}'} \mathbf{u}'^T A \mathbf{u}' + B \mathbf{u}' \quad (22)$$

where $A = U'^T (M + c \cdot C) U'$, $B = -2cU'^T C \boldsymbol{\gamma}$. The constant item has been dropped in (22). Work in [5] shows the way to solve this kind of optimization that the optimized \mathbf{u}'^* is $\mathbf{u}'^* = (A - \lambda^* I)^{-1} B$, where λ^* is the smallest eigenvalue of

$$D = \begin{pmatrix} A & -I \\ -\frac{1}{4n} B B^T & A \end{pmatrix} \quad (23)$$

With \mathbf{u}'^* , we have optimized $\mathbf{z}^* = U' \mathbf{u}'^*$, whose entries are generalized assignments for examples.

4 Experiments

In this section, we first introduce the design of the experiment, including comparison methods and the way to obtain meaningful data sets for multi-component setting. Then the experiment results are given to compare the performance among various methods. We also test some properties of CLRT, such as parameter influence and time cost.

4.1 Experiment Design

We evaluate the performance of CLRT using two public data corpora: the Reuters-21578 and the 20 Newsgroups. These two corpora have been widely used for classification evaluations. The Newsgroup contains about 20,000 documents that were collected from 20 news groups in the public domains. We view each news group as one component. Documents in Reuters have topic properties. We discard the documents with multiple topics and remove the topics with less than 60 documents. This lead to 8526 documents covering 14 topics. We view each newsgroup in 20ng and each topic in reuters as one component.

To validate that CLRT can deal well in multi-component environment, we generate categories by merging components together. We use notation $id_1_id_2\dots$

$_id_n$ to denote a multi-component category, which contains documents on components with corresponding ids according to their alphabet order. In this paper, we conduct experiments on categories containing two and three components.

We adopt following state-of-the-art algorithms as comparison methods: (1)k nearest neighbor classifier: kNN classifier implements local consistency property in an intuitive method that classifying an example into the category having most labeled examples in its k nearest neighbors; (2)spectral graph transducer (SGT) [7]: finds a constrained normalized minimum cut in the graph with distance matrix as adjacent matrix to enforce near examples classified into the same category; (3) support vector machine with rbf kernel (SVM-R)[15]: rbf kernel maps the distance matrix to a new space and then conducts svm classification; (4)transductive support vector machine with rbf kernel (TSVM-R)[6]: besides kernel mapping, it-eratively adjusts classifier by taking unlabeled examples into account.

Among four comparison methods, SGT and SVM-R are transductive approaches. KNN and Sgt are based on local consistency assumption. SVM-R and TSVM-R makes kernel mapping. Since SGT, SVM-R and TSVM-R are designed for binary classification, the comparisons in this paper are also based on binary classification for simplicity.

We employ error rate as the performance metric. Besides performance comparisons against other methods in multi-component setting, we also evaluate CLRT in traditional setting, i.e. each category contains a single component. Relations of InputSize-TimeCost and Parameter-Performance will be demonstrated finally.

4.2 Experiment Results

Overall Performance. For Reuters and 20ng, we test the various methods on categories containing two topics and three topics respectively. Thus, there are four groups of data sets. We randomly generate 100 data sets for each group. We use ids according to alphabet order to describe them. For example, 1_2_vs_3_4 means that the positive category contains documents on “comp.graphics”, “ms-windows” and the negative category contains “ibm.hardware”, “mac.hardware”. We obtain performance of various methods for all 400 data sets. For limited space, we show the results for part of them and the average cases for each group.

We randomly select 10 examples as the training set for each component to test the performance with small training set. For CLRT, we adopt the same parameters for all data sets to validate its robustness. We set local range k as 50, regularized term weight μ as 1, constraint weight c as 10. For kNN, we check 10 nearest neighbor for each test example. For SGT, we use default parameters in [7]. For SVM-R and TSVM-R, we set the variance as 0.1 and use default values for other parameters.

Table 1 shows the performance comparisons. Our CLRT approach outperforms all other methods in most of the data sets. KNN arrives at a poor performance because nearest neighbors often belong to other categories with few training examples. SGT may have good performance in some cases. But when the local same assumption is not well satisfied, the performance will evidently reduce. Thus it is not an reliable approach in our setting. Kernel mapping

Table 1. Performance comparison of 5 methods on Reuters and 20Ng

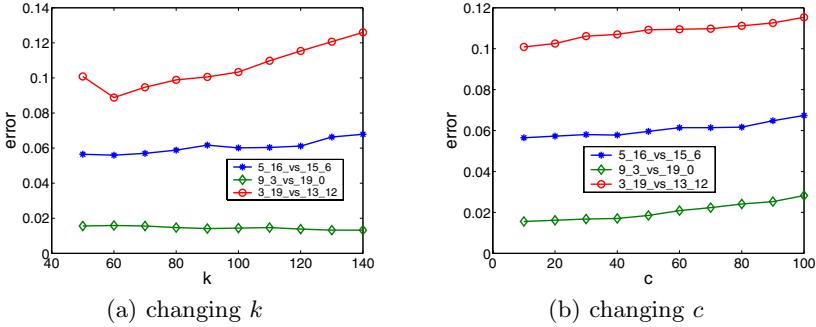
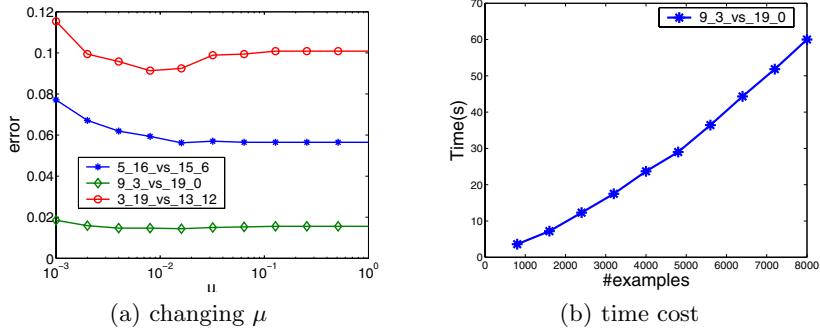
Group Name	Data Set	Error Rate					Improvement			
		CLRT	kNN	SGT	SVM-R	TSVM-R	kNN	SGT	SVM-R	TSVM-R
Average	avg.20ng-2	0.070	0.216	0.173	0.209	0.160	2.06	1.46	1.96	1.28
	avg.reuters-2	0.024	0.218	0.104	0.053	0.056	7.89	3.25	1.15	1.30
	avg.20ng-3	0.091	0.276	0.218	0.213	0.369	2.04	1.40	1.35	3.07
	avg.reuters-3	0.041	0.280	0.215	0.060	0.062	5.82	4.24	0.47	0.52
20ng-2	5.16_vs_15.6	0.056	0.272	0.253	0.165	0.487	3.81	3.47	1.92	7.62
	9.3_vs_19.0	0.016	0.046	0.045	0.265	0.025	1.98	1.87	15.99	0.60
	3.19_vs_13.12	0.101	0.283	0.318	0.384	0.550	1.81	2.16	2.81	4.46
	0.9_vs_15.12	0.064	0.246	0.226	0.287	0.474	2.83	2.53	3.47	6.39
	6.7_vs_15.12	0.140	0.313	0.275	0.263	0.155	1.23	0.96	0.87	0.11
Reuters-2	27.22_vs_30.29	0.015	0.251	0.045	0.025	0.025	16.03	2.07	0.68	0.68
	30.21_vs_24.26	0.017	0.032	0.022	0.060	0.063	0.90	0.30	2.59	2.77
	27.31_vs_22.29	0.003	0.192	0.051	0.068	0.068	75.70	19.40	26.07	26.07
	22.21_vs_32.31	0.007	0.288	0.083	0.031	0.038	41.67	11.26	3.56	4.61
	27.23_vs_21.25	0.005	0.009	0.005	0.018	0.022	1.00	0.06	3.04	3.94
20ng-3	8.18_0_vs_9.15.12	0.078	0.307	0.312	0.307	0.462	2.94	3.01	2.94	4.93
	6.19_16_vs_11.7.5	0.072	0.267	0.184	0.211	0.456	2.71	1.56	1.94	5.35
	3.19_13_vs_12.1.18	0.157	0.424	0.344	0.311	0.485	1.70	1.19	0.98	2.09
	16.15.1_vs_4.7.12	0.069	0.186	0.218	0.187	0.187	1.70	2.16	1.71	1.70
	8.15_3_vs_12.7.16	0.095	0.351	0.350	0.222	0.217	2.69	2.67	1.33	1.28
Reuters-3	32.21.23_vs_24.30.29	0.009	0.067	0.267	0.033	0.020	6.35	28.11	2.60	1.20
	21.26.22_vs_27.23.20	0.025	0.365	0.242	0.053	0.031	13.48	8.60	1.10	0.22
	26.33.30_vs_29.23.21	0.003	0.034	0.033	0.016	0.009	12.55	12.15	5.60	2.77
	28.30.32_vs_31.21.24	0.061	0.339	0.116	0.047	0.018	4.56	0.91	-0.23	-0.71
	20.29.22_vs_32.30.23	0.068	0.340	0.340	0.054	0.011	3.98	3.98	-0.21	-0.84

Table 2. Performance Comparison with SGT in traditional setting (category contains a single component)

Data Set	CLRT	SGT	Data Set	CLRT	SGT
avg.20ng	0.034	0.028	avg.reu	0.024	0.032
5_vs_16	0.008	0.005	22_vs_29	0.005	0.005
15_vs_6	0.022	0.021	20_vs_32	0.012	0.010
9_vs_3	0.008	0.006	20_vs_22	0.030	0.029
1_vs_18	0.014	0.013	21_vs_32	0.011	0.006
18_vs_6	0.033	0.030	31_vs_24	0.008	0.022
2_vs_6	0.159	0.122	23_vs_31	0.072	0.224
12_vs_8	0.024	0.022	10_vs_27	0.097	0.261
11_vs_12	0.036	0.020	23_vs_21	0.000	0.015
3_vs_19	0.013	0.013	25_vs_29	0.007	0.008
13_vs_12	0.048	0.046	30_vs_27	0.097	0.261

methods of SVM-R and TSVM-R can in some extent improve the performance. The comparison shows that CLRT based on local separable assumption still has advantage over such kernel mapping approaches.

We further test CLRT in traditional setting. We randomly generate 100 data sets for 20ng and Reteurs with categories each containing a single component. We compare CLRT with SGT on these data sets. The results in Table 2 show

Fig. 2. Parameters k and c Fig. 3. Parameter μ and time cost

that CLRT reaches comparable performance against SGT, which validates the adaptivity of CLRT.

Parameters and Time Cost. CLRT has three parameters: local range k , weight μ for regularized term and weight c for constraint. We hope the performance is not sensitive to these parameters. Experiment results in figure 2(a), 2(b) and 3(a) practically validates this claim. The y-axis is the error rate while the x-axis is the changing parameters. From these figure, we can see that CLRT reach reliable performance with k between $40 - 140$, c between $10 - 100$ and μ between $0.001 - 1$. Figure 3(b) show time cost with different input size. From this figure, we know the time complexity of CLRT is almost linear to the size of examples.

5 Conclusions and Future Works

In this paper, we derive a novel transductive learning algorithm called constrained local regularized transducer for multi-component category classification. CLRT based on the local separable assumption deals well with the challenges

that global linearly separable and local consistency properties do not hold in our setting. Our experiment validates that the proposed CLRT can reach satisfied performance robustly and efficiently which significantly outperform comparison methods. In future, we will focus on extending CLRT to new applications, such as images, sounds, videos and so on.

References

1. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph min-cuts. In: ICML, pp. 19–26 (2001)
2. Chan, P.K., Schlag, M.D.F., Zien, J.Y.: Spectral k-way ratio-cut partitioning and clustering. In: DAC, pp. 749–754 (1993)
3. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: KDD, pp. 269–274 (2001)
4. Evgeniou, T., Pontil, M., Elisseeff, A.: Leave one out error, stability, and generalization of voting combinations of classifiers. Machine Learning, 71–97 (2004)
5. Gander, W., Golub, G., von Matt, U.: A constrained eigenvalue problem. In: Linear Algebra and its Application, pp. 114/115,815–839 (1989)
6. Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML, pp. 200–209 (1999)
7. Joachims, T.: Transductive learning via spectral graph partitioning. In: ICML, pp. 290–297 (2003)
8. Kleinberg, J.M., Tardos, E.: Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. In: FOCS, pp. 14–23 (1999)
9. Bottou, L., Vapnik, V.: Local learning algorithms. Neural Computation, 888–900 (1992)
10. Mei, Q., Ling, X., Wondra, M., Su, H., Zhai, C.: Topic sentiment mixture: modeling facets and opinions in weblogs. In: WWW, pp. 171–180 (2007)
11. Pon, R., Cardenas, A., Buttler, D., Critchlow, T.: Tracking multiple topics for finding interesting articles. In: KDD, pp. 560–569 (2007)
12. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
13. Scholkopf, B., Smola, A.J.: Learning with kernels. MIT Press, Cambridge (2002)
14. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: CVPR, pp. 731–737 (1997)
15. Vapnik, V.: Statistical Learning theory. Wiley, Chichester (1998)
16. Wang, F., Zhang, C.: Regularized clustering for documents. In: SIGIR, pp. 95–102 (2007)
17. Wu, M., Scholkopf, B.: A local learning approach for clustering. In: NIPS, pp. 1529–1536 (2006)
18. Zhang, T., Oles, F.J.: Text categorization based on regularized linear classification methods. Journal of Information Retrieval 4, 5–31 (2001)
19. Zhou, D., Bousquet, O., Lal, J.W.T.N., Scholkopf, B.: Learning with local and global consistency. In: NIPS (2003)

Low Resolution Gait Recognition with High Frequency Super Resolution

Junping Zhang^{1,2}, Yuan Cheng², and Changyou Chen²

¹ Shanghai Key Laboratory of Intelligent Information Processing
jpzhang@fudan.edu.cn

<http://www.iipl.fudan.edu.cn/> zhangjp

² Department of Computer Science and Engineering
Fudan university
Handan Road 220, Shanghai 200433, China
0472404@fudan.edu.cn, cchangyou@gmail.com

Abstract. Being non-invasive and effective at a distance, recognition suffers from low resolution sequence case. In this paper, we attempt to address the issue through the proposed high frequency super resolution method. First, a group of high resolution training gait images are degenerated for capturing high-frequency information loss. Then the combination of neighbor embedding with interpolation methods is employed for learning and recovering a high resolution test image from low resolution counterpart. Finally, classification is performed based on nearest neighbor classifier. The experiment indicates that the proposed method can effectively improve the accuracy of gait recognition under low resolution case.

Keywords: Gait Recognition; Super Resolution; Gait Energy Image.

1 Introduction

Unlike other biometric authentication techniques such as face recognition and fingerprint recognition, gait recognition has attracted more and more attention in last decade due to the fact that it is effective at a distance and non-invasive, and gait is difficult to conceal. Wang et al. [1] employed procrust analysis, which is one of direction statistics based methods, to capture the mean shapes of gait silhouettes. Assuming that gait data can be viewed as cubic data, Kobayashi [2] extracted the divergences between different states of gaits based on “Cubic Higher-order Local Auto-correlation (CHLAC)”. To obtain more discriminant features, Horita1 [3] refined the definition of CHLAC in [3]. Furthermore, gait energy image (GEI) is proposed to transform gait sequences of each person into image [4].

However, gait recognition easily suffers from some exterior factors including long distance from camcorders to objects (e.g. ≥ 10 meters), and inferior imaging quality as well as uncontrollable environmental phenomena. One reason is that such factors result in the appearance of low resolution gait images or sequences.

Furthermore, in many places, low-resolution sampling devices which can degenerate the performance of gait recognition are helpful for saving cost. Therefore, utilizing inexpensive software is a economical and rational way for recovering the high resolution gait sequences. As a result, it arises a great challenge for researchers and scientists in the domain of gait recognition.

Super resolution, which is a kind of image processing method, is devoted to recovering high resolution image from at least one low resolution image. Roughly speaking, it can be divided into three categories: interpolated-based, learning-based and reconstruction-based ways [5,6]. Among them, learning-based method is to learn high resolution image from a single or a collection of low resolution pixel-based images. For example, Freeman et al. developed a one-pass example-based super resolution algorithm through taking neighborhood effects into account [5]. The disadvantage is that non-photo-realistic artifacts which look like the style of oil painting are introduced. More recently, Chang et al. proposed super resolution through neighbor embedding (SRNE) [7]. A main assumption is that neighborhood relationship between an image patch and its neighbor patches should be preserved when these low resolution patches are up-sampled to the high resolution counterparts. The reported results indicate that images being recovered by the SRNE algorithm look more natural and photo-realistic compared to Freeman's method [5]. A major disadvantage is that the performance of the SRNE algorithm depends on the selection of image features.

Considering the mentioned advantages and disadvantages of super resolution, we thus attempt to resolve the issue of low resolution gait recognition through our proposed super resolution method. First, a collection of high frequency training patches are constructed through computing the difference between high resolution training gait images and their corresponding downsampled gait images. Secondly, these downsampled gait images are downsampled again for obtaining high frequency low resolution patches. Thirdly, high frequency high resolution test patches are learned from the high frequency training gait image pairs. Fourthly, high resolution test image is formed by merging high frequency test patches and interpolated test patches. Finally, high resolution test image is classified by nearest neighbor classifier. Experiments on a gait benchmark database show the promising advantages of the proposed algorithm.

The paper is organized as follows. In Section 1, we give a brief introduction on gait recognition and super resolution. In Section 2, high frequency super resolution for low resolution gait recognition is proposed. In Section 3, experimental result is reported. Finally, we discuss some further works and conclude the paper.

2 High Frequency Super Resolution for Gait Recognition

Generally speaking, when a subject is gradually away from a camcorder or sampled under low resolution sensors, the resultant low resolution gait sequences will impair the performance of gait recognition. A possible reason is that high frequency details which are important for gait recognition have been lost. If high frequency details of gait sequences can be recovered, therefore, the accuracy of

low resolution gait recognition will be improved, and gait recognition will have a possibility of recognizing subjects at longer distance with the same or higher accuracy.

2.1 Gait Energy Image

Simple and easy to implement, Gait energy image (GEI) is an efficient way for gait recognition[4]. Let $B_1, B_2, \dots, B_n \in \mathbb{R}^m$ be a sequence of gait frames in one gait cycle. Formally, GEI is defined as follows [4]:

$$M = \frac{1}{n} \sum_{i=1}^n B_i \quad (1)$$

which is equivalent to:

$$\begin{aligned} M &= \arg_X \min \frac{1}{n} \sum_{i=1}^n \|B_i - X\|^2 \\ &= \arg_X \min \frac{1}{n} \sum_{i=1}^n (B_i - X)^T (B_i - X) \end{aligned} \quad (2)$$

From equation (2) it is not difficult to see that essentially, GEI is to minimize the sum of discrepancies of all gait frames. Fig.1 depicts some examples of GEIs.

2.2 Extraction of High Frequency Details

After the GEI is obtained, high frequency details will be extracted from both high resolution GEIs and low resolution GEIs. More specifically, a high resolution image G is β -times (e.g., $\beta = 2$) downsampled to a low resolution one GL . Then GL is upsampled by some interpolation methods such as bilinear interpolation. Because interpolation can only recover middle and low frequency information

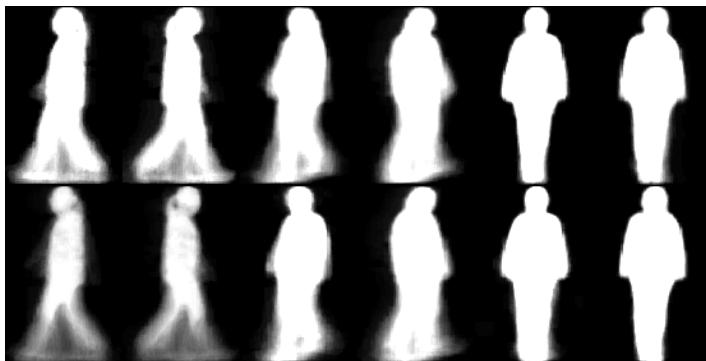


Fig. 1. Several examples of gait energy images

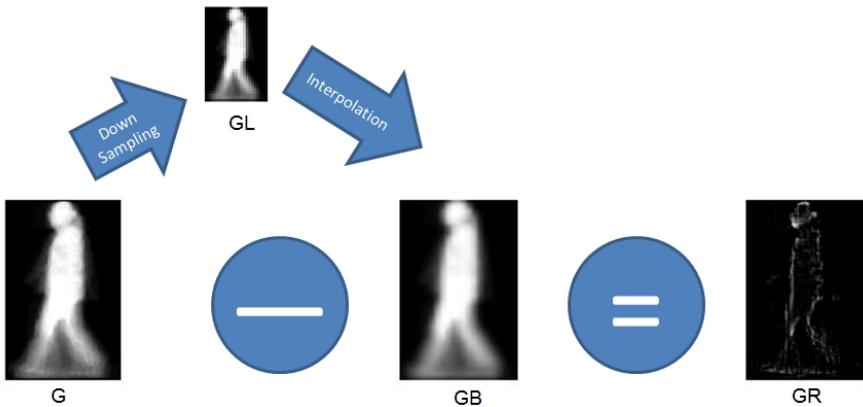


Fig. 2. Illustration of high frequency residual images of gait energy image

of images, high frequency residual information is extracted through computing the difference between high resolution image G and its interpolated one GB . Formally, let high frequency residual image GR be:

$$GR = G - GB \quad (3)$$

The procedure is illustrated as in Fig. 2 . It is worth noting that we perform the same procedure for extracting high frequency residual images for high and low resolution training GEIs and low resolution test GEIs. As shown in Fig. 3 and Fig. 4, $TRHR$ and $TRLR$ denote the high and low resolution residual training GEIs of high resolution training image TRH and corresponding low resolution training image TRL , respectively. $TELR$ means low resolution residual test GEI of low resolution test image TEL .

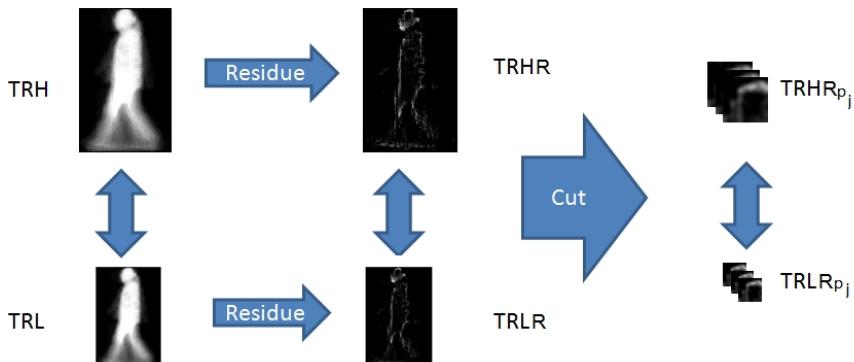


Fig. 3. Generating residual training patch pairs



Fig. 4. Generating high frequency residual low resolution test patches

2.3 Neighbor Embedding

Among learning-based super resolution methods such as the SRNE algorithm[7], images are usually cut into a collection of patches so that the super resolution algorithm can pay more attention to general but not specific images. With the same way, images are cut into patches in the proposed super resolution algorithm. More precisely, if the downsample size is s , high resolution GEIs are cut into $3s \times 3s$ patches $TRHR_{pj}$ with $2 \times s$ -pixel-width overlapped region, and the low resolution counterparts are cut into 3×3 patches $TRLR_{pj}$ with two-pixel-width overlapped region. Furthermore, each low resolution test GEI $TELR$ is also cut into 3×3 patches $TELRp_i$. The cutting procedure can be seen in Fig. 3 and Fig. 4.

After images are divided into image patches, neighbor embedding algorithm [7,8], which assumes that the neighborhood relationship between each patch and its neighbor patches should be preserved when these patches are projected up to high resolution counterparts, is performed for recovering the high resolution ones of residual test patches. Unlike the SRNE algorithm which uses the first-order and second-order gradients to be features [7], the gray-level values of high frequency residual GEI are extracted to form low resolution test patch features \mathbf{x}_t^q and low resolution training patch features \mathbf{x}_s^p . Here the superscript of variable \mathbf{x} means the $*\text{-th}$ patch. The optimal weight vector W_q is thus achieved by minimizing the local reconstruction error for \mathbf{x}_t^q :

$$\varepsilon^q = \|\mathbf{x}_t^q - \sum_{\mathbf{x}_s^p \in \mathcal{N}_q} w_{qp} \mathbf{x}_s^p\|^2 \quad (4)$$

Where \mathcal{N}_q is the neighborhood of \mathbf{x}_t^q in training set X_s , and element w_{qp} of \mathbf{W}_q is the weight for \mathbf{x}_s^p , subject to the constraints $\sum_{\mathbf{x}_s^p \in \mathcal{N}_q} w_{qp} = 1$ and $w_{qp} = 0$

for any $\mathbf{x}_s^p \notin \mathcal{N}_q$ [8]. Actually, the weights are calculated in a simpler way as follows [8]:

$$\mathbf{W}_q = \frac{\mathbf{G}_q^{-1} \mathbf{1}}{\mathbf{1}^T \mathbf{G}_q^{-1} \mathbf{1}}. \quad (5)$$

where

$$\mathbf{G}_q = (\mathbf{x}_t^q \mathbf{1}^T - \mathbf{X})^T (\mathbf{x}_t^q \mathbf{1}^T - \mathbf{X}) \quad (6)$$

Where $\mathbf{1}$ is a column vector of ones and \mathbf{X} is a $D \times K$ (D : dimension; K : neighbor factor) matrix with its columns being the neighbors of \mathbf{x}_t^q .

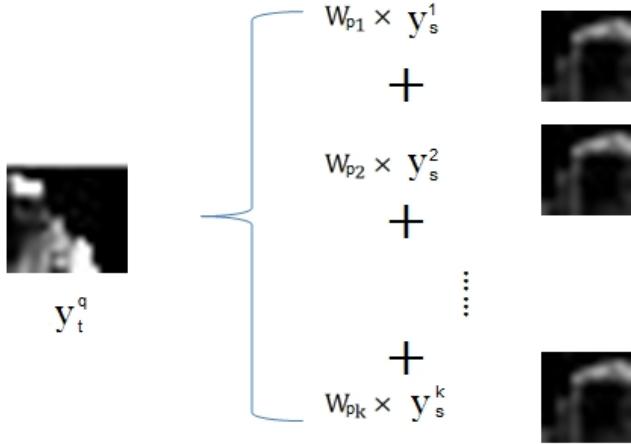


Fig. 5. An example of neighbor embedding algorithm

Once the weight \mathbf{W}_q is calculated, the target patch \mathbf{y}_t^q is computed by

$$\mathbf{y}_t^q = \sum_{\mathbf{x}_s^p \in \mathcal{N}_q} w_{qp} \mathbf{y}_s^p \quad (7)$$

where \mathbf{y}_s^p is the corresponding high resolution residual patch features of \mathbf{x}_s^p . Fig. 5 is an illustration on the recovery of high resolution residual test patch by neighbor embedding algorithm.

2.4 Generating High Resolution Test Images and Recognition

After each high resolution residual test patch is computed, we glue them into a high resolution residual test patch shown in Fig. 6. It is noticeable that for such overlapping region among $TEHR_{Rpi}$ s, an average value for each pixel in the overlapping area is calculated.

Finally, interpolation techniques such as Bilinear interpolation are employed to get an estimate($TEHB$) of the high resolution image for the test low resolution one(TEL). Then add the residue $TEHR$ on $TEHB$ to generate the final result TEH as shown in Fig. 7. From the figure it is not difficult to see that the boundary of TEH is sharper than the interpolated one $TEHB$.

Once the high resolution test images are attained, nearest neighborhood classifier is employed for gait recognition.

3 Experiment

To evaluate the performance of the proposed algorithm, experiments are carried out on the CASIA Gait Database, which consists of 20 individuals and totally



Fig. 6. Generating high resolution residual test image

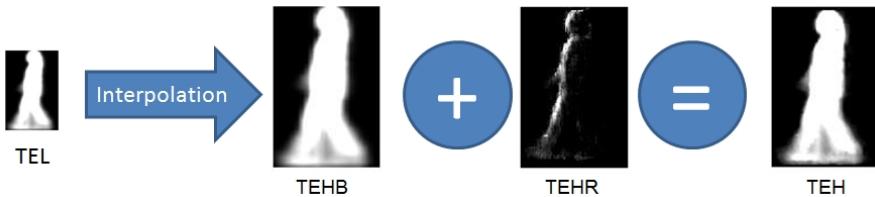


Fig. 7. High resolution test image by super resolution method

240 gait sequences. The gait data are sampled from three view angles, namely, 0° , 45° , 90° between camera and walking direction. For each individual in a specific view angle, gait data are sampled forward and backward to the camera, and each gait sequence is collected twice. Fig.8 shows several examples on this gait database. More details on this database can be seen in [9]. In this experiment, one sequence is used for training, the other one is for testing. It is noticeable that for the proposed algorithm, an additional training set is necessary for learning high resolution ones from low resolution test gait images. Therefore, 6 GEIs of the first subject are used for constructing a training pairs for achieving super resolution.

When distance between camera and objects increases, as we mentioned before, the gait sequences will be changed into the low-resolution counterparts which impair the performance of recognition systems. Therefore, we here attempt to evaluate the effectiveness of the proposed method compared to other interpolation methods and the SRNE algorithm [7] under low resolution case. For this intention, gait data are down-sampled followed by up-sampling via three interpolate techniques, namely, Bicubic interpolation, Bilinear interpolation and

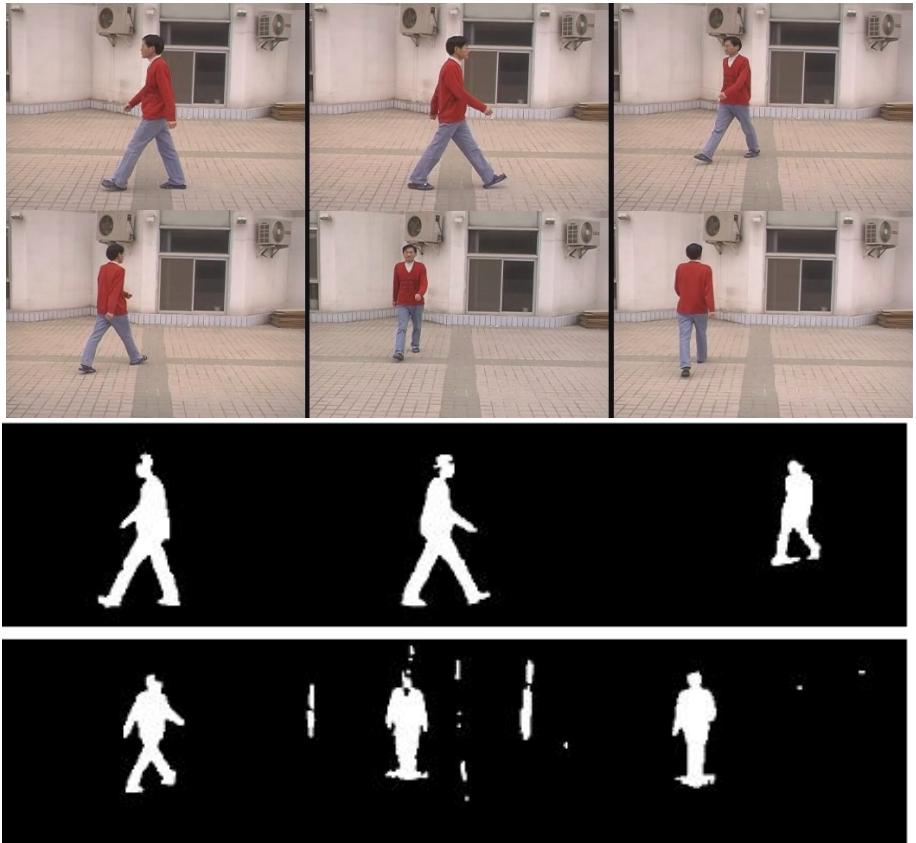


Fig. 8. Original gait data and data after background subtraction

Nearest-neighbor interpolation. The gait data are down-sampled under 5 different scale levels, which are 2-time, 4-time, 8-time, 16-time and 32-time of the original gait frames. Fig.9 shows the example of down-sample gait frames under three mentioned interpolate methods. It can be seen from Fig.9 that gaits will deform a lot under low resolutions, especially when the down-sample rate is 32, gaits lose their original intrinsic structures. Furthermore, the high frequency low resolution residual images are obtained using two different scale levels, namely, 2-time and 4-time. As a result, we have six variants of the proposed SRGR algorithm. Finally, a neighbor factor K of neighbor embedding is set to be 5 without loss of generality. The experimental results can be seen in Tab.1.

From Tab. 1 it can be seen that when down-sampling levels are less than 8, the proposed algorithms make a big improvement in the aspect of accuracy. For example, when down-sampling level is equal to 4, the accuracy of using high frequency super resolution with bilinear-2 is 85.00%, while the accuracy of using bilinear interpolation is 71.67% and the accuracy of the SRNE algorithm is 83.33%. It indicates that the recovery of high frequency loss is indeed helpful

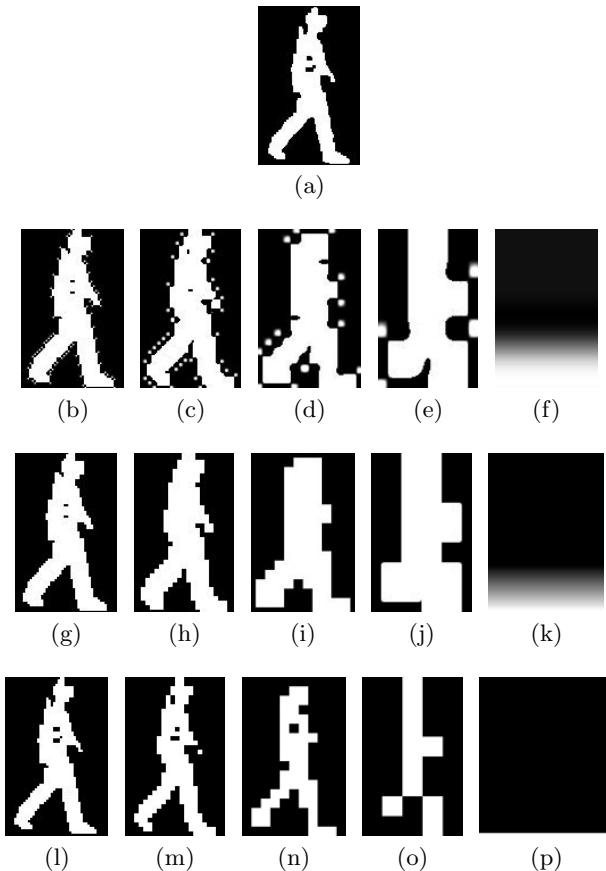


Fig. 9. Example of down-sample gait frame under three different interpolate methods, where (b)-(f) is the gait under 2-time, 4-time, 8-time, 16-time and 32-time down-sample level respectively, using Bicubic interpolation method, and (g)-(k) is using Bilinear interpolation method, while (l)-(p) is the Nearest-neighbor interpolation method

for improving the accuracy of low resolution gait recognition. Furthermore, the recovery of middle and low frequency loss is also important to the refinement of accuracy. From Tab. 1 it is clear that different interpolation methods have their advantages at different down-sampling levels. Thirdly, the down-sampling levels of test image which are used for extracting high frequency residue of low resolution test image have a slight influence on the accuracy of gait recognition. Finally, it can be observed that the super resolution method has its limitation. That is to say, when down-sampling level is equal to 32 in which image information is seriously missed, the super resolution method and other interpolated methods have little help to the improvement of accuracy of gait recognition.

Table 1. Recognition Rates on different level of resolutions using three interpolation methods. In the proposed SR algorithm, “-2” and “-4” denote different downsample level for obtaining high frequency residue of low resolution test image.

Down-sampling levels	2	4	8	16	32	
Bicubic interpolation	84.17%	72.50%	65.33%	39.17%	9.17%	
Bilinear interpolation	84.17%	71.67%	64.17%	31.67%	10.00%	
Nearest-neighbor interpolation	83.33%	58.33%	47.50%	32.50%	10.00%	
SRNE [7]	83.33%	83.33%	66.67%	30.83%	7.50%	
The proposed SR algorithm	bicubic-2	84.17%	83.33%	65.83%	42.50%	12.50%
	bicubic-4	84.17%	85.00%	69.17%	34.17%	14.17%
	bilinear-2	84.17%	85.00%	70.00%	39.17%	11.67%
	bilinear-4	84.17%	85.00%	69.17%	35.00%	15.00%
	nearest-2	95.00%	80.83%	50.83%	29.17%	9.17%
	nearest-4	82.50%	81.67%	60.83%	31.67%	15.00%

4 Conclusions and Future Work

In this paper, we propose a high frequency based super resolution for low resolution gait recognition. First of all, high frequency residue lost in the low resolution gait sequence is extracted by down-sampling followed by up-sampling with interpolation methods. Secondly, high frequency residual of high resolution is obtained with the neighbor embedding algorithm. Finally, with the combination of high frequency residual and interpolation results, the high resolution test gait images are classified. The proposed algorithm has a potential value for gait recognition at longer distance and low resolution case.

It is worth noting that the processing speed in this paper is indeed slower than using the interpolation methods since more steps are involved. In the future, we will consider to solve this problem by using KD tree to optimize the storing and searching process, meanwhile, utilize parallel computing for the searching and training process. Furthermore, the influence of noise to the performance of the proposed algorithm will also be investigated.

Acknowledgement

This work is sponsored by National Natural Science Foundation of China (no. 60635030, no. 60505002), 863 Project (2007AA01Z176), and Huawei Technologies Co., Ltd. The authors would like to acknowledge the invaluable comments and constructive suggestions by anonymous reviewers of PRICAI 2008.

References

1. Wang, L., Tan, T.N., Hu, W.M., Ning, H.Z.: Automatic gait recognition based on statistical shape analysis. *IEEE Transactions on Image Processing* 12(9), 1120–1131 (2003)
2. Kobayashi, T., Otsu, N.: Action and simultaneous multiple-person identification using cubic higher-order local auto-correlation. In: *Proceedings of the 17th International Conference on Pattern Recognition*, Cambridge (2004)
3. Horita, Y., Ito, S., Kaneda, K., Nanri, T., Shimohata, Y., Taura, K., Otake, M., Sato, T., Otsu, N.: High precision gait recognition using a large-scale pc cluster. In: *Proceedings of the 3rd IFIP International Conference on Network and Parallel Computing* (2006)
4. Liu, Z., Sarkar, S.: Simplest representation yet for gait recognition: Averaged silhouette. In: *Proc. IEEE International Conference on Pattern Recognition*, vol. 4, pp. 211–214 (2004)
5. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning Low-Level Vision. *International Journal of Computer Vision* 40(1), 25–47 (2000)
6. Baker, S., Kanade, T.: Limits on Super-Resolution and How to Break Them. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(9), 1167–1183 (2002)
7. Chang, H., Yeung, D.Y., Xiong, Y.: Super-resolution through neighbor embedding. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 275–282 (2004)
8. Roweis, S.T., Saul, L.K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* 290(5500), 2323–2326 (2000)
9. Wang, L., Tan, T.N., Ning, H.Z., Hu, W.M.: Silhouette analysis based gait recognition for human identification. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 12(25), 1505–1518 (2003)

NIIA: Nonparametric Iterative Imputation Algorithm

Shichao Zhang^{1,2,3}, Zhi Jin⁴, and Xiaofeng Zhu¹

¹ College of CS & IT, Guangxi Normal University, PR China

² Faculty of EIT, University of Technology, Sydney, Australia

³ State Key Laboratory for Novel Software Technology, Nanjing University, PR China

⁴ School of EE and CS, Peking University, PR China

zhangsc@mailbox.gxnu.edu.cn, zhijin@math.ac.cn,

xiaozhu@comp.nus.edu.sg

Abstract. Many missing data imputation methods are based on only complete instances (instances without missing values in a dataset) when estimating plausible values for the missing values in the dataset. Actually, the information within incomplete instances (instances with missing values) can also play an important role in missing value imputation. For example, the information has been applied to identifying the neighbors of an instance with missing values in NN (nearest neighbor) imputation, and the class of the instance in clustering-based imputation, where NN and clustering-based imputations are well-known efficient algorithms. Therefore, in this paper we advocate to well utilize the information within incomplete instances when estimating missing values. As an attempt, a simple and efficient nonparametric iterative imputation algorithm, called *NIIA method*, is designed for imputing iteratively missing target values. The NIIA method imputes each missing value several times until the algorithm converges. In the first iteration, all complete instances are used to estimate missing values. The information within incomplete instances is utilized since the second iteration. We conduct intensive experiments for evaluating the proposed approach. Our experimental results show: (1) The utilization of information within incomplete instances is of benefit to capture the distribution of a dataset much better and easier than parametric imputation. (2) NIIA method outperforms the existing methods at the accuracy, and this advantage is clearly highlighted when datasets are with high missing ratio.

1 Introduction

Missing values must be faced in intelligent data analysis, and various solutions for dealing with such issues have been developed in, such as data mining and statistics. Typical strategies for missing data include, for example, omitting all incomplete instances with missing values from the datasets, re-weighting the complete records, and imputing missing values. In real application, missing data imputation is a popular strategy comparing to the others. Missing data imputation is a procedure of “guess” of the missing values based on the complete instances (instances without missing values) in a dataset. Usually used imputation methods include parametric regression and non-parametric regression imputation methods. In the imputation strategy, missing data

treatment is independent of the learning algorithm used. This allows users to select the most suitable imputation method for their learning applications.

From existing imputation algorithms, the information within incomplete instances (instances with missing values) can also play an important role to estimate missing values. For example, the information has been applied to identifying neighbors of an instance with missing values in NN (nearest neighbor) imputation algorithms, and the class of the instance in clustering-based imputation algorithms, where NN and clustering-based imputations are well-known efficient algorithms. Therefore, in this paper we advocate to well utilize the information within incomplete instances when estimating missing values.

This is crucial due to the fact that there are great many incomplete datasets in real world applications that have not enough complete instances for estimating missing values, even if the datasets have only a low missing ratio. For example, the missing ratio in UCI dataset, Bridge, is only 5.56%. This is a low missing ratio in real applications because many datasets in industrial area often reach to 50% or above. However, there are 38 complete instances out of all 108 instances in the dataset with 6 class labels. We impute missing values with existing imputation algorithms, based on the 38 complete instances. Our experimental results show that the imputation frequently generates bias due to the few complete instances, because the size of a large sample should be beyond 30 in statistics. Moreover, there are 6 classes in this dataset. And the maximal number of complete instances in a class is only 11. Therefore, it is difficult to obtain a satisfied classification accuracy based on the few complete instances, even if the most excellent classification algorithm is employed.

As an attempt to utilize the information within incomplete instances, in this paper a simple and efficient nonparametric iterative imputation algorithm, called *NIIA method*, is designed for imputing iteratively missing target values when there is no priori knowledge to the distribution of a dataset. The NIIA method imputes each missing value several times until the algorithm converges. In the first iteration, all complete instances are used to estimate missing values. The information within incomplete instances is used since the second iteration. Our experimental results (see Section 4) show: (1) The information within incomplete instances is of benefit to capture the distribution of a dataset much better and easier than parametric imputation. (2) NIIA method outperforms the existing methods at the accuracy, and this advantage is clearly highlighted when datasets are with high missing ratio.

In the reminder parts, we recall related work in Section 2. Section 3 presents our NIIA method that is a kernel-based approach. We illustrate the efficiency of the proposed method with various kinds of experiments in Section 4. Finally, we conclude our work and put forward the future work in Section 5.

2 Related Works

There are at least three different ways of dealing with missing values based on [4]: single imputation, multiple imputation, and iterative procedure.

Single imputation strategies provide a single estimate for each missing data value. Many methods for imputing missing values are single imputation methods, such as, C4.5 algorithm, kNN method, and so on. We can partition single imputation methods

into parametric methods and nonparametric ones. The parametric regression imputation methods (such as, linear regression imputation method, nonlinear imputation method) are superior if a dataset can be adequately modeled parametrically, or if users can correctly specify the parametric forms for the dataset. If the model is misspecified (in fact, it is usually impossible for us to know the distribution of the real dataset), the estimations of parametric methods may be highly biased and optimal control factor settings may be miscalculated. Moreover, we must expense much time to model the real distribution even if we can know the real distribute of the datasets. Non-parametric imputation method offers a nice alternative if users have no idea on the actual distribution of a dataset because the method can provide superior fits by capturing structure in datasets (a mis-specified parametric model cannot). In real application, we usually have no priori knowledge on our datasets, so in this paper, we will introduce an algorithm NIIA to impute iterative missing target values under the assumption of nonparametric model.

A disadvantage of single imputation strategies is that they tend to artificially reduce the variability of characterizations of the imputed dataset. Moreover, single-imputation cannot provide valid standard errors and confidence intervals, since it ignores the uncertainty implicit in the fact that the imputed values are not the actual values. The alternatives are to fill in the missing values with multiple imputation methods (e.g., Multiple Imputation (MI) [3]) and iterative imputation methods (EM algorithm). Multiple imputation strategies generate several (typically < 20) different imputed datasets and subject to be performed the same analysis, giving a set of results from which typical (e.g., mean) characterizations and variability estimates (e.g., standard deviations). In multivariate analysis, MI methods provide good estimations of the sample standard errors. However, data must be missed at random in order to generate a general-purpose imputation. In contrast, iterative approaches can be better developed for missing data since it can utilize all useful information including the instances with missing values [6]. That can receive significant performance in the datasets with high missing ratio. The well-known of these methods is the Expectation-Maximization (EM) algorithm for parametric model. [2, 6] present an EM-style nonparametric iterative imputation model embedded with kNN algorithm to impute missing attribute values.

In this paper, we present an iterative imputation method for dealing with missing target values. At first, we impute missing values with general methods (e.g., mean based regression method) in order to utilize any possible information in a dataset, then we impute each missing values iteratively based on kernel regression imputation method until the algorithm converges. In this process, we perform nonparametric iterative imputation algorithm (NIIA) while no supplement on the datasets. Different from the existing parametric methods, the proposed method can be easily applied to real application because we usually have no supplement knowledge on our datasets. Different from single imputation method and multiple imputation method, our iterative method (NIIA) utilizes the information within incomplete instances for improving imputation performance. In real application, most of datasets contain missing values with high missing ratio. If we give up the information in incomplete instances, there is not enough information for us to impute missing values based on only the complete instances. And this will result in low imputation performance. Therefore,

our iterative imputation method is reasonable. Different from the existing nonparametric iterative imputation methods which are designed to focus on missing attribute values, we employ kernel regression method which is popular in statistics and is supported by widely theorem to impute missing **target** values rather than kNN method in [2, 6].

3 NIIA Algorithm

This section presents our NIIA method, in which the information within incomplete instances is utilized when estimating missing values. This information assists in capturing the distribution of a dataset. In the first imputation iteration of NIIA method, we employ a certain existing method, which fits for statistical proof (such as, mean/mode method), to estimate plausible values for all missing values in a dataset. Since the second imputation iteration, imputation is based on all instances in the dataset, where missing values in incomplete instances have been replaced with the plausible values estimated in last iteration.

Generally, we denote missing value as $MV_i, i = 1, \dots, n$ (n is the number of missing values) corresponding to imputed missing values denoted as $\hat{MV}_i^j, i = 1, \dots, n, j = 1, \dots, t$ (j is the imputation time), all missing values MV_i are imputed as \hat{MV}_i^1 with the first imputation. Since the second imputation, the observed information will include $\hat{MV}_i^{j-1}, i = 1, \dots, k-1, k+1, \dots, n, j = 2, \dots, t-1$ while we want to impute a missing value $\hat{MV}_k^j, k \neq i, j = 2, \dots, t$, the imputation process will continue till algorithms reach to approximate convergence. Meanwhile, since the second imputation, we will employ kernel regression method for imputing missing values under nonparametric model. The pseudo codes of algorithm NIIA is presented as follows:

```

//the first imputation, details in section 3.1
FOR each  $MV_i$  in Y
   $\hat{MV}_i^1 = \text{mode} (S_r \text{ in } Y)$            // if Y is discrete variable
   $\hat{MV}_i^1 = \text{mean} (S_r \text{ in } Y)$         // if Y is continuous ones
END FOR

//t-th iterative imputation(t>1), details in section 3.1
t=1;
REPEAT
  t++;
  FOR each missing value  $MV_i$  in Y
    If  $MV_i$  is current imputed missing value
       $MV_i = \hat{MV}_i^t, p \in S_m, p = 1, \dots, m, p \neq i$  // if Y is continuous variable
       $MV_i = \begin{cases} 0 & \text{if } \hat{MV}_i^t < \chi, \\ 1 & \text{if } \hat{MV}_i^t \geq \chi, \end{cases} \hat{MV}_i^t, p \in S_m, p = 1, \dots, m, p \neq i$  // if Y is discrete variable
    Else

```

```

 $MV_i = \hat{MV}_i^{t-1}, p \in S_m, p = 1, \dots, m, p \neq i$ 
END FOR
UNTIL //finishing iterative imputation, details in section 3.3
 $\frac{M_l}{M_{l+1}} \rightarrow 1$ , and  $\frac{V_l}{V_{l+1}} \leq \varepsilon$ 
3.0 //output the imputation times and imputation results, details in section 3.2
OUTPUT
t; // t is the iterative times
Completed dataset;

```

The Pseudo-code of NIIA Algorithm

3.1 The First Imputation Iteration

There exist many methods to fill in missing values in the first imputation including any single imputation methods, such as, C4.5 algorithm, kNN algorithm, and so on. In [6], authors compute mean (or the mode if the attribute is discrete) to impute missing values in the first time. They think that the method is a popular and feasible imputation method in data mining and statistics. Meanwhile, they also believe to impute with the mean (or mode) is valid if and only if the dataset is chosen from a population with a normal distribution. However, in real world application, we cannot know the real distribution of the dataset in advance. So running the extra iteration imputations to improve imputation performance is reasonable based on the first imputation for dealing with the missing values. Caruana in [2] thinks the first step, which imputes each missing value with the mean/mode values calculated from cases that are not missing that value, will cause cases missing many values to appear to be artificially close to each other. Hence, the author proposes a new method for avoiding this case. And the paper demonstrates this subtlety is not critical for the proper behavior of the method, but does speed convergence on datasets that have many missing values. However, the method in [2] is designed to impute missing attribute values rather than missing target values. In our paper, we will employ mean/mode method to impute missing values in the first imputation iteration.

3.2 Successive Imputation Iterations

In Section 2, we analyzed nonparametric techniques will be utilized to impute missing while there are no any prior knowledge for the current dataset. There exist many methods on kernel methods, such as, deterministic kernel method in [5], random kernel imputation method in [7, 8].

Assuming multi-dimension vectors without missing values are denoted as X_i and one dimension vector Y_i with missing values. Let S_r and S_m denote the sets of observed and missing respectively, n is the number of instances in the dataset, $r = \sum_{i=1}^n \delta_i$, $m = n - r$ ($\delta_i = 0$ if Y_i is missing, otherwise $\delta_i = 1$). Let \hat{Y}_i , $i \in S_m$ be the imputed values and can be imputed with kernel imputation methods in [7, 8] as:

$$\hat{Y}_i = \hat{m}_n(X_i) + \varepsilon * i \in S_m \quad (1)$$

Where $\{\varepsilon^*\}$ is a simple random sample of size m with replacement from $\{Y_j - \hat{m}_n(X_i), j \in S_r\}$, and

$$\hat{m}_n(x) = \frac{\sum_{i=1}^n \delta_i Y_i K(\frac{x - X_i}{h})}{\sum_{i=1}^n \delta_i K(\frac{x - X_i}{h}) + n^{-2}}$$

$\hat{m}_n(x)$ is based on the completely observed pairs (X_i, Y_i) . Where h is a bandwidth sequence that decreases toward 0 as the sample size n increases toward ∞ ; the term n^{-2} is introduced to avoid the denominator to zero. $K(\frac{x - X_i}{h})$ is a symmetric probability density function and claimed kernel function. There are some widely used kernel functions in nonparametric inference, i.e. Gaussian kernel (standard normal density function) and uniform kernel. In practice, there is not any significant difference using these kernel functions. In the algorithm NIIA, we use the Gaussian kernel in our experiments.

In NIIA algorithm, we will revise deterministic kernel method in [5] rather than random kernel imputation methods in [7, 8] as the variation of the value of ε^* is difficult to be controlled in iterative method. Moreover, our iterative method can reach to the aim for setting ε^* which is designed to avoid large variation of the imputation values. Hence, we define the t -th imputation values of the i -th missing value as \hat{Y}_i^t :

$$\hat{Y}_i^t = \hat{m}_t(X_i) \quad (2)$$

Where t is the number of iterative imputation, $\hat{m}_t(x)$ denotes kernel estimator for $m_t(x)$ based on the completely observed pairs (X_i^t, Y_i^t) :

$$\hat{m}_t(x) = \frac{n^{-1} \sum_{i=1}^n \delta_i Y_i^t K(\frac{x^t - X_i^t}{h})}{n^{-1} \sum_{i=1}^n \delta_i K(\frac{x^t - X_i^t}{h}) + n^{-2}}$$

Where $Y_i^t = \begin{cases} Y_i, & \text{if } \delta_i = 0 \text{ or } i = 1, \dots, r \\ \hat{Y}_{i-1}^{t-1}, & \text{if } \delta_i = 1 \text{ or } i = r+1, \dots, n \end{cases}$

In particular, $\hat{Y}_i^1 = \frac{1}{r} \sum_{i=1}^r Y_i$, coming from the result of the first imputation. The selection of kernel function $K(\frac{x^t - X_i^t}{h})$ is same the kernel function in deterministic kernel method. Hence, in algorithm NIIA, since the second imputation, we use Equation 2 to impute missing target values for continuous missing target attribute till the algorithm converges.

In fact, the imputed values based on Equation 2 always is continuous values, and our NIIA algorithm can also impute discrete missing target attribute which is presented in pseudo of NIIA algorithm 2.0. In our paper, we consider the case with two classes and the reader can extend our method to the case with multiple classes. In NIIA algorithm, instances are defined as belonging to class 0 if $\hat{M}V_i^t < \chi$, and class 1 otherwise. The actual value of the class for each incomplete instance x_i is denoted by MC_{x_i} . The new class assignment based on the imputed class is denoted by $\hat{MC}_{x_i}^t$ in this imputation to stress the dependence of the classification on x_i . More specifically, the imputed value $\hat{m}_t(x) \in \mathbb{R}$ is transformed into a (binary) class $MC_{x_i}^t \in \{0, 1\} \forall x_i \in D$ based on the rule specified in NIIA algorithm. χ is specified by the user of the technique and in many applications is set so that $|\{x_i | MC_{x_i}^t = 1\}| = |\{x_i | \hat{MC}_{x_i}^t = 1\}|$ (i.e., the number of class 1 instances before and after the application of our technique is the same). This rule for class assignment is the most natural choice and is the one primarily considered in our case studies, although the user of the technique may explore different choices near this preferred cutoff point. Based on this rule, our proposed algorithm NIIA can also be used to impute discrete missing target values.

Finally, we can output the final imputation result after algorithm converged. Note that, the imputation times is $(t+1)$ rather than $(t+2)$ times even if the iterative procedure is performed $(t+1)$ times and the first iteration will be added. That is because the last imputation does not generate imputation result and only judge the fact whether the imputation reaches to convergence.

3.3 Algorithm Convergence and Complexity

An important practical issue concerning iterative imputation method is to determine at which point additional iterations have no meaningful effect on imputed values, i.e., how to judge the convergence of the algorithm. Literatures [2] and [8] conclude that the average distance that missing attribute values move in successive iterations drops to zero, that no missing values have changed and that the method has converged in nonparametric model. Here, we outline a strategy for the stopping criterion for our algorithms. With t imputation times, assuming mean and variance of three successive imputations are M_l, M_{l+1}, M_{l+2} , and V_l, V_{l+1}, V_{l+2} , ($1 < l < t-2$) respectively. If

$$\frac{M_l}{M_{l+1}} \rightarrow 1, \text{ and } \frac{V_l}{V_{l+1}} \leq \varepsilon$$

That can be inferred that there is little change in imputations between the last and the former imputation, and the algorithm can be stopped for imputing without substantial impact on the resulting inferences. Different from the converged condition in existing algorithms, we summarize our stopping strategy using terminology such as ‘satisfying a convergence diagnostic’ rather than ‘achieving convergence’ to clarify that convergence is an elusive concept with iterative imputation.

While the complexity of kernel method is $O(mn^2)$, where n is the number of instances of the dataset, m is the number of attributes, so the algorithm complexity of both NIIA and SIIA is $O(kmn^2)$ (k is the number of iteration imputation).

4 Experiments Analyses

In order to show the effectiveness of the proposed methods, extensive experiments are done on real dataset with VC++ programming by using a DELL Workstation PWS650 with 2G main memory, 2.6G CPU, and WINDOWS 2000. We compare the performance of NIIA with the existing iterative method kNN in [8] as well as single imputation methods for imputing continuous missing target attribute in terms of imputation accuracy with RMSE in Section 4.1, and we present the performance of NIIA algorithm with existing methods for imputing discrete missing target attribute in terms of classification accuracy in real dataset in Section 4.2.

4.1 Experimental Study on Continuous Missing Target Attribute

At first, we design different algorithms to impute target missing values, such as, our proposed algorithm NIIA, kNN algorithm in [8], and the two single imputation methods (deterministic method for single imputation in [7], DS for shorted, random method for single imputation in [9,10], RS for shorted). We use RMSE to assess the predictive ability after the algorithm has converged for iterative imputation methods or the missing values are imputed for single imputation methods:

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (e_i - \tilde{e}_i)^2}$$

Where e_i is the original attribute value; \tilde{e}_i is the estimated attribute value, and m is the total number of predictions. The larger the value of the RMSE, the less accurate is the prediction.

Two datasets from UCI in [1], Housing and Auto-mpg are used in our experiment. Housing contains 506 instances and 10 continuous attributes. Auto-mpg contains 398 instances and 8 attributes. All the two datasets have no missing values, and we did not select intentionally those datasets that originally come with missing values, because if they contain missing values, we could not know the real values for the missing values. So we adopt the complete datasets and miss data at random to systematically study the performance of the proposed method, the percentage of missing values (missing ratio for short) was fixed at 10%, 20%, and 40% respectively for each dataset.

Figures 1, 3, 5 and 2, 4, 6 present the values of RMSE in dataset Housing and Auto-mpg with missing ration 10%, 20% and 40% respectively. The experimental results show:

NIIA algorithm can converge in all cases in the experiments. For example, in dataset Housing, NIIA algorithm converges with the imputation times 5, 8 and 9 at different missing ratio 10%, 20% and 40% respectively. Corresponding to the dataset Auto-mpg, the number is 6, 8 and 10. The higher missing ratio, the more imputation time is. It is obviously as more missing values must be imputed and the imputation performance will be improved slower than the case with lower missing ratio. The results present our proposed iterative imputation method is effective.

Comparing iterative imputation methods (such as, kNN and NIIA) with single imputation methods (such as, DS and RS), in the first imputation, iterative imputation methods receive lower imputation performance than single imputation methods. It is

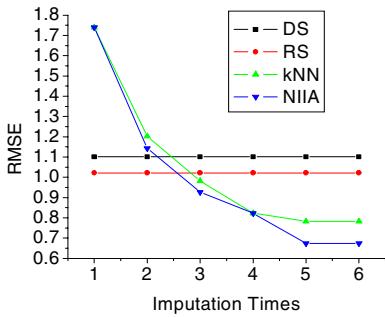


Fig. 1.

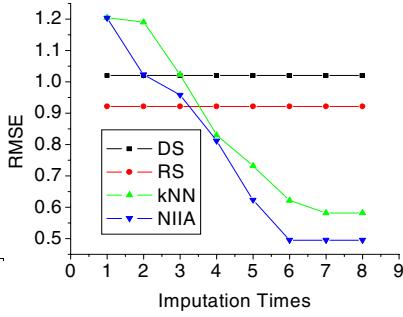


Fig. 2.

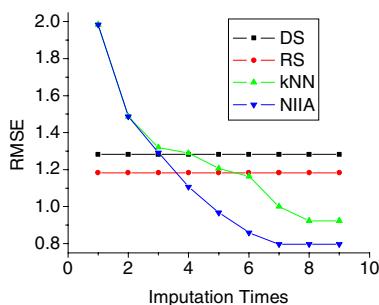


Fig. 3.

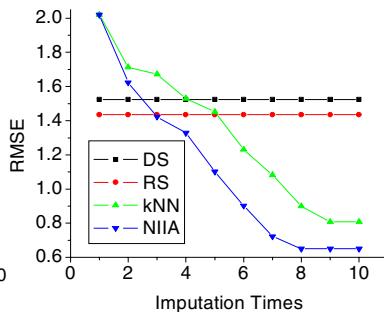


Fig. 4.

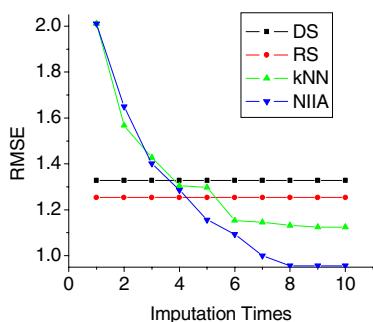


Fig. 5.

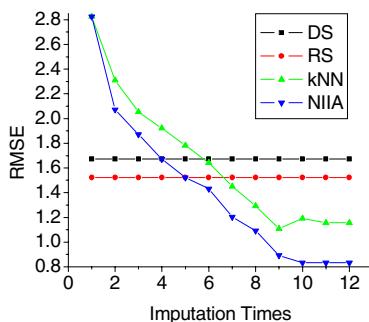


Fig. 6.

the fact since iterative imputation methods employ the most popular method (i.e., mean/mode method) to impute missing values. However, since then, the situation varies, that is, the iterative algorithms outperform single methods. That can imply the facts. Firstly, it is reasonable for us to employ kernel regression imputation method to impute iterative missing values rather than mean/mode which is the simplest imputation method since the second imputation. As to select mean/mode method as the first imputation method, it is because the method is simple and presents low computation complexity, and the most highlight thing in iterative imputation method is the successive imputation can incremental improve imputation performance. For instance, in the

former imputation times after the first imputation, the values of RMSE in NIIA or kNN algorithm are worse than the algorithm DS or RS, or the values of RMSE in NIIA or kNN algorithm are a little better than DS and RS algorithm. However, NIIA or kNN algorithm is better than DS or RS algorithm while the algorithm nearly converges. That presents iterative imputation methods can improve imputation performance by successive imputation. Secondly, experimental results from Figure 1 to 6 show iterative imputation methods outperform than single imputation methods. Furthermore, the difference between iterative imputation methods and single ones present the maximal values while the missing ratio reach to 40%. That is because, the higher missing ratio, the more obvious advantages the iterative imputation methods are.

Comparing kNN algorithm with NIIA method, NIIA is better than kNN method in [8] in terms of the values of RMSE or the imputation times after the algorithm converges. We can find they contain the same performance in the first imputation because the two methods employ mean/mode method for the first imputation. Since the second imputation, kNN algorithm presents a little profit than our NIIA, for example, in dataset Housing, the values of RMSE of kNN algorithm is 1.486 at missing ratio 20% in the second imputation, and the corresponding value in NIIA is 1.487. In the left iteration, the results of NIIA algorithm are better than the kNN, in particular in the high missing ratio, such as, at missing ration 40%.

4.2 Experimental Study on Discrete Missing Target Attribute

The UCI datasets ‘Abalone’, ‘Vowel’, and ‘CMC’, in which class attribute is discrete, are applied to compare the performances in terms of classification accuracy on the above four methods.

We need to assess the performance of these prediction procedures. We will evaluate their Classification Accuracy (**CA**), which is defined as:

$$CA = \frac{1}{n} \sum_{i=1}^n I(IC_i, RC_i)$$

Where t is the number of missing values, n is the number of instances in the dataset. The indicator function $I(x, y) = 1$ if $x = y$; otherwise it is 0. IC_i and RC_i are the imputation and real class label for the i -th missing value respectively. Obviously, the larger value of CA, the more efficient is the algorithm.

Table 1 shows the results of classification accuracy after iterative imputation algorithms (such as, kNN and NIIA) reach to convergence or the result for single imputation methods (i.e., DS and RS) at the different missing ratio 10%, 20% and 40% respectively. Table 2 presents the results of iterative times after these two iterative algorithms have been terminated at the different missing ratio 10%, 20% and 40% respectively on datasets ‘Abalone’, ‘Vowel’, and ‘CMC’. Due to lack of space, we do not present the details as Section 4.1 with pictures. However, similar to the results in Section 4.1, the classification accuracy of iterative algorithms for imputing discrete missing values are better than the results of single imputation in different missing ratio, in particular, in the case with high missing ratio. For example, the minimal difference between iterative algorithms and single algorithms are 0.029 (kNN vs. RS in Abalone), 0.043 (kNN vs. RS in Vowel), 0.033 (kNN vs. RS in CMC), and the

maximum are 0.042 (NIIA vs. DS in abalone), 0.081 (NIIA vs. DS in Vowel), and 0.11 (NIIA vs. DS in CMC) respectively while missing ratio is 40%. However, the corresponding minimal values only are 0.018, 0.024, and 0.021 at missing ration 10%, 0.01, 0.011, and 0.023 at missing ration 20% respectively. And the corresponding maximum are 0.065, 0.042, and 0.048 at missing ratio 10%, 0.08, 0.059, and 0.068 at missing ration 20%. From table 2, we can find the two iterative algorithms (such as, kNN and NIIA) can reach to convergence with the similar imputation times while the missing ratio is low, such as, 10% or 20%. However, while the missing ratio is high, for example 40% in our experiments, the difference of imputation times begin to vary, that is, kNN algorithm converges slower than NIIA algorithm. Combining Tables 1 and 2, we can make a conclusion that NIIA algorithm outperform kNN algorithm on both classification accuracy and imputation times for imputing discrete missing target values.

Table 1. Classification Accuracy: iterative Algorithms reach to converge vs. single imputation algorithms

	Abalone			Vowel			CMC		
	10%	20%	40%	10%	20%	40%	10%	20%	40%
NIIA	0.781	0.775	0.711	0.855	0.83	0.807	0.873	0.843	0.827
kNN	0.773	0.742	0.672	0.842	0.801	0.773	0.859	0.821	0.792
RS	0.755	0.732	0.643	0.818	0.788	0.73	0.838	0.798	0.739
DS	0.716	0.695	0.639	0.813	0.771	0.726	0.825	0.775	0.717

Table 2. Imputation Times after the Algorithms converge

	Abalone			Vowel			CMC		
	10%	20%	40%	10%	20%	40%	10%	20%	40%
NIIA	6	8	10	8	10	13	8	9	12
kNN	7	10	12	8	11	17	8	10	15

5 Conclusion and Future Work

By experience from developed missing data imputation techniques, in this paper we have advocated to well utilize the information within incomplete instances in existing imputation algorithms. This is because that there are great many incomplete datasets in real world applications that have not enough complete instances for estimating missing values. As an attempt to utilize the information within missing data, a nonparametric iterative imputation algorithm, NIIA method, has been designed for imputing iteratively missing target values when there is no priori knowledge to the distribution of a dataset. The NIIA method imputes each missing value several times until the algorithm converges. The information within incomplete instances is used since the second iteration. We have conducted intensive experiments for evaluating the proposed approach. The experimental results have demonstrated that our NIIA method outperforms the existing methods in term of RMSE (for continuous missing

target attribute), or classification accuracy (for discrete missing target attribute) and the convergence times at different missing ratios in different real datasets. In particular, they have illustrated that the utilization of information within incomplete instances is of benefit to capture the distribution of a dataset much better and easier than parametric imputation. The experiments have also shown that our iterative imputation methods are totally better than single imputation methods.

In our future work, we will focus on the study how to more effective estimate and impute missing values with semi-parametric model.

Acknowledgement

This work was supported in part by the Australian Research Council (ARC) under grant DP0667060, the Nature Science Foundation (NSF) of China under Major Research Program 60496327, China NSF under grant 90718020 and Distinguished Young Scholars Program 60625204, the China 973 Program under grant 2008CB317108, the Research Program of China Ministry of Personnel for Overseas-Return High-level Talents, and the Guangxi NSF (Key) grants.

References

1. Blake, C., Merz, C.: UCI Repository of machine learning databases (1998)
2. Caruana, R.: A Non-parametric EM-style algorithm for Imputing Missing Value. *Artificial Intelligence and Statistics* (January 2001)
3. Little, R., Rubin, D.: Statistical Analysis with Missing Data, 2nd edn. John Wiley and Sons, New York (2002)
4. Pearson, P.K.: Mining imperfect data: dealing with contamination and incomplete records. SIAM, Philadelphia (2005)
5. Wang, Q., Rao, J.N.K.: Empirical likelihood-based inference under imputation for missing response data. *Ann. Statist.* 30, 896–924 (2002)
6. Zhang, C.Q., et al.: Efficient Imputation Method for Missing Values. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 1080–1087. Springer, Heidelberg (2007)
7. Qin, Y.S., et al.: Optimized parameters for missing data imputation. In: Yang, Q., Webb, G. (eds.) PRICAI 2006. LNCS (LNAI), vol. 4099, pp. 1010–1016. Springer, Heidelberg (2006)
8. Qin, Y.S., et al.: POP Algorithm: Kernel-Based Imputation to Treat Missing Values in Knowledge Discovery from Databases. *Expert Systems with Applications* (accepted, 2008), <http://dx.doi.org/10.1016/j.eswa.2008.01.059>

Mining Multidimensional Data through Element Oriented Analysis

Yihao Zhang¹, Mehmet A. Orgun¹, Weiqiang Lin², and Rohan Baxter²

¹ Department of Computing, I.C.S., Macquarie University Sydney, NSW 2109, Australia
{yihao,mehmet}@ics.mq.edu.au

² Australian Taxation Office, Canberra ACT 2601, Australia
{wei.lin,rohan.baxter}@ato.gov.au

Abstract. Mining multidimensional data has two major concerns. One is how to select the most salient attributes and another one is how to guarantee the precision of mining results. This paper introduces a novel approach to mine multidimensional data through Element Oriented Analysis (EOA). In our approach, each observational data is considered to be comprised by two essential elements, the structure elements and the numerical elements. EOA firstly targets Structural Element Pattern (SEP) that is an aggregation of the structural elements. The successful SEP will be referenced by a Numerical Element Pattern (NEP) that is composed of the numerical elements. Given the results from both SEP and NEP, a global discriminant will be created for the efficient evaluation by the consequent data. In this paper, publicly available Turkish bank records are analyzed in an experiment that demonstrates the practical utility of our approach.

Keywords: Element Oriented Analysis, Multidimensional Data.

1 Introduction

The essential purpose of multidimensional data mining is to group useful information and uncover hidden knowledge from variant dimensions. In the last decade, the techniques of multidimensional data mining have been developed in two main streams, classification and clustering [13]. The classification approach sets the related labels and uses these labels to influence the data mining process. Hence, many utilized algorithms have preferential target classes. Those algorithms include decision tree based Iterative Dichotomize 3 (ID3) and its extension C4.5 [11], and Chi-square Automatic Interaction Detective (CHAID) [10], and statistical based Representative Bagging [2] and AdaBoost [1]. However, the effective performance of classification highly depends on data labels (classifiers), which is partially non-robust to multidimensional data. For example, the classification approach is liable to mislead in predicting corporate collapse if observational financial data exhibits significant diversity of its dimensions over a variant temporal domain. The predefined classifiers may not easily recognize changeable data space, and these spaces are difficult to frequently reset during the mining process. The usual solution is to enlarge the range of classifiers or increase the number of classifiers.

Clustering is another important attempt to solve the curse of dimensionality. The major goal of clustering is to group data points into a set of clusters so that a data point has more similarity with the data points within the cluster than those outside of the cluster, such as K-Means clustering, which is to divide a set of data into k different groups according to some criterion of similarity [12]. Since the clustering approach starts without predefined objective functions, it has been widely applied to multidimensional data. However, clustering approach has to face how to guarantee the precision of mining performance while it discards the limitation of classification's label. Over-random searching and evaluating augment the error rate of mining results, especially in mining multidimensional data. For example, suppose that there exists a multidimensional dataset $D = \{f_1, f_2, \dots, f_m\}$ where $f_i (1 \leq i \leq m)$ is a dimension of D and our mining goal is to divide D into n sound groups. Due to the fact that assigning data into different groups in multidimensional data extraordinarily depends on the covariance matrix, this process might be exceedingly complicated to establish the number of groups. In addition, it is difficult to guarantee the precision of data grouping.

In this paper, we provide a new approach of mining multidimensional data, called Element Oriented Analysis (EOA) which alters analytical orientation from data points to internal elements. According to EOA, we restructure traditional mining framework from a single level to local and global levels. In the local level, Structural Element Pattern (SEP) and Numerical Element Pattern (NEP) are discovered to define rough structural and accurate relationships, respectively. Then the integrated results are referenced by the global level and a set of general discriminant scores are created to classify the entire dataset. The goal of our approach is to make multidimensional data mining as good as possible in both flexibility and accuracy.

The rest of the paper is organized as follows. Section 2 introduces some basic concepts of the Element Oriented (EO) theory. In section 3, we present the detailed implementation of EOA. In section 4 we apply our approach in an experiment based on Turkish Bank Data to identify the effect of EOA and discuss the results of our experiments. Last section discusses related work and concludes the paper with a brief summary of our contributions.

2 Element Oriented (EO) Theory

Element Oriented (EO) theory is derived from Lin et al [8] and Zhang et al [14] and considered as a novel theory differing from the existing ones which only analyze pure data values directly. In EO, the observational data, either from sample or population is recognized as several elements. To better understand EO theory, we first give four basic definitions regarding elements and element patterns in dataset.

Definition 1. Let $D = \{d_1, d_2 \dots d_m\}$ be a dataset and $d_i (1 \leq i \leq m)$ be the i^{th} value in the dataset. If all d_i can be divided into k components, i.e. $d_i = e_{i1} \otimes e_{i2}, \dots, e_{ik}$, $(1 \leq i \leq m)$ where \otimes represents a specified conjunctive relationship between different components, then any $e_{il} (1 \leq l \leq k, 1 \leq i \leq m)$ is an element of data point d_i and d_i is an Element Oriented data.

Since the element addresses observational data from different point of views, the dataset can be described as the element patterns.

Definition 2. Let $D = \{d_1, d_2 \dots d_m\}$ be a dataset and d_i ($1 \leq i \leq m$) be the i^{th} value in the dataset. If for all i ($1 \leq i \leq m$) $d_i = e_{i1} \otimes e_{i2}, \dots, e_{i(k-1)} \otimes e_{ik}$, the dataset can be represented as $D = \{e_{11} \otimes \dots \otimes e_{1k}, e_{21} \otimes \dots \otimes e_{2k}, \dots, e_{m1} \otimes \dots \otimes e_{mk}\} = \{(e_{11}, \dots, e_{m1}) \otimes (e_{12}, \dots, e_{m2}), \dots, (e_{1k}, \dots, e_{mk})\}$. If for all $\{e_{1l}, \dots, e_{ml}\}$ ($1 \leq l \leq k$) is represented by E_l ($1 \leq l \leq k$), then the dataset is represented as $D = \{E_1 \otimes E_2 \otimes \dots \otimes E_k\}$ and E_l ($1 \leq l \leq k$) is an element pattern of dataset D .

Definition 3. An analysis is named as Element Oriented Analysis (EOA) if the analysis is aimed at Element Oriented data and the analytic architecture is structured through finding different element patterns.

Logically, a common Element Oriented data might be decomposed into two elements. On the one hand, an element which contains the general data information, such as the structure of data, change tendency of data or qualitative relationship within the data, can be defined as the structural element. On the other hand, we might define another numerical element to describe the quantitative relationships within all structural elements. These two elements discover the information from two different points of view. Hence, the integrated results of these two elements can be better used for data analysis.

Definition 4. Let $D = \{d_1, d_2 \dots d_m\}$ be a dataset and d_i ($1 \leq i \leq m$) be the i^{th} value in the dataset. Suppose that for all $d_i = (s_i) \otimes (n_i)$ ($1 \leq i \leq m$), where s_i is named the structure element and n_i is named the numerical element. Then there exist two patterns:

1. Structure Element Pattern (SEP) aggregates all structural elements of the dataset.
2. Numerical Element Pattern (NEP) aggregates all structural elements of the dataset.

Then the dataset can be represented as function (1)

$$D = \{S \otimes N\} \quad (1)$$

where $S = (s_1, s_2, \dots, s_m)$ is a SEP corresponding to every data point with element s_l ($1 \leq l \leq m$) and $N = (n_1, n_2, \dots, n_m)$ is NEP corresponding to every data point with element n_l ($1 \leq l \leq m$).

We can build a new analytical framework based on Definition 4. On the one hand, we attempt to uncover the conditional dependence among the data through the SEP. After this initial detection, the SEP can be described as the overall structure or regularity for the dataset. The NEP on the other hand refers to the deeper detection of accurate relationships from given SEP and presents them by a polynomial form.

3 Element Oriented Analysis (EOA) in Multidimensional Data

Element Oriented Analysis (EOA) method then applies EO theory into multilevel analysis area. Three different analytical levels, the Local Structural Level (LSL), the Local Numerical Level (LNL) and the Global Level (GL) are constructed by the element patterns.

In LSL, the specified mapping M is used to transfer the data set $D = \{d_1, d_2, \dots, d_m\}$ to the SEP $S = \{s_1, s_2, \dots, s_m\}$. An effective SEP might save

computing resource by simplifying the original dataset. In addition, we can obtain an orientation or a hint in the further knowledge exploration stage since the SEP contains a general tendency or a rough regularity from the original dataset.

The task in LNL is to estimate a numerical function based on the SEP from LSL and calculate the accurate coefficients. To perform this task, a SEP $S = \{s_1, s_2, \dots, s_m\}$ is assigned into a NEP $N = \{n_1, \dots, n_k\}$ through a local polynomial regression function.

The results from LNL are delivered to GL. The distinct global values are identified in this level, which might be employed for different purposes. For example, it might be regarded as a discriminant score to accurately form the original data into variant groups.

The framework of EOA is therefore designed and shown in Fig.1 below.

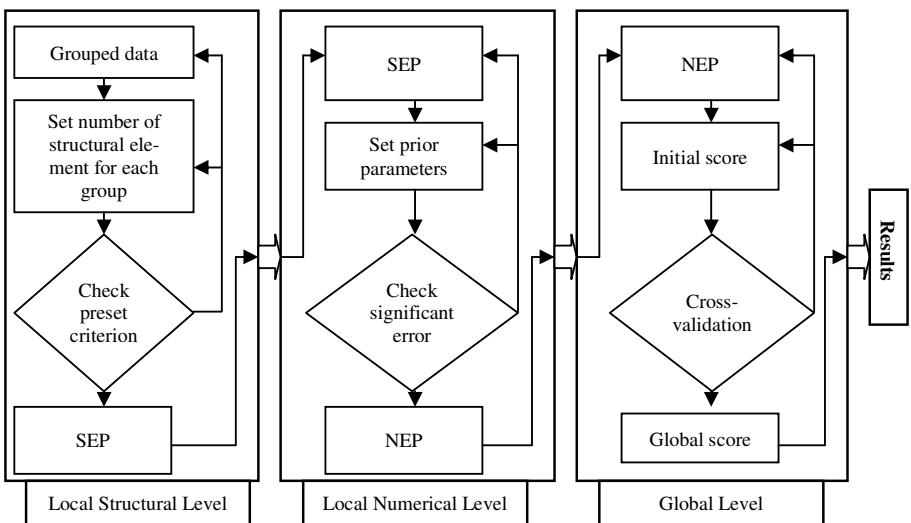


Fig. 1. The Framework of Element Oriented Analysis

In order to implement our EOA and demonstrate its effectiveness in this paper, we adopt two algorithms, Principle Component Analysis (PCA) and Multiple Discriminant Analysis (MDA) into LSL and LNL respectively.

3.1 Finding SEP in LSL

According to Fig.1, the major purpose in LSL is to find a SEP $S = \{s_1, \dots, s_k\}$ where s_i ($1 \leq i \leq k$) is any of the structural elements. The successful SEP contains most of information from the original data. We apply Principal Component Analysis (PCA) to detect SEP. PCA is a statistical procedure to qualify the major dimensions by the proportion criterion, which leads to a structural ranking with a specific gravity of information containing. The structural elements situate at the front ranking should comprise the most inherent content of the original data. Since the process in this level is in terms of total variance, only small numbers of structural elements are preserved.

For the purpose to achieve precise results, we firstly rebuild normalized data $D = (d_1, \dots, d_i)$ into a matrix with Pearson's Correlation γ , from $\gamma_{11} = \gamma(d_1, d_1)$ to $\gamma_{k(k-1)} = \gamma(d_k, d_{k-1})$. Thus, Pearson's coefficient γ_{ij} can be generally worked out by the following formula (2)

$$\gamma_{ij} = \frac{\sum(d_i - \bar{d})(d_j - \bar{d})}{\sqrt{\sum(d_i - \bar{d})^2 \sum_{k=1}^n (d_k - \bar{d})^2}}, (i, j \in k) \quad (2)$$

According to the lemma of linear invariability $\gamma(zd_i, zd_j) = \gamma(d_i, d_j)$, where z is any nonzero constant, all different possible Pearson's coefficients are arranged into the new matrix R

$$R = \begin{bmatrix} \gamma_{11} & \cdots & \gamma_{k1} \\ \vdots & \ddots & \vdots \\ \gamma_{1k} & \cdots & \gamma_{kk} \end{bmatrix}$$

And formula (3) is then created

$$C_R = \frac{1}{n-1} RR^T \quad (3)$$

where C_R comprises the symmetric matrix with N columns and N rows.

According to $|C_R V - \lambda V| = 0$, the eigenvalue λ is expressed as $\lambda = V^T C_R V$, where $v_n^T v_m = 0$, it is also regarded as one of potential structural elements. Then we rank the valuable structural elements through the λ ordering, $\lambda_1 \geq \dots \geq \lambda_p \geq 0$. At the same time, the cumulative proportion σ regarded as the threshold criterion is worked out by the following function (4).

$$\sigma = \frac{\sum_{p=1}^i \lambda_p}{\sum_{p=1}^k \lambda_p}, (i = 1, 2, \dots, k) \quad (4)$$

After enough λ s are collected, we convert them to the structural elements s_i ($1 \leq i \leq k$) and define the SEP $S = \{s_1, \dots, s_k\}$.

3.2 Finding NEP in LNL

Given the SEP $S = \{s_1, s_2, \dots, s_k\}$, the task in LNL is to define a NEP $N = \{n_1, \dots, n_p\}$, where each n_i ($1 \leq i \leq p$) can be associated with a specific polynomial function of SEP. Hence, the first step of finding NEP is to estimate all relationships between the numerical elements and SEP, which can be obtained by applying the multiple discriminant functions (5).

$$n_i = \alpha_{i1}s_1 + \alpha_{i2}s_2 + \dots + \alpha_{ik}s_k + \beta_i, (1 \leq i \leq p) \quad (5)$$

where n_i is any numerical element of the NEP. The second step is to select the most accurate discriminant functions and corresponding numerical elements into the NEP.

A successful NEP quantitatively estimates the relationship between SEP and the numerical elements on the one hand. On the other hand, this NEP can be used for reference to define discriminant or predictive score in the subsequent Global Level (GL).

3.3 Mining in Global Level (GL)

The results from LNL are delivered to the last level of EOA, the Global Level (GL). The distinct discriminant scores are identified in this level so that the original data set can be uncovered accurately.

The observational data is segmented into several groups in GL and each single group is reserved as the validation data to test the accuracy of discriminant scores. This validation process is repeated until all groups have been considered. An optimal set of discriminant scores is produced based on iterative comparison and set as predictive benchmark in the further analysis. (Please note that the optimal discriminant scores might be variant due to the different observations).

4 Experiments

This section is dedicated to the empirical evaluation of our EOA that is described above. In the following, we discuss the real datasets in our experiment. We then address the experimental setting and evaluation. We also report on the comparative results between EOA and K-Means Clustering.

4.1 Experimental Data

This section presents an experimental study on real-world datasets to test EOA in multidimensional data mining. In our experiments, the chosen dataset is the data from privately owned Turkish commercial banks during the period of 1997 to 2003, which is available at www.tbb.org.tr/english/default.htm. The selected data from privately owned Turkish commercial banks is based on the financial ratios. Those ratios are grouped into seven general financial fields, which are Capital Ratios, Assets Quality, Liquidity, Income-Expenditure Structure, and Share in Sector, Branch Ratios, and Activity Ratios.

4.2 Experimental Setting

In contrast to many other techniques that observe the original value of records, our approach needs data which is reduced into a limited and unique range because the perspective of our approach is to gain useful analysis through all variables together. The raw Turkish bank data is comprised by financial ratios and is cataloged to seven variant fields. Each field has a different ratio range. To better implement our experiment, we firstly apply 0-1 normalization measure to transfer the original data value into the range of zero to one.

In this experiment, we apply Principal Component Analysis (PCA) to detect SEP and we predetermine the cumulative proportion $\sigma=80\%$ to filter the effective structure elements from each sub-catalog. The selected structure elements are joined into a vector matrix of SEP and renamed to $S = [s_1, s_2 \dots, s_k]^T$ where $s_k = \lambda_p$.

We utilize Multiple Discriminant Analysis (MDA) in LNL to obtain NEP. Two Multiple Discriminant Functions are therefore expected because the experimental data consists of two groups, bankrupt banks and non-bankrupt banks. Furthermore, we set prior probabilities 50% of non-bankruptcy versus 50% of bankruptcy and significant level 0.05.

In GL, we need to determine the discriminant scores to predict Turkish bank data into bankrupt or non-bankrupt. The decision to classify a bank as bankrupt or non-bankrupt is made by comparing its score to the discriminant score.

4.3 Experimental Results

In LSL, an eigenvalue greater than a certain level indicates a principal component that accounts for more of the variance than the original data. This can be confirmed by our statistical summary in Table 1 below and visual output of analysis between the eigenvalues and principal components that are shown in Fig.2 below.

Table 1. The Result from LSL

Catalogs	Structural Elements	Number of Structural Elements	Reductive Percentage of Features	Cumulative Percentage of Variance
1	s1 s2 s3	3	50%	94.02%
2	s4 s5 s6	3	25%	92.24%
3	s7 s8	2	33.30%	88.92%
4	s9 s10 s11 s12	4	76.47%	82.64%
5	s13 s14	2	66.67%	96.69%
6	s15 s16 s17	3	57.14%	88.72%
7	s18 s19 s20	3	50%	93.89%
Mean		2.857143	51.23%	91.02%
Std		0.638877	0.1657	0.0432

According to Table 1, 2.857 structural elements are selected to replace the original data on average. It assists to reduce observational size by 51.23%. However, it still contains 91.02% of the variance of the original data on average. A SEP is built by 20 structural elements, s_1 to s_{20} .

The seven plots combined in Fig. 2 show the change tendency between the quantity of eigenvalues and increasing numbers of principal components in all seven catalogs. The intersection between the reference line at a given point and the decline level of eigenvalues indicates the optimum numbers of principal components to be extracted. In this experiment, the principal components over reference line are selected as structural elements.

A regression function is generated to mathematically describe the SEP which is shown in formula (6) below

$$\begin{aligned}
 SEP = & 1.1768s_1 - 0.0635s_2 + 5.9788s_3 + 7.4251s_4 + 8.5415s_5 + 6.1552s_6 \\
 & + 11.6425s_7 + 8.1323s_8 + 2.7031s_9 - 0.8992s_{10} + 2.4225s_{11} \\
 & - 2.0981s_{12} + 0.1624s_{13} - 0.0182s_{14} + 1.0232s_{15} + 0.5715s_{16} \\
 & + 1.5985s_{17} - 0.6453s_{18} - 1.5788s_{19} - 2.0647s_{20}
 \end{aligned} \tag{6}$$

According to statistical testing which is shown in Table 2, this SEP has highly significant statistical value and is suitable to utilize in LNL.

Table 2. The Multivariate Statistic Testing

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.22975257	6.03	20	36	<0.0001
Pillai's Trace	0.77024743	6.03	20	36	<0.0001
Hotelling-Lawley Trace	3.35250847	6.03	20	36	<0.0001
Roy's Greatest Root	3.35250847	6.03	20	36	<0.0001

However, we are still concerned with the disordered plot of correlation among structural elements s1 to s20. We cannot get any useful information from Fig. 3 to uncover bankrupt or non-bankrupt bank grouping.

Remark 1. The intersection between “blue” decline tendency line and “red” reference line indicates the optimal numbers of principal components that need to be selected).

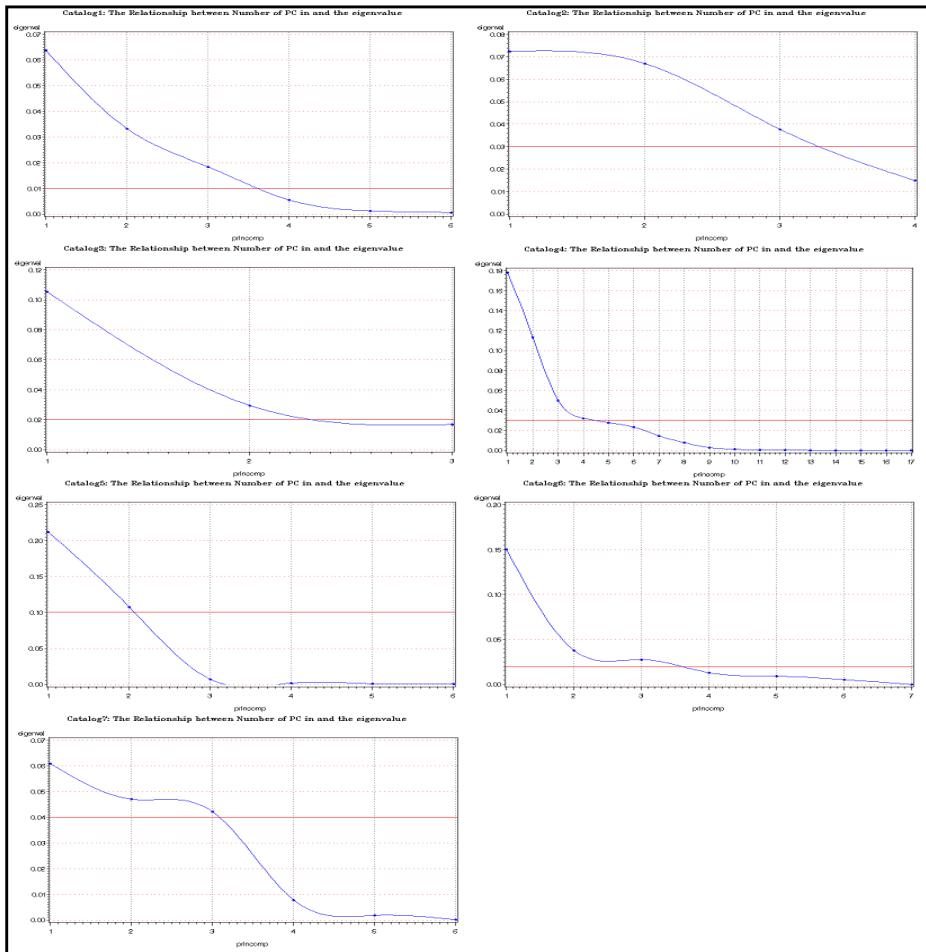


Fig. 2. The Relationship Between Number of Principal Components and Decline Line of Eigenvalues in Seven Catalogs



Fig. 3. The Correlationship in Structural Element s1 to s20

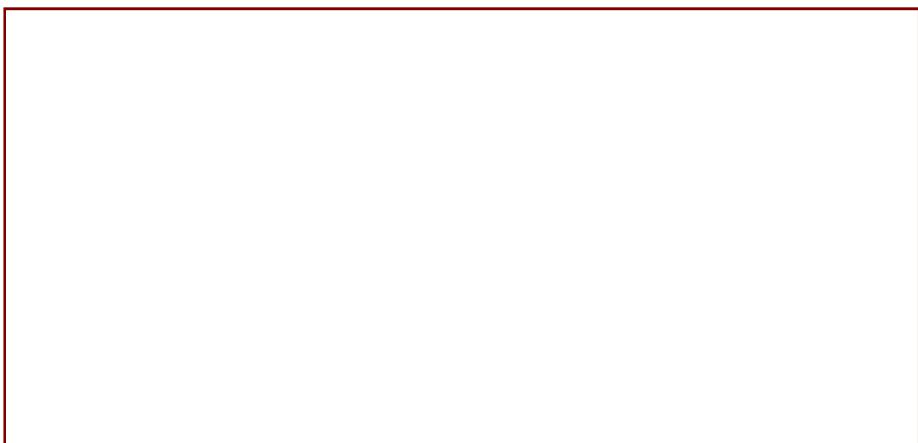


Fig. 4. Bankrupt and Non-Bankrupt Data are Grouped by Numerical Element N0 and N1

Then, in order to express the quantitative correlation between NEP and the structural elements, LNL produces two numerical elements, n0 by formula (7)

$$\begin{aligned}
 n0 = & -1.6366 + 1.6166s_1 - 0.0872s_2 + 8.2129s_3 + 10.1998s_4 + 11.7333s_5 \\
 & + 8.4553s_6 + 15.9931s_7 + 11.1712s_8 + 3.7132s_9 - 1.2353s_{10} \\
 & + 3.3278s_{11} - 2.8821s_{12} + 0.2231s_{13} - 0.0249s_{14} + 1.4055s_{15} \\
 & + 0.7850s_{16} + 2.1959s_{17} - 0.8864s_{18} - 2.1688s_{19} - 2.8363s_{20}
 \end{aligned} \quad (7)$$

and n1 by formula (8)

$$\begin{aligned}
n1 = & -3.4659 - 2.7714s_1 + 0.1495s_2 - 14.0794s_3 - 17.4854s_4 - 20.1143s_5 \\
& - 14.49479s_6 - 27.4168s_7 - 19.1507s_8 - 6.3655s_9 + 2.1176s_{10} \\
& - 5.7048s_{11} + 4.9408s_{12} - 0.3824s_{13} + 0.0428s_{14} - 2.4094s_{15} \\
& - 1.3458s_{16} - 3.7644s_{17} + 1.5196s_{18} + 3.7180s_{19} + 4.8622s_{20}
\end{aligned} \quad (8)$$

The results of discriminant by n0 and n1 are clearly shown in Fig. 4. The label of the blue circle in Fig. 4 addresses the bankrupt group. And the label of the red triangle represents the group of non-bankrupt bank.

According to Table 3, we can obtain that the accuracy rates of bankruptcy prediction are 95.24%, 95.24%, 95.24%, 100% and 100%. They uncover that EOA has extraordinarily strong capability in bankruptcy prediction. The precision and stability of EOA also demonstrate its feasibility in multidimensional data mining.

Table 4 portrays the comparative results of predictive performance between EOA and K-Means Clustering. In general, our EOA presents distinct advantage in signaling bankruptcy over K-Means Clustering. These advantages are addressed by the following observations.

Table 3. The Accuracy of EOA in Signaling Bankruptcy of Turkish Bank Data

Period	Actual Bank Status	Discriminant Group Membership	
		Bankrupt	Non-Bankrupt
Year 1	Bankrupt	95.24%	4.76%
	Non-Bankrupt	0%	100%
Year 2	Bankrupt	95.24%	4.76%
	Non-Bankrupt	5.56%	94.34
Year 3	Bankrupt	95.24%	4.76%
	Non-Bankrupt	5.56%	94.34
Year 4	Bankrupt	100%	0%
	Non-Bankrupt	16.67%	83.33%
Year 5	Bankrupt	100%	0%
	Non-Bankrupt	25%	75%

Table 4. The Comparison of Overall Performance between EOA and K-Means

Period	Model Accuracy		Model Precision		Type I Error Rate		Model Specificity	
	EOA	K-Means	EOA	K-Means	EOA	K-Means	EOA	K-Means
Year 1	98.25%	70.18%	97.30%	91.30%	4.76%	9.52%	97.30%	90.48%
Year 2	94.74%	43.86%	97.14%	100%	4.76%	0%	97.14%	100%
Year 3	94.74%	49.12%	97.14%	81.82%	4.76%	9.52%	97.14%	90.48%
Year 4	89.74%	45.61%	100%	77.78%	0%	9.52%	100%	90.48%
Year 5	84.21%	59.65%	100%	63.83%	0%	80.95%	100%	19.05%
Average	92.34%	53.70%	98.32%	82.95%	2.86%	21.90%	98.32%	88.10%

- The Accuracy Rate of EOA which is used to evaluate model predictive capability appears strongly accurate. On average, the result from EOA

stands at a very significant level, 92.34%. It is also 38.64% higher than the accuracy rate from the K-Means Clustering approach.

- Model Precision of EOA that measures the quality of non-bankruptcy prediction of the model shows 98.32% on average. In addition, the variation of Model Precision during five years is quite slight. However, the result from K-Means Clustering indicates 82.95% on non-bankrupt forecast on average. In particular, its results on Year 4 (77.78%) and Year 5 (63.83%) present obviously imprecision compared to the results of EOA from the same period.
- Type I Error Rate of EOA is 2.86% on average. In particular, it is 0% error rate in Year 4 and Year 5. In other words, we only have a tiny chance to make a mistake when we detect a “Bankrupt Bank” by EOA. Hence, the corresponding Model Specificity that is used to assess the correctness of a bankrupt bank likewise shows 98.32% on average. At the same time, K-Means Clustering produces 21.90% error rate on average. We especially note that the occurrence of error rate in Year 5 is 80.95%.

Therefore, all the assessments of EOA and the comparative results with K-Means Clustering show that EOA is a successful approach in multidimensional data mining.

5 Conclusion and Future Work

In this paper, we have provided EOA as a novel approach to perform multidimensional data mining. In the experiment, this approach is employed to detect bankruptcy status of Turkish Bank data over five years. The high accuracy of results shows that EOA is an effective and efficient approach.

EOA achieves multidimensional data mining through three levels. Firstly, it searches valuable structural elements and targets them as qualitative observations. Secondly, EOA provides quantitative analysis on these qualitative observations and generates numerical elements. Lastly, the results comprised of numerical elements which address precise correlation of previous structural elements are delivered to predictive validation.

Our future work will include more in improving structural and numerical patterns. For example, we will attempt to use some decision tree algorithms such as C5.0 in SEP finding and apply AdaBoost to NEP, etc. In addition, we are planning to create dynamic structural patterns to replace current static patterns. Dynamic structural patterns will be easily utilized in the detection of time-changing data and incomplete data. In the future, our EOA will be applied to stock data, medical data and taxation data.

Acknowledgements

This work has been supported under Australian Research Council’s Linkage Projects Funding Scheme (project number LP0561985).

References

1. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning* 36(1/2), 105–139 (1999)
2. Breiman, L.: Bagging Predictors. *Machine Learning* 24(2), 124–140 (1996)
3. Huang, T., Kecman, V., Kopriva, I.: *Kernel Based Algorithms for Mining Huge Data Sets*. Springer, Heidelberg (2006)
4. Hossari, G.: A Dynamic Ratio-Based Model for Signalling Corporate Collapse. *The Journal of Applied Management Accounting Research* 4(1), 11–32 (2006)
5. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Nédellec, C., Rouveiro, C. (eds.) ECML 1998. LNCS, vol. 1398. Springer, Heidelberg (1998)
6. Kim, Y., Street, W., Menczer, F.: Feature selection in unsupervised learning via evolutionary search. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 365–369. ACM Press, Boston (2000)
7. Kim, Y., Street, W., Menczer, F., Russell, G.: Feature Selection in Data Mining. In: *Data Mining: Opportunities and Challenges*, pp. 80–105. Idea Group Publishing (2003)
8. Lin, W., Orgun, M., Williams, G.: Temporal Data Mining Using Hidden Markov-Local Polynomial Models. In: Cheung, D., Williams, G.J., Li, Q. (eds.) PAKDD 2001. LNCS (LNAI), vol. 2035, pp. 324–335. Springer, Heidelberg (2001)
9. Liu, H., Yu, L.: Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Transactions on Knowledge and Data Engine* 17(3) (March 2005)
10. Magidson, J.: The CHAID approach to segmentation modeling. In: *Handbook of Marketing Research* (1993)
11. Quinlan, J.R.: *C 4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo (1993)
12. Wagsta, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained K-means Clustering with Background Knowledge. In: *The Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pp. 577–584 (2001)
13. Wang, W., Yang, J.: Mining High-Dimensional Data. In: *The Data Mining and Knowledge Discovery Handbook*, pp. 793–799. Springer, Heidelberg (2005)
14. Zhang, Y., Orgun, M., Lin, W.: Unsupervised Learning Aided by Clustering and Local-Global Hierarchical Analysis in Knowledge Exploration. *Journal of Digital Information Management* 5(4), 237–246 (2007)

Evolutionary Feature Selections for Face Detection System

Zalhan Mohd Zin¹, Marzuki Khalid², and Rubiyah Yusof²

¹ Section of Industrial Automation – UniKL-Malaysia France Institute (UniKL-MFI)

² Center for Artificial Intelligence and Robotics (CAIRO) – Universiti Teknologi Malaysia (UTM)

zalhan@mfi.unikl.edu.my, marzuki@utmkl.utm.my

Abstract. Various face detection techniques has been proposed over the past decade. Generally, a large number of features are required to be selected for training purposes of face detection system. Often some of these features are irrelevant and does not contribute directly to the face detection algorithm. This creates unnecessary computation and usage of large memory space. In this paper we propose to enlarge the features search space by enriching it with more types of features. With an additional seven new feature types, we show how Genetic Algorithm (GA) can be used, within the Adaboost framework, to find sets of features which can provide better classifiers with a shorter training time. The technique is referred as GABOost for our face detection system. The GA carries out an evolutionary search over possible features search space which results in a higher number of feature types and sets selected in lesser time. Experiments on a set of images from BioID database proved that by using GA to search on large number of feature types and sets, GABOost is able to obtain cascade of boosted classifiers for a face detection system that can give higher detection rates, lower false positive rates and less training.

Keywords: Genetic Algorithm, cascade of classifiers, Adaboost, rectangle features.

1 Introduction

To detect human faces in images in real-time is a real challenging problem. Viola and Jones [1] were the first who developed a real-time frontal face detector by introducing boosted cascade of simple features that achieves comparable detection and false positive rates to actual state-of-the-art systems [4][5][6][7]. Many researchers have proposed to enhance the idea of boosting simple weak classifiers. Li et al. [8] describe a variant of Adaboost called Floatboost for learning better classifiers. Lienhart et al. [2] showed that by extending the basic feature types and sets, detectors with lower error rates are produced. However, extension of feature types automatically leads to much higher number of feature sets and expand the search space thus increases the training times. Recapitulating the research that is based on the publication of Viola and Jones [1] we can see that there are mainly two problems to deal with: (1) Extending the feature sets and being able to search over these very large sets in reasonable time. (2)

The best feature solution sets are not known in advance. To overcome these problems, we use GA in combination with Adaboost to search over a large number of possible features. Our goal is to find better cascade of classifiers by using a very large feature sets in less time, and that achieves comparable or even better classification results compared to the cascade of classifiers that are trained exhaustively over a small feature sets. The use of Evolutionary Algorithms in the field of image processing, especially automatic learning of features for object detection has received growing interest. Treptow and Zell [3] showed an Evolutionary Algorithm can be used within Adaboost framework in single stage classifiers to find features which provide better classifiers for object detections such as faces and balls. In 2006, J. S. Jang and J. H. Kim [9] introduced the employment of Evolutionary Pruning in cascaded structure of classifiers which has a purpose to reduce the number of weak classifiers found before by Adaboost for each stage of cascade training. However, this approach was aimed to focus more on increasing the performance of face detection speed by reducing the number of weak classifiers in already built cascade while ignoring the cascade training time. Our approach is focused on reducing the computational training time to build 15 stages cascade of boosted classifiers while having similar or better cascade performance. In order to achieve that goal, we enrich the possible feature solutions by adding seven new types of features. These additional features will increase the size of the search space thus the cascade of classifiers training time. GA is then implemented into Adaboost framework to select good features sets which represent sets of strong classifiers for each stage of cascade training. The cascade built consists of the combination of eight basic features and seven new feature types.

The paper is organized as follows: In the next Section, Adaboost learning procedure is introduced. Section 3 describes cascade of boosted classifiers. In Section 4, application of GA into Adaboost framework to build a cascade of classifiers using large number of feature types is introduced. This includes the characteristics of seven new feature types. Section 5 shows the result of performance of the cascade trained using GA compare with cascade trained exhaustively in terms of hit rates, missed rates, false positive rate and training time. The experiments were done by using open source software, Intel OpenCV, as done in [2]. Section 6 summarizes and concludes our work and points out perspectives for further future research.

2 Adaboost Learning of Object Detectors

Viola and Jones [1] developed a reliable method to detect objects such as faces in images in real-time. An object that has to be detected is described by a combination of a set of simple Haar-wavelet like features shown in Fig. 1.

The sums of pixels in the white boxes are subtracted from the sum of pixels in the black areas. The advantage of using these simple features is that they can be calculated very quickly by using “integral image”. An integral image $I\!I$ over an image I is defined as follows:

$$I\!I(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \quad (1)$$

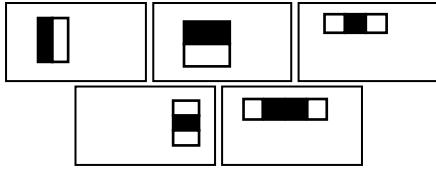


Fig. 1. Five different basic types of rectangle features within their sub window of 24x24 pixels. These five types of features are the initial features used to train cascade of classifiers exhaustively.

In [1] also, it is shown that every rectangular sum within an image can be computed with the use of an integral image by four array references. A classifier has to be trained from a number of available discriminating features within a specific sub window in order to detect an object. The possible positions and scales of the five different feature types as shown in Fig. 1 produce about 90,000 possible alternative features within a sub window size of 24x24 pixels. This number exceeds largely the number of pixels itself. Therefore, a small set of features which best describe the object to be detected, has to be selected. Adaboost [10] is a technique that initially selects good classification functions, such that a final “strong classifier” will be formed, which is in fact, a linear combination of all weak classifiers. In the general context of learning features, each weak classifier $h_j(x)$ consists of one single feature f_j :

$$h_j(x) = \begin{cases} 1 & : p_j f_j(x) < p_j \vartheta_j \\ 0 & : \text{otherwise} \end{cases} \quad (2)$$

where ϑ_j is a threshold and p_j a parity to indicate the direction of the inequality. The description of Adaboost algorithm to select a predefined number of good features given a training set of positive and negative example images is shown in Fig. 2.

The Adaboost algorithm iterates over a number of Trounds. In each iteration, the space of all possible features is searched exhaustively to train weak classifiers that consist of one single feature. During this training, the threshold ϑ_j must be determined for the feature value to discriminate between positive and negative examples. Therefore, for each possible feature and given training set, the weak learner determines two optimal values (thresholds), such that no training sample is misclassified. For each weak classifier $h_j(x)$ the error value ε_j will be calculated using misclassification rate of all positive and negative training images. It gives each feature $h_j(x)$ trained with its respective error value ε_j which is between 0 and 1. The best feature $h_i(x)$ found with the lowest error rate ε_i will be selected as the weak classifier for this $1/T$ iteration. After the best weak classifier is selected, all training examples concerned are re-weighted and normalized to concentrate in the next round particularly on those examples that were not correctly classified. At the end, the resulting strong classifier is a weighted linear combination of all Tweak classifiers.

- 1) Input: Training examples (x_i, y_i) , $i = 1..N$ with positive ($y_i = 1$) and negative ($y_i = 0$) examples.
 - 2) Initialization: weights $\omega_{l,i} = \frac{1}{2m}, \frac{1}{2l}$ with m negative and l positive examples
 - 3) For $t=1, \dots, T$:
 - a) Normalize all weights
 - b) For each feature j train classifier h_j with error

$$\varepsilon_j = \sum_i |h_j(x_i) - y_i|$$
 - c) Choose h_t with lowest error ε_t
 - d) Update weights: $\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-\varepsilon_t}$ where $e_i = 0$ if x_i is correctly classified and $e_i = 1$ otherwise and $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$
 - 4) Final strong classifier: $h(x) = \begin{cases} 1 : \sum_{t=1}^T \alpha_t h_t(x) \geq 0.5 \sum_{t=1}^T \alpha_t & \text{with} \\ 0 : \text{otherwise} & \end{cases}$
- $$\alpha_t = \log\left(\frac{1}{\beta_t}\right)$$

Fig. 2. Adaboost learning algorithm as used in [1]. This algorithm is used to select sets of weak classifiers to form strong classifiers from all possible features types. The search of good feature h_t was done exhaustively as stated in step 3b above.

3 Cascade of Boosted Classifiers

This section describes an algorithm for constructing a cascade of classifiers which drastically reduces the computational time. The main idea is to build a set of cascade boosted classifiers which are smaller, but more efficient, that will reject most of the negative sub-windows while detecting almost all positive instances. Input for the cascade is the collection of all sub-windows also called scanning windows. They are first passed through the first stage in which all sub-windows will be classified as faces or non faces. The negative results will be discarded while the remaining positive sub-windows will trigger the evaluation of the next stage classifier. The sub-windows that reach and pass the last layer are classified as faces (See Fig. 3). Every layer actually consists of only a small number of features. In the early stages, with only small number of selected features it is possible to determine the existence of a non-face. On the other hand, determining the presence of a face usually needs more features. The trained cascade boosted classifiers usually have an increasing number of features in each layer until its last layer and therefore became increasingly more complex.

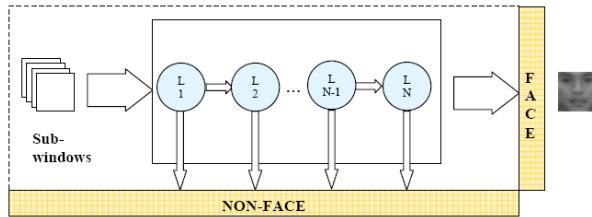


Fig. 3. Cascade from simple to complex classifiers with N layers

During the training of the cascade of boosted classifiers, the number of features per layer or stage was driven through a “trial and error” process. In this process, the number of features was increased until a significant reduction in the false positive rate could be achieved. In our case, each stage was trained to eliminate 50% of the non-face patterns while falsely eliminating only 0.005% of the frontal face patterns; 15 stages were trained and a false alarm rate about $0.5^{15} \approx 3 \times 10^{-5}$ and a hit rate about $0.995^{15} \approx 0.93$ were expected. More features were added until the false positive rate on the each stage achieved the desired rate while still maintaining a high detection rate. At each training stage, the false positive images from previous stages are added to the sets of negative or non-faces images and this set of images is used as negative images in the next stages training [9].

4 Genetic Algorithm for Feature Selections

Genetic Algorithms (GAs) [11] are a family of computational models inspired by natural evolution. GAs comprise a subset of evolution-based optimization techniques focusing on the application of selection, mutation, and recombination or crossover to a population of competing problem solutions. In our paper, the chromosome of GA represents the specific type and location of one single feature in the sub-window of 24x24 pixels. Each chromosome has six elements. The first five elements are integer types which consist of:

- *type* : type of feature
- *x* : coordinate x in sub-window
- *y* : coordinate y in sub-window
- *dx* : width of feature
- *dy* : height of feature

The sixth element is a decimal number type between 0 and 1 which stores the fitness value of the respective feature. As described previously in Adaboost, every trained feature or weak classifier produce an error value ε_j and the feature with the lowest ε_j

will be selected. Therefore, fitness function chosen here is $1 - \varepsilon_i$ where *i* is the number between 1 and population size N. By using this fitness function, the higher fitness value indicates the lower error value ε_j . To increase the possibility of selecting good features, we propose to enlarge the number of feature types. The existing three feature

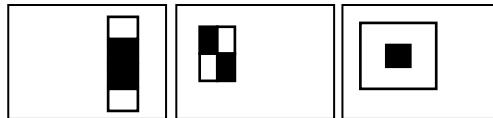


Fig. 4. The existing three types of features within their sub window of 24x24 pixels. These feature sets are added in training of cascade of classifiers with GA search.

types in Fig. 4 are added into the total feature sets. These three features are part of right rectangle features which already exist in Intel OpenCV along with the 5 basic feature types that had been shown in Fig. 1. In addition to that, we add our newly proposed seven types of features that increase the search space and possibilities of getting better cascade of classifiers with higher performance.

In Fig. 5, features *a*, *b*, *c* and *d* with the style of an L-shape and an inverse L-shape should have the ability to distinguish the image of face and non-face based on the pattern of the side of the face specifically on the left and right foreheads, temples and jaw lines. Feature *e* is the inverse of the middle feature shown in Fig. 4. And lastly feature *f* should distinguish the pattern of both eyes region with forehead and feature *g* should make the difference between face images and non-face images based on the pattern of eyes location. With all these, the total feature types equaled to 15. As a result, while more valuable and better types of features might be created or existed as the potential sets of good feature solutions, the search space of all of these feature types increased dramatically. Therefore GA is used to select the features and to avoid the exhaustive search and high computational time.

Initially, in the first generation of GA, all chromosomes are randomly generated and evaluated to determine their fitness value. Ranking Scheme selection is used

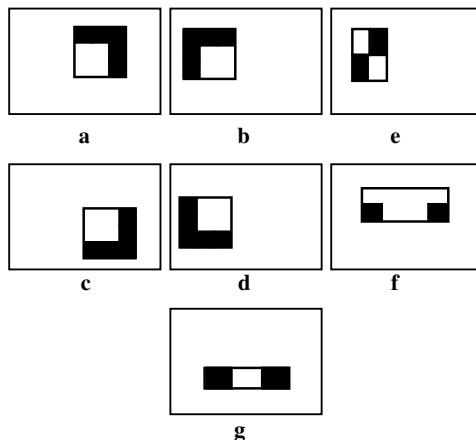


Fig. 5. The newly proposed seven types of features within their sub window of 24x24 pixels. These feature sets are proposed and added in training of cascade of classifiers with GA search.

when all chromosomes are ranked based on their fitness values. The first chromosome will have the highest fitness value while the last one will have the lowest one. In each generation, the chromosomes from half of the total populations will be selected and put into a mating pool and will become parent chromosomes. The two-point standard crossover is applied here. Based on the crossover probability rate, p_{co} , where $p_{co} \in \{0...1\}$, two different chromosomes are chosen from the mating pool. At the same time, two random positions of chromosome's elements are chosen which are m and n such that $m, n \in \{1..5\}$ with $m \neq n$. Then, the elements of m and n are crossover within these two parents to produce new children. In mutation process, mutation probability rate p_{mt} where $p_{mt} \in \{0...1\}$ is used. Based on this rate, mutation process will take place on the chromosome chosen from the mating pool. Once mutation occurs, another random number between 1 and 5 is selected. This number represents the location of the gene to be mutated in this chromosome. If the first gene *type* is selected, the new and different type of feature will be selected randomly between 1 and 15. While in the other cases such as genes x , y , dx or dy , its value will be added with an integer value randomly chosen between -2 and 2.

All parent chromosomes that have undergone crossover and mutation process will produce new children chromosomes and they are re-evaluated to determine their fitness values, $1 - \epsilon_i$. The new chromosomes with high fitness values will have a better chance to be selected and inserted into its appropriate rank in main GA population. For the new chromosomes that become invalid or no longer feasible, for example the sum of their x-coordinate and their width exceed the allowed width of the sub-window, here is 24x24, then their fitness value will be assigned with zero, in other word, these features are not the good features and they shall be discarded. While the chromosomes with modest or average fitness values will not have an opportunity to be selected and inserted neither in the main population nor in the next mating pool for the next generation. The 200-GA population size chosen is the result of a trial and error process in which a trade-off is made between speed and error rate. The size of a population should be sufficiently large to create sufficient diversity covering the possible solution space. Other parameters of GA are the probabilities for crossover and mutation operators. In their paper, Treptow and Zell [3] had preferred 20% of crossover rate and 80% of mutation rate. Since they used EA to select about 200 features in only a single stage of classifiers, the ratio used seems to be reasonable. In our case, we have proposed to use 90% as crossover rate and 10% as mutation rate. This is due to the fact that the training of 15 stages cascade of classifiers is slightly different from the training of a single stage classifiers as described in Section 3 even though both use Adaboost algorithm. At each different stage in the cascade training, GABOost will select features from slightly different search spaces. This is due to the new generated images samples incorrectly classified (false positive) in the previous stages that are added into the training sets. In order to avoid early local optima or to loose good features solutions, the exploration of feature search space should have higher priority. So, to train cascade of classifiers which involved 15 stages, as in our case, the ratio 9:1 crossover-mutation rate seems to be logical and reasonable. The parameters of Genetic Algorithm are shown in Table 1.

Table 1. Parameters used in Genetic Algorithm

GA parameters	
Population Size	200
Crossover Type	Two-points Std
Crossover Rate	0.9
Mutation Type	Single Gene
Mutation Rate	0.1

5 Experiments of Evolutionary Search Using Genetic Algorithm

In this Section, we compare the performance of the 15 stages cascade of classifiers built by using Adaboost with exhaustive search which we refer to as ExBoost and Adaboost with GA to select features, which we refer to as GABOost. ExBoost search over only five basic feature types while GABOost search over a larger set of 15 feature types. The training sets used consist of 7,000 positive images and 3,000 negative images that we gather from various sources (See Fig. 6). The dimensions of these gray value images are 24x24 pixels and are different one from another.



Fig. 6. Examples of faces and non-faces images in the training set

The test set consists of face dataset images from BioID [13]. This dataset contains 1,200 face images with different people, gender, face expression, size and location, level of illumination and also appearance of non-face objects. The examples are shown in Fig. 7. All images in the training and testing datasets are different from one to another.



Fig. 7. Examples of images containing faces with various conditions used in the BioID test set [13]

The total trainings were stopped after a set of 15 cascaded stages has been built. The population size is 200, all chromosomes are initialized randomly, crossover rate is 0.9 and mutation rate is set to 0.1. GA is converged and will stop if no better single feature found within the next 50 consecutive generations. In case of no convergence, GA will stop as well if it reaches the maximum number of generations which is set to 200. Experiments were carried out on an Intel Pentium IV 3.0 GHz processor. GABOost was run 10 times and the average results are taken. In Table 2, we can see that GABOost performs 3.7 times faster than ExBoost. The comparison of computational training time between ExBoost and GABOost in the ten experiments is shown in Fig. 8.

Table 2. Training time taken to build 15 stages cascade of classifiers and their number of features selected

Algorithm	Average training time (sec)	Average time to select single feature (sec)	Average number of features in cascade
ExBoost 5F	134225	336.4	400
GABOost 15F	35634	56.6	630

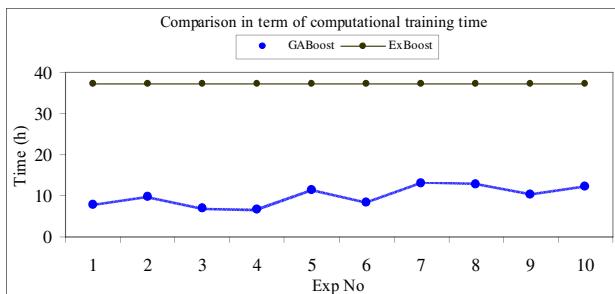


Fig. 8. Comparison of computational training time between ExBoost and GABOost in ten experiments

Table 2 also shows the total number of features found by both algorithms. We can see that GABOost selected more features compared to ExBoost. However, with less training time, selection of a single feature just require 56.6 seconds in GABOost while in ExBoost, 336.4 seconds need to find a single feature. This showed that selection of a single feature is 6 times faster in GABOost than in ExBoost.

Table 3 presents the average hit rate of GABOost (90.13%) is slightly superior to the hit rate of ExBoost (90.01%). From the experiments also, the best hit rate achieved by GABOost is 94.25% while the worst hit rate is 85.68%. GABOost also performs better in false positive rates. False positive rate is calculated based on the sum of all false positives rectangles detected divided by the total number of detection rectangles in the whole test sets. The examples are shown in Fig. 9.

Table 3. Comparison of hit rates and false positive rate performed by the cascades of classifiers built by ExBoost and GABoost

Algorithm	Average hit rate (%)	False positive detection/total number of detection (%)
ExBoost 5F	90.01	62.97
GABoost 15F(10 Exp)	90.13	61.56
GABoost 15F(best-hit)	94.25	62.32
GABoost 15F(worst-hit)	85.68	49.83

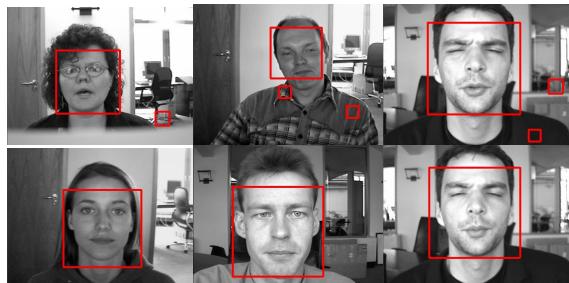


Fig. 9. The examples of the test images show faces are detected. The top three images however show false positive detections in single image while in the bottom three images, detection of faces are done perfectly.

In our experiments, the average false positive rate achieved by GABoost is 61.56% that is lower than false positive rate of ExBoost (62.97%). The hit rates and false positive rates of the ten experiments of GABoost is shown in Fig. 10.

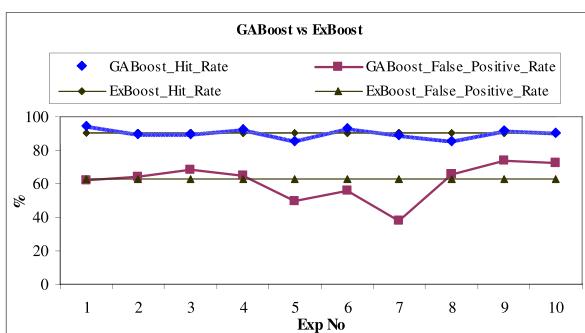


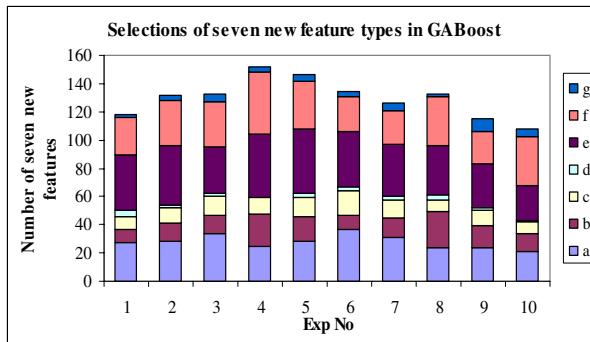
Fig. 10. Hit rates and False Positive rates in GABoost and ExBoost

Table 4. Details of the average number of seven new feature types selected by GABOost

Feature Type	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
Average number of feature selection	27.90	15.40	11.50	2.20	37.30	30.90	4.50
Total seven new features	129.70						
% from total features	20.60						

In Table 4, the results of the seven newly proposed feature types are examined. For each feature type *a*, *b*, *c*, *d*, *e*, *f* and *g* in Fig. 5, the average number of times that they are selected during cascade training using GABOost is shown.

These seven feature types contribute about 21% of the total features selected by GABOost in various stages. Among them, feature types *a*, *e* and *f* have important roles in the cascade training as they are selected many times. These three feature types contributes about 76.17% from the total seven new feature types proposed and about 15.67% from the total features selected in the cascades of boosted classifiers built by GABOost. Feature types *b* and *c* have little impact on the cascades as they were selected only 15.4 and 11.5 times. These numbers represent only 4.26% of the total features selected. Table 4 also shows that feature types *d* and *g* can not be considered as good feature types since they are rarely selected in the cascade training. Fig. 11 show the distributions of number of feature types selected among these seven new feature types.

**Fig. 11.** Selection of seven new feature types in GABOost

6 Conclusions and Future Works

In this paper, we have extended the work of Treptow and Zell [3] by implementing GA inside the AdaBoost framework to select features. Fifteen stages of cascaded classifiers are set up rather than building just a single stage classifier only. The technique of Viola and Jones [1] is also referred as we implemented the cascade training. Seven new feature types were proposed in order to increase the quality of feature solutions. As a consequence, the feature solutions sets become bigger and we have

shown how GA can be used to overcome the problem of feature selections in this huge search space. Training time was drastically reduced and the cascade of boosted classifiers trained using GABOost has achieved a better hit rate and lower false positive rate compared to the original exhaustive technique. This lower false positive rate might happen due to the higher number and more variety of features types selected by GABOost. We have shown also that the best cascade of boosted classifiers obtained by GABOost has outperformed the original cascade built exhaustively in computational training time, hit rate and false positive rate. On the other hand, the seven new feature types have shown different levels of importance in this training. Three of these new feature types have shown very significant roles in the training of cascade of boosted classifiers. However, we believe that the performance of these seven feature types could be different if another set of training samples is used. Nevertheless, they are many directions for further research in this field. Other search algorithms such as Memetic Algorithm [12] may be used to replace GA to select features in cascade training.

In GA itself, the dynamic rates of crossover and mutation could also be implemented and analyzed. This dynamic rate had been used as a tuning tool in one part of the research done by T. L. Seng, M. Khalid and R. Yusof [11]. On the other hand, since the speed of detection rate was not addressed in this paper and by looking at the number of features selected, it seems that the performance of the face detector will be slightly slower. This is due to a higher number of features selected by GABOost. Thus, the improvement of the cascaded classifiers built by GABOost with very large feature types could be done by using the Evolutionary Prunning, an approach introduced recently in 2006 by J. S. Jang and J. H. Kim [9]. Its purpose is to reduce the number of weak classifiers while maintaining the performance achieved by the cascade of classifiers. GABOost can then be further improved using this approach.

References

1. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR), Hawaii, USA, December 11-13 (2001)
2. Lienhart, R., Kuranov, A., Pisarevsky, V.: Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 297–304. Springer, Heidelberg (2003)
3. Treptow, A., Zell, A.: Combining Adaboost Learning and Evolutionary Search to select Features for Real-Time Object Detection. In: Proceedings Of the Congress on Evolutionary Computational CEC 2004, San Diego, USA, vol. 2, pp. 2107–2113 (2004)
4. Rowley, H., Baluja, S., Kanabe, T.: Neural Network-based Face Detector. IEEE Transc. on Pattern Analysis and Machine Intelligence 20(1), 23–28 (2000)
5. Sung, K., Poggio, T.: Example-based Learning For View-based Face Detection. IEEE Transaction on Pattern Analysis and Machine Intelligence 20, 39–51 (1998)
6. Schneiderman, H., Kanabe, T.: A Statistical method for object detection applied to faces and cars. In: International Conference on Computer Vision and Pattern Recognition, pp. 1746–1759 (2000)
7. Roth, D., Yang, M., Ahuja, N.: A Snowbased Face Detector. In: Advances in Neural Information Processing Systems 12 (NIPS 12), vol. 12 (2000)

8. Li, S.Z., Zhang, Z.Q., Shum, H., Zhan, H.J.: Floatboost Learning for Classification. In: 16th Annual Conference on Neural Information Processing Systems, NIPS (2002)
9. Jang, J.S., Kim, J.H.: Evolutionary Pruning for Fast and Robust Face Detection. In: IEEE Congress on Evolutionary Computation CEC 2006, Vancouver, Canada, pp. 1293–1299 (July 2006)
10. Freund, Y., Schapire, R.E.: A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence* 14(5), 771–780 (1999)
11. Seng, T.L., Khalid, M., Yusof, R.: Tuning of A Neuro-Fuzzy Controller by Genetic Algorithm With An Application to A Coupled-Tank Liquid-Level Control System. *International Journal of Engineering Applications on Artificial Intelligence* 11, 517–529 (1998)
12. Areibi, S., Moussa, M., Abdullah, H.: A Comparison of Genetic/Memetic Algorithms and Other Heuristic Search Techniques. In: International Conference on Artificial Intelligence, Las Vegas, Nevada, pp. 660–666 (2001)
13. BioID Face Database,
<http://www.bioid.com/downloads/facedb/index.php>

A Probabilistic Approach to the Interpretation of Spoken Utterances

Ingrid Zukerman, Enes Makalic, Michael Niemann, and Sarah George

Faculty of Information Technology, Monash University

Clayton, VICTORIA 3800, Australia

{ingrid, enes, niemann, sarahg}@csse.monash.edu.au

Abstract. In this paper we describe *Scusi?*, the speech interpretation component of a spoken dialogue module designed for an autonomous robotic agent. *Scusi?* postulates and maintains multiple interpretations of the spoken discourse, and employs a probabilistic formalism to assess and rank hypotheses regarding the meaning of spoken utterances. These constituents in combination enable *Scusi?* to cope gracefully with ambiguity and speech recognition errors. The results of our evaluation are encouraging, yielding good interpretation performance for utterances of different types and lengths.

1 Introduction

The *DORIS* project aims to develop a spoken dialogue module for an autonomous robotic agent, which supports the generation of responses that require physical as well as dialogue actions. In this paper, we describe *Scusi?*, *DORIS*'s language interpretation component, focusing on the techniques used to postulate and assess hypotheses regarding the meaning of a spoken utterance.

Minimally, a language interpretation component must be able to postulate promising interpretations, and decide whether there is a clear winner or several likely candidates to be passed to the dialogue system. These capabilities provide the basis for additional desiderata, viz recovering from erroneous interpretations, and adjusting interpretations dynamically as new information becomes available. The dialogue system in turn must determine an appropriate action. For example, consider the request “get me the blue mug”. If there is an aqua mug, an indigo mug and a light blue mug in view, the robot could do one of the following: (1) pick the ‘bluest’ mug among these candidates, (2) select one of these mugs at random, (3) ask a clarification question, or (4) look for a mug that better fits the request. The chosen action depends on the certainty associated with the options returned by the language interpretation module, and the decision procedures applied by the dialogue system.

In order to support the above capabilities, a discourse interpretation system should (1) maintain multiple interpretations, and (2) apply a ranking process to assess the relative merit of each interpretation. *Scusi?* does this, employing a probabilistic mechanism for the ranking component. Its interpretation process comprises three stages: speech recognition, parsing and semantic interpretation. Each stage produces multiple candidate options, which are ranked according to their probability of matching the speaker's

intention (Section 3). This probabilistic framework, together with the maintenance of multiple interpretations at each stage of the process, enable *Scusi?* to cope with ambiguity and speech recognition errors (Section 5). In addition, these constituents support the re-ranking of interpretations as new information becomes available, and hence the recovery from erroneous interpretations; and they enable *Scusi?* to abstract features of the interpretations which support the generation of appropriate dialogue or physical actions. Examples of these features are: number of highly ranked interpretations, the difference in their probability, and the similarity between them.

This paper is organized as follows. Section 2 outlines the interpretation process. The estimation of the probability of an interpretation appears in Section 3, and the semantic interpretation procedure in Section 4. Section 5 details our evaluation. Related research and concluding remarks are given in Sections 6 and 7 respectively.

2 Multi-stage Processing

Scusi? processes spoken input in three stages: speech recognition, parsing and semantic interpretation (Figure 1(a)). Our probabilistic approach resembles that of Miller *et al.* [1]. However, they considered textual input, and used semantic grammars tailored to a slot-filling application. In contrast, our grammars are syntactic, and we incorporate domain-related information only in the final stage of the interpretation process, which yields Conceptual Graphs [2] — a more general structure than frames.

In the first stage of our interpretation process, *Scusi?* runs Automatic Speech Recognition (ASR) software (Microsoft Speech SDK 5.1) to generate candidate texts from a speech signal. Each text is assigned a score that reflects the probability of the words given the speech wave. The second stage applies Charniak’s probabilistic parser (`ftp://ftp.cs.brown.edu/pub/nlparser/`) to generate parse trees from the texts. The parser generates up to N ($= 50$) parse trees for each text, associating each parse tree with a probability. During semantic interpretation, parse trees are successively mapped into two representations based on Conceptual Graphs: first *Uninstantiated Concept Graphs (UCGs)*, and then *Instantiated Concept Graphs (ICGs)*. UCGs are obtained from parse trees deterministically — one parse tree generates one UCG (but a UCG can have more than one parent parse tree).

A UCG represents syntactic information, where the concepts correspond to the words in the parent parse tree, and the relations between the concepts are directly derived from syntactic information in the parse tree and prepositions. Each UCG can generate many ICGs. This is done by nominating different instantiated concepts (relations) from *DORIS*’s knowledge base as potential realizations for each concept (relation) in a UCG (Section 4). Instantiated concepts are objects or actions in the domain, and instantiated relations are similar to semantic role labels [3]. Figure 1(b) illustrates the generation of one ICG for the request “leave the blue mug on the table”. The noun “mug” in the parse tree is mapped to the concept *mug* in the UCG, which in turn is mapped to the instantiated concept *mug03* in the ICG. The preposition ‘on’ in the parse tree is mapped to the relation *on* in the UCG, and then to the relation *Destination* in the ICG. Noun modifiers, such as colour and size, are treated as features to be matched to those of instantiated objects in the knowledge base. For instance, the colour *BLUE* is represented

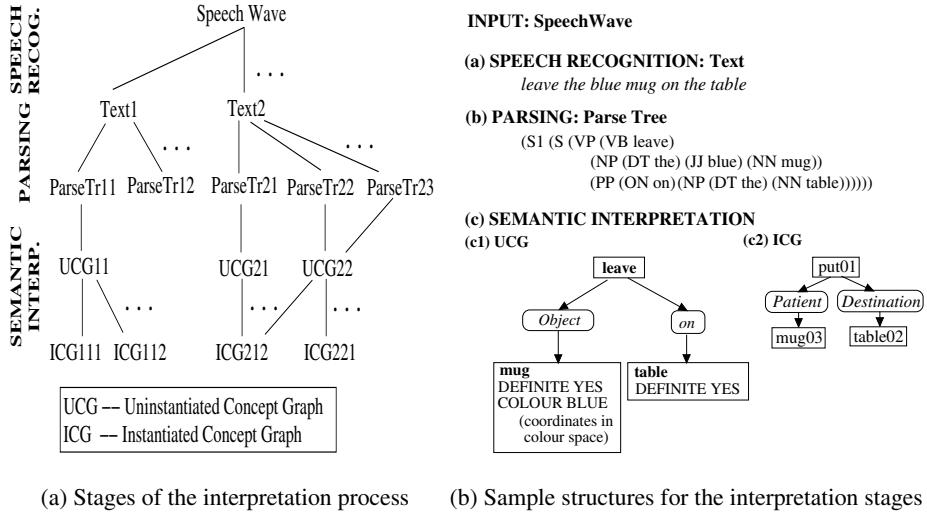


Fig. 1. *Scusi?*'s spoken language interpretation process

as a set of colour coordinates, which are then matched against the colour coordinates of stored objects [4].

The consideration of all possible options at each stage of the interpretation process is computationally intractable. *Scusi?* uses two computational devices to generate interpretations in real time: (1) an *anytime* algorithm [5], and (2) a processing threshold.

The **anytime algorithm** ensures that the system can return a list of ranked interpretations at any point after generating an interpretation component (text, parse tree, UCG or ICG). In each stage of the interpretation process, the algorithm applies a selection-expansion cycle to add an element to a search graph (Figure 1(a)) as follows. First, it selects an option for consideration (speech wave, textual ASR output, parse tree or UCG), and expands this option to the next level of interpretation. When an option is expanded, a single candidate is returned for this next level, but additional options reside in a buffer, which is created the first time the option is expanded. For example, when we expand a particular text, the parser returns the next most probable parse tree (but the first time this text is expanded, a buffer with at most N parse trees is created). Similarly, when we expand a UCG, the ICG-generation module returns the next most probable ICG, but the first time the UCG is expanded, a buffer of at most k_{max} ICGs is created (Section 4). Buffers are used, rather than piecemeal generation of alternatives, due to two reasons: (1) the ASR and parser return all the options at once; and (2) owing to the complex interactions between the components of ICGs, ICGs are not generated in descending order of probability (i.e., the best ICGs are often generated later on). By maintaining an ICG buffer for each UCG, higher-probability ICGs that are generated later can be slotted into the buffer (and considered by the selection-expansion process) in the order that reflects their probability. The selection-expansion process is repeated until one of the following happens: all options are fully expanded, a time limit is reached, or a specific number of iterations is performed. At any point after completing

an expansion, the anytime algorithm can return a list of ranked interpretations with their parent sub-interpretations (text, parse tree(s) and UCG(s)).

The **thresholding** approach is based on the observation that the probabilities of the texts returned by the ASR drop quite dramatically after the first few texts, as do the probabilities of the parse trees. We take advantage of this observation to prevent the consideration of unpromising alternatives as follows. When the probability of the next child of a parent node n drops below a threshold Thr relative to the probability of the most probable child of n , no additional children of n are considered. For example, for $Thr = 50\%$, if the probability of the next parse tree for text T_i is less than half of the probability of the first (best) parse tree generated for T_i , no more parse trees are considered for T_i .

3 Probability of an Interpretation

Scusi? ranks candidate ICGs according to their probability of being the intended meaning of a spoken utterance. The principles of this calculation were set out in [6]. Here we refine this process, focusing on the calculation of the probability of ICGs.

Given a speech signal W and a context \mathcal{C} , the probability of an ICG I is represented as follows.

$$\Pr(I|W, \mathcal{C}) \propto \sum_{\Lambda} \Pr(I|U, \mathcal{C}) \cdot \Pr(U|P) \cdot \Pr(P|T) \cdot \Pr(T|W) \quad (1)$$

where the UCG, the parse tree and the textual interpretations are denoted by U , P and T respectively. The summation is taken over all possible paths $\Lambda = \{P, U\}$ from the parse tree to the ICG, because a UCG and an ICG can have more than one parent. The ASR and the parser return an estimate of $\Pr(T|W)$ and $\Pr(P|T)$ respectively. In addition, $\Pr(U|P) = 1$, since the process of generating a UCG from a parse tree is deterministic. Hence, we still have to estimate $\Pr(I|U, \mathcal{C})$.

Consider an ICG I containing concepts $c^{ICG} \in \Omega_c$ and relations $r^{ICG} \in \Omega_r$ (Ω_c and Ω_r are the concepts and relations in the domain knowledge respectively). The parent UCG (denoted by U) comprises concepts $c^{UCG} \in \Gamma_c$ and relations $r^{UCG} \in \Gamma_r$ (Γ_c and Γ_r are the concepts and relations from which UCGs are built). The probability of I given U and context \mathcal{C} can be stated as follows.

$$\begin{aligned} \Pr(I|U, \mathcal{C}) &= \prod_{\substack{c^{ICG} \in \Omega_c \\ r^{ICG} \in \Omega_r}} \Pr(c^{ICG}, r^{ICG} | c^{UCG}, r^{UCG}, \Omega_c^-, \Omega_r^-, \mathcal{C}) \\ &= \prod_{\substack{c^{ICG} \in \Omega_c \\ r^{ICG} \in \Omega_r}} \left\{ \Pr(r^{ICG} | c^{ICG}, c^{UCG}, r^{UCG}, \Omega_c^-, \Omega_r^-, \mathcal{C}) \times \Pr(c^{ICG} | c^{UCG}, r^{UCG}, \Omega_c^-, \Omega_r^-, \mathcal{C}) \right\} \end{aligned} \quad (2)$$

where c^{UCG} and r^{UCG} denote the UCG concept and relation corresponding to the ICG concept c^{ICG} and relation r^{ICG} respectively; and Ω_c^- and Ω_r^- denote the sets Ω_c and Ω_r without the concept c^{ICG} and relation r^{ICG} respectively.

It is difficult to estimate Equation 2, as each concept and relation in an ICG depends on the other ICG concepts and relations. We therefore make the following simplifying assumptions.

- The probability of an ICG relation r^{ICG} depends only on the corresponding UCG relation, the parent ICG concept of r^{ICG} , and the context.
- The probability of an ICG concept c^{ICG} depends only on the corresponding UCG concept, the parent ICG relation and grandparent ICG concept of c^{ICG} , and the context (e.g., the parent relation of `mug03` in the ICG in Figure 1(b) is *Patient*, and its grandparent concept is `put01`).

These assumptions are justified by the information in the knowledge base, which stores the location and ownership of many objects, and by the available linguistic information regarding concepts and relations (e.g., the action `fetch01` has a mandatory *Patient* relation, but an optional *Beneficiary*, and any mug is a suitable *Patient* for most actions). Now, say we have the request “get the mug from the table”, and one of the candidate ICGs has the fragment [`mug03 → Location → table01`] (*Location* is the parent of `table01`, and `mug03` is its grandparent). If `mug03` is indeed on `table01`, the probability of this ICG increases, otherwise it decreases. In the absence of this information, we back off to bigram probabilities (e.g., whether `table01` is a possible *Location*). These assumptions yield

$$\Pr(I|U, \mathcal{C}) \approx \prod_{\substack{c^{ICG} \in \Omega_c \\ r^{ICG} \in \Omega_r}} \left\{ \Pr(r^{UCG}|r^{UCG}, c_p^{ICG}, \mathcal{C}) \times \Pr(c^{ICG}|c^{UCG}, r_p^{ICG}, c_{gp}^{ICG}, \mathcal{C}) \right\} \quad (3)$$

where the parent concept of relation $r^{ICG} \in \Omega_r$ is $c_p^{ICG} \in \Omega_c$, and the grandparent concept and parent relation of concept $c^{ICG} \in \Omega_c$ are $c_{gp}^{ICG} \in \Omega_c$ and $r_p^{ICG} \in \Omega_r$ respectively.

After applying Bayes rule, and making additional simplifying assumptions about conditional dependencies, we obtain

$$\Pr(I|U, \mathcal{C}) \approx \prod_{\substack{c^{ICG} \in \Omega_c \\ r^{ICG} \in \Omega_r}} \left\{ \underbrace{\frac{\Pr(r^{UCG}|r^{ICG})}{\Pr(c^{UCG}|c^{ICG})}}_{\text{segment 1}} \underbrace{\frac{\Pr(r^{ICG}|c_p^{ICG})}{\Pr(c^{ICG}|r_p^{ICG}, c_{gp}^{ICG})}}_{\text{segment 2}} \underbrace{\Pr(c_p^{ICG}|\mathcal{C})}_{\text{segment 3}} \right\} \quad (4)$$

The first segment in Equation 4 represents the probability that a user who intended r^{ICG} and c^{ICG} said r^{UCG} and c^{UCG} respectively; the second segment represents the probabilities of relations and concepts in the ICG in light of their parent and grandparent nodes; and the third segment represents the prior probabilities of the concepts in the ICG (judicious conditionalization obviates the calculation of prior probabilities of relations in the ICG). Ideally, all these probabilities should be estimated from data, but this would require the development of a large database of UCGs and ICGs corresponding to different utterances. Such a database is currently not available. Hence, *Scusi?* estimates the necessary probabilities using functions which give a probabilistic interpretation to the closeness between requested features and features of the candidate concepts and relations. These functions are described in detail in [7,4]. Here we outline our approach.

- The probabilities in the first segment of Equation 4 are estimated on the basis of the goodness of the match between candidate instantiated concepts (relations) in the ICG and concepts (relations) mentioned in the UCG [4]. For relations, this probability depends on the lexical match between a stated relation and an instantiated

relation. For concepts we also take into account the left modifiers of the head noun — at present we consider colour and size. For example, if the user said “blue mug”, then a light blue cup will yield a lower probability than a royal blue mug.

- The probabilities in the second segment are estimated based on how well children nodes match the expectations of their parent (and grandparent) nodes in the ICG [7]. For example, the probability of the ICG bigram [$g_002 \rightarrow Destination$] depends on whether *Destination* is a compulsory complement of g_002 (high probability) or optional (lower probability); the probability of the trigram [$cup_05 \rightarrow Owned-by \rightarrow Susan01$] is 1 if *Susan01* owns *cup05*, and 0.5 if ownership is unknown. At present, grandparent concepts are considered only for location and ownership of objects, which may be determined from the system’s knowledge base. In other cases or if the information is unknown, we back-off to the parent relation of a concept, e.g., the probability of *kitchen* being a *Location*.
- The prior probability of an instantiated concept depends on the context. At present, the context includes only domain knowledge, i.e., all instantiated concepts have the same prior, hence it does not affect the performance of the system. However, visual and dialogue context will come into play when *Scusi?* interacts with the robot’s vision system and *DORIS*’s dialogue module. To support these interactions, in the future, we propose to estimate the prior probabilities of concepts by combining salience scores obtained from dialogue history [8] with visual salience [9].

4 Generating ICGs

The process of generating ICGs from a UCG and estimating their probability is carried out by Algorithm 1, which refines the procedure presented in [6]. This algorithm generates a buffer containing up to k_{max} ($= 400$) ICGs ranked in descending order of probability the first time a UCG is expanded (the size of the buffer was empirically determined). Every time a new ICG is requested for that UCG, the next ranked ICG is returned. The algorithm has two main stages: *concept and relation postulation* (Steps 2–10), and *ICG construction* (Steps 11–16).

4.1 Postulating Concepts and Relations

In this stage, the algorithm proposes instantiated concepts (relations) from the knowledge base for each UCG concept (relation), and sorts each candidate list of instantiated concepts (relations) in descending order of probability.

In **Step 5**, for each concept c^{UCG} in the UCG, the algorithm estimates the probability that each instantiated concept in the knowledge base matches c^{UCG} . The same is done for relations. The probability of this match, which corresponds to the first segment in Equation 4, is estimated by means of comparison functions [4]. The probability of a match between an instantiated relation and a UCG relation depends only on the goodness of the lexical match between these relations. In contrast, the probability of a concept match also depends on the match between intrinsic features mentioned in the UCG, such as colour and size, and the actual values of these features for a candidate instantiated concept. For instance, given the UCG concept *cup*, the instantiated concepts *mug01*, . . . ,

Algorithm 1. Generate candidate ICGs for a UCG

Require: UCG U comprising concepts c^{UCG} and relations r^{UCG} , context \mathcal{C}

- 1: Initialize buffer \mathcal{I}_U of size k_{\max} (=400)
- { **Postulate concepts and relations for UCG }**
- 2: **for all** concepts c^{UCG} (relations r^{UCG}) in U **do**
- 3: Initialize a list of candidate concepts $L_{c^u} \leftarrow \emptyset$ (list of relations $L_{r^u} \leftarrow \emptyset$)
- 4: **for all** instantiated concepts c^I (instantiated relations r^I) **do**
- 5: Compare c^{UCG} with c^I (r^{UCG} with r^I), yielding a probability for the match (segment 1 in Equation 4)
- 6: Calculate the prior probability of c^I according to the context \mathcal{C} (segment 3 in Equation 4)
- 7: Multiply the probabilities obtained in Steps 5 and 6
- 8: Insert c^I in the list L_{c^u} (r^I into L_{r^u}) in descending order of probability
- 9: **end for**
- 10: **end for**
- { **Construct ICGs }**
- 11: **for** $j = 1$ to k_{\max} **do**
- 12: Generate the “next best” ICG I_j by going down each list L_{c^u} and L_{r^u} in turn
- 13: Perform internal consistency checks to calculate the probabilities of the bigrams and trigrams in ICG I_j (segment 2 in Equation 4)
- 14: Estimate $\Pr(I_j|U, \mathcal{C})$ by multiplying the probabilities obtained in Step 7 with the probabilities obtained in Step 13
- 15: Insert I_j into buffer \mathcal{I}_U in descending order of probability
- 16: **end for**

mug05 and cup01, ..., cup04 have a good lexical match with the UCG concept. If the UCG concept had been *blue cup*, then the colour coordinates of the mugs and cups in the knowledge base would be matched against the coordinates for the term ‘blue’.

Upon completion of Step 5, we prune ICG candidates that do not have a good match with the concept (relation) in the UCG. For example, the UCG concept *chair* could refer to an armchair, a stool, a pouf, etc. Hence, all the armchairs, stools, poufs, etc in the knowledge base are retained for further processing, while lamps, tables, cups, etc are discarded. Similarly, red cups are discarded if a blue mug is requested, and there are blue mugs in the knowledge base.

The prior probability of the retained candidate concepts (third segment in Equation 4) is estimated in **Step 6**. In **Step 7**, this probability is multiplied by the probability calculated in Step 5.

This stage of the algorithm yields a list of candidate instantiated concepts L_{c^u} for each UCG concept c^{UCG} , and a list of candidate instantiated relations L_{r^u} for each UCG relation r^{UCG} . These lists are sorted in descending order of probability. For example, Table 1 shows the sorted lists of concepts and relations postulated for the request in Figure 1(b) “leave the blue mug on the table”: there are four objects that are a good match for the concept *blue mug*, three candidate tables, three candidate actions for *leave* (leave the room (leave01), put in a specific place (put01), and put down (put02)), two relations for *on*, and one for *object*.

Table 1. Concepts and relations used to build ICGs for the utterance “leave the blue mug on the table”

<i>leave</i>	<i>blue mug</i>	<i>table</i>	<i>on</i>	<i>object</i>
leave01	mug02	table01	<i>Destination</i>	<i>Patient</i>
put01	cup01	table02	<i>Location</i>	
put02	mug03	table03		
		cup02		

4.2 Constructing ICGs

In this stage, the algorithm uses the list of instantiated concepts (relations) built for each concept (relation) in a UCG to construct candidate ICGs for this UCG, and sorts these ICGs in descending order of probability. First, **Step 12** applies an enumerative process to generate different combinations of concepts and relations from the list L_{c_u} (L_{r_u}) maintained for each UCG concept (relation). This is done by iteratively selecting one candidate concept (relation) from each list. For instance, the concepts and relations in Table 1 are combined as follows to build candidate ICGs. First, the top line $\{\text{leave01}, \text{mug02}, \dots\}$, which has the highest probability, is used. The next four combinations are generated by replacing one element from this line at a time, i.e., leave01 is replaced with put01 , yielding $\{\text{put01}, \text{mug02}, \dots\}$; then mug02 is replaced with cup01 , yielding $\{\text{leave01}, \text{cup01}, \dots\}$; and so on.

The probabilities of the bigrams and trigrams in each ICG (second segment in Equation 4) are then estimated in **Step 13**. These probabilities reflect the extent to which the relationships between neighbouring nodes in an ICG match the known reality. As mentioned in Section 3, for relations this calculation is done based on the type of relations admitted by each concept (e.g., compulsory, optional or absent), and for concepts the calculation reflects the current state of the world. For example, if we request “the mug on the table” and according to the knowledge base, mug03 is on table02 , the probability of an ICG that contains $[\text{mug03} \rightarrow \text{Location} \rightarrow \text{table02}]$ is increased, whereas if cup01 is not on a table, the probability of an ICG containing cup01 is decreased. In **Step 14**, these ‘structural’ probabilities are combined with the probability calculated in Step 7 (candidate postulation stage) to obtain the final probability of an ICG produced for a given UCG. This ICG is inserted in the buffer for that UCG in descending order of the ICG’s probability.

5 Evaluation

Our evaluation test set comprised 100 utterances: 43 declarative (e.g., “the book is on the desk”, “in the kitchen”, “the red mug”) and 57 imperative (e.g., “open the door”).¹ These utterances were based on interactions between users and a “robot” (enacted by

¹ We acknowledge the modest size of this test set compared to that of some publicly available corpora, e.g., ATIS and GeoQuery. However, we must generate our own test set since our task differs significantly from the slot-filling tasks where these large corpora are used. This is due to the domain itself and the open-ended nature of the utterances.

one of the authors) in a virtual home scenario; they were recorded by one of the authors, as the ASR software is speaker dependent, and at present we do not handle features of spontaneous speech. The utterances (which were not used during system development) were chosen to test *Scusi?*'s ability to identify target objects (the intended book, mug, table, etc), and its ability to handle phenomena such as synonyms (e.g., "wash" and "clean") and homonyms (e.g., "leave the mug on the table" versus "leave the room"). The average utterance length was 8.5 words, with a maximum length of 12 words.

Scusi? was set to generate at most 300 sub-interpretations in total (including texts, parse trees, UCGs and ICGs) for each utterance in the test set; on average, it took 6 seconds to go from parse trees to ICGs. An interpretation was deemed successful if it correctly represented the speaker's intention within the limitations of *Scusi?*'s knowledge base, which comprises 135 items (24 relations and 111 concepts, which consist of abstract concepts and concrete objects normally found in a house). This intention was represented by one or more *Gold ICGs* that were manually constructed by one of the authors. Multiple *Gold ICGs* were allowed if there were several objects in the knowledge base that matched a requested object, e.g., "get a mug".

Ideally, we would like to evaluate separately the impact of our probabilistic framework and that of maintaining multiple interpretations. However, the design of an alternative, baseline hypothesis-ranking framework is outside the scope of this project. We therefore designed our experiments to measure (1) *Scusi?*'s overall interpretation performance, (2) the impact of maintaining multiple interpretations on performance, and (3) the impact of different thresholds (Section 2). Thus, tests were conducted under the following settings.

- BASELINE – a beam search was executed, where only the best ASR result was parsed, and only the best parse tree yielded a UCG. We then selected the top ICG among those in the buffer for the UCG (Section 4). Note that the selection of the top-ranked item for each of these stages is still done on the basis of its probability. Hence, our baseline enables us to isolate only the impact of maintaining multiple interpretations.
- Threshold – No Threshold, and thresholds of 10%, 20%, 50%, 80% and 90%. For instance, given a 10% threshold, *Scusi?* stops expanding a text T such that $\Pr(\text{current parse tree for } T) < 0.1 \times \Pr(\text{highest-probability parse tree for } T)$.

Table 2 summarizes our results, which were obtained with an ASR that had a 20% error rate (the correct text was top ranked by the ASR in 80% of the cases). Column 1 displays the test condition (baseline and threshold value). Columns 2-3 show how many utterances had *Gold ICGs* whose probability was among the top 1 or top 3, e.g., the 20% threshold yielded 70 *Gold ICGs* with the highest probability (top 1), and 83 within the top 3 probabilities. The average *adjusted rank* and *rank* of the *Gold ICG* appear in Column 4. The rank of an ICG I is its position in a list sorted in descending order of probability (starting from position 0), such that all equiprobable ICGs are deemed to have the same position (recall that the baseline returns a single ICG, whose rank is therefore 0). The adjusted rank of an ICG I is the mean of the positions of all ICGs that have the same probability as I . For example, if we have 3 top-ranked equiprobable ICGs, each has a rank of 0, but an adjusted rank of $\frac{0+2}{2}$. Column 5 shows how many

Table 2. *Scusi?*?s interpretation performance

	# Gold ICGs with prob in top 1	Average adj rank (rank)	Not found	Avg # of ICGs to Gold ICG (avg # of iters)
	top 3			
BASELINE	53	53	0 (0)	47
No Thrsh	69	82	3.85 (1.15)	7
10%	67	81	2.63 (0.91)	8
20%	70	83	2.47 (0.87)	7
50%	70	84	2.37 (0.81)	7
80%	70	85	2.31 (0.80)	7
90%	70	85	2.31 (0.78)	7
Total	100	100		

utterances didn't yield a Gold ICG, and Column 6 indicates the average number of ICGs created and iterations done until the Gold ICG was found (from a total of 300 iterations).

As seen in Table 2, the baseline yielded significantly fewer top-ranked Gold ICGs than our anytime algorithm ($p < 0.05$).² The 20% ASR error was substantially exacerbated by the baseline approach, which failed to return Gold ICG(s) in 27 of the 80 cases where it was presented with the correct text. In contrast, *Scusi?*? performed significantly better, failing to produce top-ranked Gold ICG(s) in only 10 of those cases. Further, for the top-3 ranking, *Scusi?*? overcame the ASR error (i.e., its error rate is less than 20%).³ These results confirm the need to maintain multiple interpretations in combination with a probabilistic hypothesis assessment.

Interestingly, the threshold did not affect the number of top-ranked Gold ICGs and not found ICGs, and the number of iterations to Gold. However, the average rank of the Gold ICGs decreases (improves) as the threshold increases, which is consistent with the slight improvement in the number of top-3 ICGs. We also performed additional experiments that examined the effect of using a different threshold for each level of interpretation. However, the new scheme did not yield any improvement over a single system-wide threshold.

6 Related Research

This research builds on the work described in [5,6]. The contributions of this paper pertain to (1) the anytime algorithm and processing threshold (Section 2); (2) the probabilistic calculation of the factors of $\Pr(I|U, \mathcal{C})$, instead of using heuristics (Section 3); and (3) the modifications of the UCG representation to cater for intrinsic features of a node (Section 2), and ensuing changes in the algorithm for generating ICGs (Section 4). These modifications led to an improved performance of the system, which was evaluated on a larger corpus than that used in [6].

² Sample paired t-tests were used for all statistical tests.

³ Clearly, it is not fair to compare *Scusi?*?s top-3 rank with the baseline's top-1 rank. However, the top-3 rank supports the generation of clarification questions for ICGs with similar probabilities — an option that is not available if only the top-ranked interpretation is returned.

Many researchers have investigated numerical approaches to the interpretation of spoken utterances in dialogue systems, e.g., [10,11,12,13]. Pfleger *et al.* [10] and Hüwel and Wrede [11] employ modality fusion to combine hypotheses from different analyzers (linguistic, visual and gesture), and apply a scoring mechanism to rank the resultant hypotheses. In contrast, He and Young [12] and Gorniak and Roy [13] apply a probabilistic approach to spoken language interpretation, using Hidden Markov Models for the ASR stage. Additionally, as mentioned above, *Scusi?*'s probabilistic formalism resembles in style that employed by Miller *et al.* for discourse interpretation in a text-based system [1]. However, all these systems employ semantic grammars, while *Scusi?* employs a three-stage interpretation process, which uses generic, syntactic tools, and incorporates semantic- and domain-related information only in the final stage of the interpretation process. Knight *et al.* [14] compare the performance of a dialogue system based on a semantic grammar to that of a system based on a statistical language model and a robust phrase-spotting grammar. The latter performs better for relatively unconstrained utterances by users unfamiliar with the system. The probabilistic approach and intended users of our system are in line with this finding.

From the view point of application domain, robot-mounted dialogue systems were also studied in [15,16,11]. Matsui *et al.* [15], like Gorniak and Roy [13], use contextual information to constrain the alternatives returned by the ASR early in the interpretation process. This allows their system to process expected utterances efficiently, but makes it difficult to interpret unexpected utterances. In contrast, *Scusi?* incorporates contextual information in the final stage of the interpretation process. Unlike *Scusi?*'s probabilistic reasoning formalism, Bos *et al.* [16] use a logic-based language interpretation framework to understand instructions and descriptions, and employ formal proofs for conflict resolution. Consequently, alternatives are not considered when an utterance is ambiguous or the preferred option proves undesirable. This is also the case for Hüwel and Wrede's [11] system, which considers only a single alternative as a result of each stage of the interpretation process.

7 Conclusion and Future Work

We have described *Scusi?*, a spoken language interpretation system that maintains multiple options at each stage of the interpretation process, and ranks interpretations based on estimates of their posterior probability. In particular, we presented the algorithm used by *Scusi?* to postulate hypotheses regarding the meaning of a spoken utterance, and detailed the estimation of the probabilities of these hypotheses.

Our empirical evaluation shows that *Scusi?* performs well for declarative and imperative utterances of varying length, with the Gold ICG(s) receiving one of the top three probabilities for most test utterances. Our results also show that using a threshold has a small impact on *Scusi?*'s performance (by slightly improving the number of ICGs ranked top-3, and hence the average rank of the Gold ICGs). In the near future, we will further investigate the impact of thresholds on performance speed and accuracy. An additional avenue of investigation pertains to the consideration of different weightings for combining the scores obtained from the three interpretation stages.

References

1. Miller, S., Stallard, D., Bobrow, R., Schwartz, R.: A fully statistical approach to natural language interfaces. In: ACL 1996 – Proceedings of the 34th Conference of the Association for Computational Linguistics, Santa Cruz, California, pp. 55–61 (1996)
2. Sowa, J.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading (1984)
3. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. *Computational Linguistics* 28(3), 245–288 (2002)
4. Zukerman, I., Makalic, E., Niemann, M.: Using probabilistic feature matching to understand spoken descriptions. In: AI 2008 Proceedings – the 21st Australasian Joint Conference on Artificial Intelligence, Auckland, New Zealand (2008)
5. George, S., Zukerman, I., Niemann, M., Marom, Y.: Considering multiple options when interpreting spoken utterances. In: Proceedings of the 5th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Hyderabad, India, pp. 7–14 (2007)
6. Niemann, M., Zukerman, I., Makalic, E., George, S.: Hypothesis generation and maintenance in the interpretation of spoken utterances. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 466–475. Springer, Heidelberg (2007)
7. Makalic, E., Zukerman, I., Niemann, M., Schmidt, D.: A probabilistic model for understanding composite spoken descriptions. In: Ho, T.-B., Zhou, Z.-H. (eds.) PRICAI 2008. LNCS (LNAI), vol. 5351, pp. 581–592. Springer, Heidelberg (2008)
8. Zukerman, I., George, S.: A probabilistic approach for argument interpretation. *User Modeling and User-Adapted Interaction, Special Issue on Language-Based Interaction* 15(1-2), 5–53 (2005)
9. Wyatt, J.: Planning clarification questions to resolve ambiguous references to objects. In: Proceedings of the 4th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Edinburgh, Scotland, pp. 16–23 (2005)
10. Pfleger, N., Engel, R., Alexandersson, J.: Robust multimodal discourse processing. In: Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue, Saarbrücken, Germany, pp. 107–114 (2003)
11. Hüwel, S., Wrede, B.: Spontaneous speech understanding for robust multi-modal human-robot communication. In: Proceedings of the COLING/ACL Main conference poster sessions, Sydney, Australia, pp. 391–398 (2006)
12. He, Y., Young, S.: A data-driven spoken language understanding system. In: ASRU 2003 – Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, St. Thomas, US Virgin Islands, pp. 583–588 (2003)
13. Gorniak, P., Roy, D.: Probabilistic grounding of situated speech using plan recognition and reference resolution. In: ICMI 2005 – Proceedings of the Seventh International Conference on Multimodal Interfaces, Trento, Italy, pp. 138–143 (2005)
14. Knight, S., Gorrell, G., Rayner, M., Milward, D., Koeling, R., Lewin, I.: Comparing grammar-based and robust approaches to speech understanding: A case study. In: EUROSPEECH 2001 – Proceedings of the Seventh European Conference on Speech Communication and Technology, Aalborg, Denmark, pp. 1779–1782 (2001)
15. Matsui, T., Asoh, H., Fry, J., Motomura, Y., Asano, F., Kurita, T., Hara, I., Otsu, N.: Integrated natural spoken dialogue system of Jijo-2 mobile robot for office services. In: AAAI 1999 – Proceedings of the Sixteenth National Conference on Artificial Intelligence, Orlando, Florida, pp. 621–627 (1999)
16. Bos, J., Klein, E., Oka, T.: Meaningful conversation with a mobile robot. In: EACL10 – Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics, Budapest, Hungary, pp. 71–74 (2003)

Towards Autonomous Robot Operation: Path Map Generation of an Unknown Area by a New Trapezoidal Approximation Method Using a Self Guided Vehicle and Shortest Path Calculation by a Proposed SRS Algorithm

K. Ahmed, M.S. Munir, A.S.M Shihavuddin, M.A. Hoque, and K.K. Islam

Department of Electrical and Electronic Engg. Islamic University of Technology (IUT)
Board Bazar, Gazipur 1704, Bangladesh

{kabir_89, smunir}@iut-dhaka.edu, shihav407@yahoo.com,
ahoque@yahoo.ca, kkislam@iut-dhaka.edu

Abstract. This paper deals with the road map generation of an unknown environment by an autonomous vehicle using a proposed trapezoidal approximation. Subsequently a novel shortest path calculation method named smallest road segment (SRS) detection method has been proposed. At first we have generated a blind map of an unknown environment in a computer. Image of the unknown environment is captured by the vehicle and sent to the computer using wireless transmitter module. The image is pre-processed and the detected road boundaries help to update the blind map with road positions. When the complete map having all possible road branches is generated then based on the source and destination point, the shortest path is calculated using the proposed SRS method. This shortest path is forwarded to the vehicle to reach at the destination through the best path available.

Keywords: Edge detection, Sobel filter, Hough transform, map generation, shortest path calculation.

1 Introduction

This paper is related to the map generation of the roads of an unknown environment and finding shortest path in between two points using a self guided vehicle. Various methods are proposed in the past for road detection. A single camera with Conic Projection Sensor System has been used in [1], while a two camera system has been used in [2] to get real-time images and the 3-D perspective is removed for vehicle guidance. In [3], Hough Transform has been used to extract 3-D road boundaries in the images coming from a stereo-vision system. Geometrical features are used in [4] to find road boundaries. A multi order line segment description of the road boundary has been achieved through a statistical test, in [5]. In [6], low level structures are extracted via a non-linear transform and it proposes an algorithm for segmentation of images at multiple scales. In [7], a fast edge detection method is proposed. Also different techniques of vehicle guidance system in detected road are proposed in the past [8-13].

These papers concentrate on road detection by discarding the sky, plant, vehicle etc with complex calculation. In our paper, a different simpler area thresholding technique is used for discarding anything other than road and a new trapezoidal approximation is used for map generation. We have used a vehicle which is self controlled by detected road parameters. A wireless video camera is installed on the robot which takes the front view video of the robot and transmits it back to a PC. In the PC the video is analyzed to detect the road. At first, the vehicle moves from the source through the unknown environment having approximated boundary. The unknown environment is modeled using a base matrix. Image frames are extracted from the video, converted to grayscale image and noise is reduced by using a 3×3 median filter [14]. This image is converted into binary image. By applying appropriate area threshold, only the road area is kept in the binary image. Then the edge is detected using Sobel method followed by detection of the continuous straight lines and curves of the edges using Hough transform [15]. Then we approximated a trapezium section of the road having defined length. The vehicle then moves through the trapezium and reaches the next approximated trapezium having a tilt angle with the previous one. According to the distances moved by the vehicle and corresponding tilt angles, the base matrix is updated. As the vehicle explores all possible roads of the unknown environment, the complete map of the unknown environment is finally obtained from the base matrix.

Once the complete map is generated, the shortest path from the source to destination on the map is estimated. Different methods are proposed in the past to find shortest path from such model [16-19]. In our case we have used a new clustering algorithm for shortest path detection where detected roads are divided into clusters having individual identity. Each junction points are also assigned identity. Based on those, the shortest path is calculated.

To avoid collision with any obstacle of the environment, an IR based obstacle detector is used [20]. As the vehicle can move in the environment without any collision, it can generate the complete map of the environment for subsequent shortest path estimation. The complete system is represented as block diagram in Fig. 1.

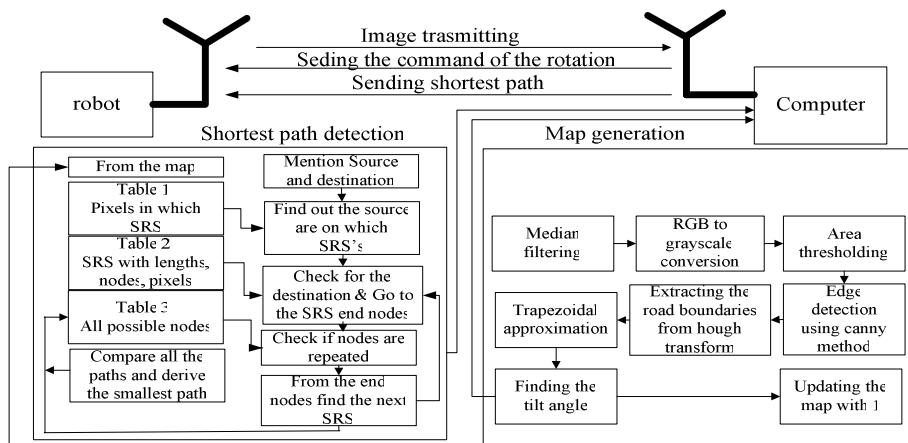


Fig. 1. Block diagram representation of proposed system

2 Data Set

The environment is represented by four states - *SOURCE* = “*”, *DESTINATION* = “#”, *UNKNOWN* = “-” and *ROAD* = “1” for the analysis of all the data concerned with the robot like its motion, dimension, decision and detected road. We have used the following two data matrixes to represent all those data,

1. **Base matrix:** Initially all the points of Base matrix will be set to “-” except “*” and “#”. As the vehicle moves through the road and generates Progression Set, those data will be converted into X and Y coordinate values starting from “*” and each detected road coordinate is marked as “1”.
2. **Progression matrix:** Initially the vehicle will move from the source to destination by detecting the unknown road. Each time, a segment of the road having length l will be selected through which the vehicle will move and will enter into the next segment having a tilt angle with the previous segment. Every tilt angle will be stored in Progression matrix.

3 Map Generation

Path map of unknown environment is generated by the vehicle as it moves along the road with and creates a database of the motion. The movement has memory of order 1, i.e. next step of the decision is based upon the last one. The path is assumed to have very small linear approximated segment of length l . Extracting road edges from unknown environment image is difficult to do when road surface is not homogeneous as inhomogeneities appear as noise. To eliminate them, we ran a Median filtering [21] on the road image. Let the image be represented as $I(x, y)$, with x and y being the two coordinate directions. Median filter can be represented as,

$$v(m, n) = \text{median}\{I(m - k, n - l), (k, l) \in W\} \quad (1)$$

Where, W is a suitably chosen window.

The median filtered image is then converted into binary image where only the road, plants having significant color discontinuity and sky areas are detected. Using the road area as threshold, we eliminated all other smaller detected areas as,

$$\sum I_{ROAD}(x, y) > \sum I_{SKY}(x, y), \sum I_{PLANT}(x, y) \quad (2)$$

Sobel filter [22], are used for boundary extraction. Performances of both the filters are almost similar. The output image after boundary extraction is represented as:

$$E(x, y) = I(x, y) \times h(x, y) \quad (3)$$

Where, $E(x, y)$ is the image data containing extracted edge, $I(x, y)$ is the area thresholded binary image data, $h(x, y)$ is the filter matrix.

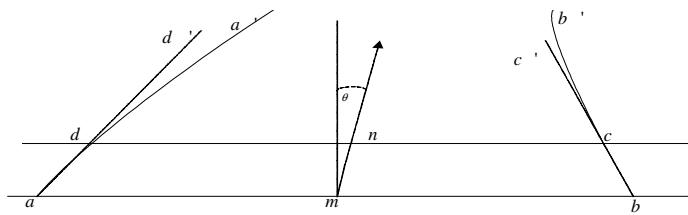


Fig. 2. Linear straight line approximation of road boundaries

To apply our proposed trapezium approximation method for road segmentation, we need straight road boundaries. In our case, we used Hough transform for detecting straight lines describing the road boundaries in the edge detected binary image. To move along detected road, rotation angle is required. For this, a portion from the processed image of particular length l is taken. Which is approximated as trapezium.

In the Fig. 2, ada' and bcb' are the road boundaries, where $abcd$ is considered to be trapezium. Direction vector for motion of vehicle is given by vector indicated by mn and the angle to rotate is the angle between mn and the normal drawn at the point ' m '. With the constant distance l and angle, the vehicle will be guided through the path. According to the movement, corresponding map is generated in the remote computer.

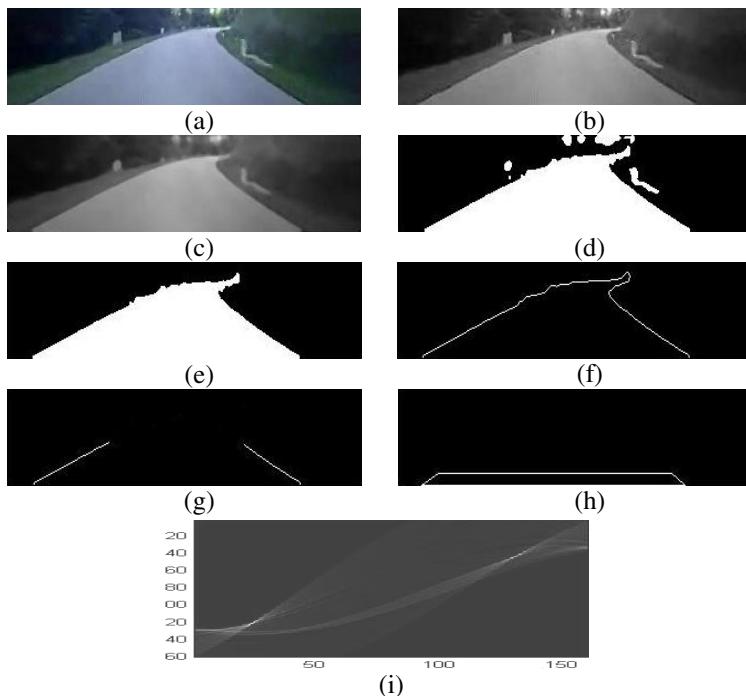


Fig. 3. (a) Original image, (b) Gray scale image, (c) Median filtered image, (d) Binary image, (e) Area thresholded image, (f) Edge detected image, (g) Detected straight line by Hough transform, (h) Trapezoidal approximation, (i) Hough accumulator space

When the trapezium is available from the road image, we can extract the decision parameter for the vehicle movement according to the following geometrical analysis:

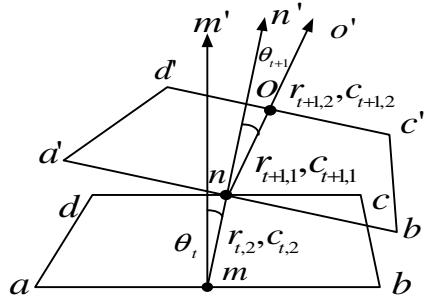


Fig. 4. Calculation of the decision parameter for the vehicle movement

At Fig. 2. two consecutive road frames are placed together to make the analysis. The trapezium $abcd$ represent the road segment at time 't' and $a'b'c'd'$ represents for time ' $t+1$ '. 'r' and 'c' denotes the row and column number of the image matrix. For t time frame the angular motion towards mnn' can be derived from the ' m ' and ' n ' pixels row and column position ($r_{t,1}, c_{t,1}$) and ($r_{t,2}, c_{t,2}$) according to below relation

$$\theta_t = 90 - \tan^{-1} \left(\frac{r_{t,2} - r_{t,1}}{c_{t,2} - c_{t,1}} \right) \quad (4)$$

Where, θ_t is the angle of the rotation. As the length to be advance is predefined and fixed so no further calculation is required for the wheel motion of the vehicle.

Similarly after the completion of the auto generated command for the time frame ' t ', the next road frame for the decision moment of $t+1$ will be like the trapezium $a'b'c'd'$ and the angle to be rotated will be equal to

$$\theta_{t+1} = 90 - \tan^{-1} \left(\frac{r_{t+1,2} - r_{t+1,1}}{c_{t+1,2} - c_{t+1,1}} \right) \quad (5)$$

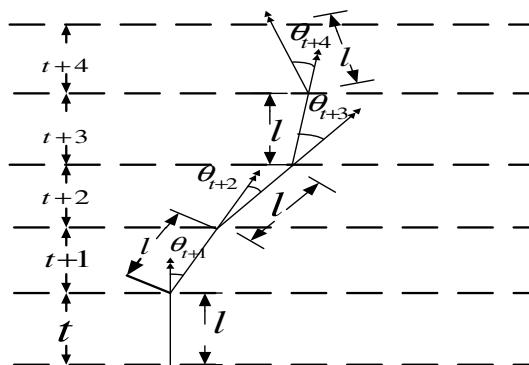


Fig. 5. Calculation of the value of Progression matrix

At every decision moment we have the angle at which a straight line of length l will tilted. Along this line of movement, the map initiated in the base matrix will be updated with “1” values.

If the current position of the robot is (x_i, y_i) then at the next moment it will be in (x_{i+1}, y_{i+1}) . From (x_i, y_i) , (x_{i+1}, y_{i+1}) is calculated using the following equations,

$$\sqrt{((x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2)} = l \quad (6)$$

$$\frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \tan(90 - \theta_i) \quad (7)$$

Base matrix is updated as 1 using position of (x_{i+1}, y_{i+1}) . As the vehicle reaches to destination, the explored path map is saved. This map of the unknown environment is used in the later for finding the shortest path.

4 Shortest Path Detection

Image of the map is divided into sets of smallest straight line in between all the nodes where no turning is required. Nodes are those points on the map from where there are possible chances to change direction or move from one SRS to others.

The image of the road map is deformed into three sets of information data types. From the binary image where only paths are denoted by “1”, the first set of data is derived, which is the representation of all the valid pixels on road by an individual integer value. In the $M \times N$ matrix any valid path position is named as following

$$\text{Binary Image} = \begin{bmatrix} P(0,0) & \dots & \dots & P(0, M-1) \\ P(1,0) & \dots & \dots & P(1, M-1) \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \vdots \\ P(L-1,0) & \dots & \dots & P(L-1, M-1) \end{bmatrix} \quad (8)$$

For every non zero P_{xy} of binary image, we calculated a unique index for their identification later on by the following formula:

$$\text{Unique index of } P_{xy}, P(k) = Mx + y \quad (9)$$

Only if, $P_{xy} = 1$ in the $M \times N$ matrix. The uniquely indexed set then becomes like,

$$\text{Unique index of } P_{xy}, = \begin{bmatrix} P(k=0) & \dots & \dots & P(k=M-1) \\ P(k=M) & \dots & \dots & P(k=2M-1) \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \vdots \\ P(k=\{L-1\}M) & \dots & \dots & P(k=LM-1) \end{bmatrix} \quad (10)$$

The second sets of data needed is the collection of all possible smallest road segments (SRS) which fulfills the following criteria,

1. The segments must be straight line what ever the inclination is.
2. The segments should be in between the all possible nodes.
3. The direction between two consecutive segments may change or may not.

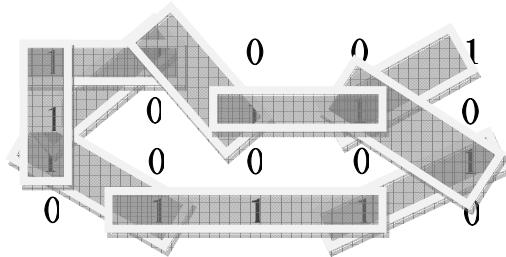


Fig. 6. Deformation of the 5×5 Road map into smallest road segments (SRS)

Mathematically SRS can be expressed as a set consists of points defined as:

$$\bigcup_{z=1}^B (SRS)_z = [P_{xy}, \dots, P_{(x+i)(y+j)}]; \quad (11)$$

Here, B= the total number of SRS, $-m < i < m$ and $-n < j < n$;

The length, L_z of the each SRS can be calculated for different condition.

$$\text{Condition 1: If } i = j, \text{ then } L_z = \sqrt{i^2};$$

$$\text{Condition 2 : If } i = 0 \text{ and } j \neq 0, \text{ then } L_z=j;$$

$$\text{Condition 3 : If } i \neq 0 \text{ and } j = 0, \text{ then } L_z=i;$$

The sets of SRS and their lengths are saved in a lookup table. The nodes are to be calculated to create the second types of data set. Pixel which is common in more than one SRS is a node. The entire sets of pixels that show true result are saved in the lookup table named list of nodes.

$$NODE_i \subset \bigcup_{j=1}^n (SRS)_j; \quad (12)$$

Where, $i = 1, 2, \dots, C$ and $j = \text{index of connected SRSs}$. The 3rd table will keep the database of all the valid pixels with the name of SRS they are situated in. It will help to link the give the possible paths through which the destination can be reached.

After reaching any new node, it scans the connected SRS and proceeds through that avoiding repetition. Doing this process, the destination will be reached following one or more possible solutions. In each solution, possible path is scanned out. In this each possible paths the length of all the SRS though which destination is reached will

be summed up to calculate the total path distance. And the one with the minimum length is selected as the shortest one.

5 Obstacle Detection

In the obstacle detection system, one microcontroller is used to generate a 50 Hz pulse signal that is to be sent by the Infra-red LED. Another microcontroller was programmed to generate 38 kHz carrier signal. These two signals were modulated using an AND gate and then amplified using an 8050 transistor and finally sent through an IR-LED. If there is any obstacle then the signal will reflect back to the receiving module of the sensor. At the receiving side, the IR receiver module picks up the 38 kHz modulated signal and then demodulates it, to give the original 50 Hz signal to another microcontroller. This microcontroller then matches the received signal with the sent signal. If the microcontroller can match the received signal with the sent signal then it takes the decision that there is an obstacle and moves the vehicle avoid the obstacle.

6 Hardware Implementation

The project was implemented on a mobile robot vehicle platform called MOPHEUS [20]. This platform was driven by two stepper motors 12V 0.12Kwatt each. The microcontroller used is PIC 16F84A. A CCD camera was mounted on the mobile robot platform, and an analog output was connected to the UHF Audio/Video Transmitter unit. While the RS232 RF receiver module connected to a PIC 16F84A I/O ports was used as remote controller.

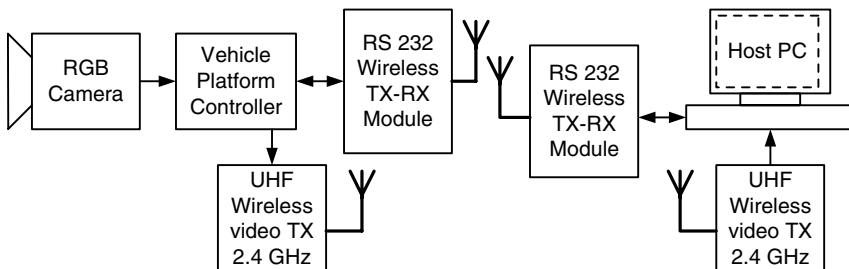


Fig. 7. Block diagram the hardware for the algorithm implementation

The wireless video system used a 2.4 GHz carrier frequency and does not interfere with the RS232 RF data module transmission due to different carrier frequencies. After the images were received into the computer, map is generated and shortest path position is sent via RS232 serial port COM1 to an RS232 RF transmitter unit as a command to the vehicle.

7 Results

Initially the boundary of the unknown environment, source position and destination is specified to generate the blind matrix. Then the vehicle explored all the connected roads of that unknown area and the map is generated accordingly by the proposed trapezoidal approximation method. After the vehicle completes exploring all possible roads and finds the destination, the complete path map of the environment is generated (Fig. 8.a). The generated From the generated map, shortest path is calculated by SRS method (Fig. 8.b).

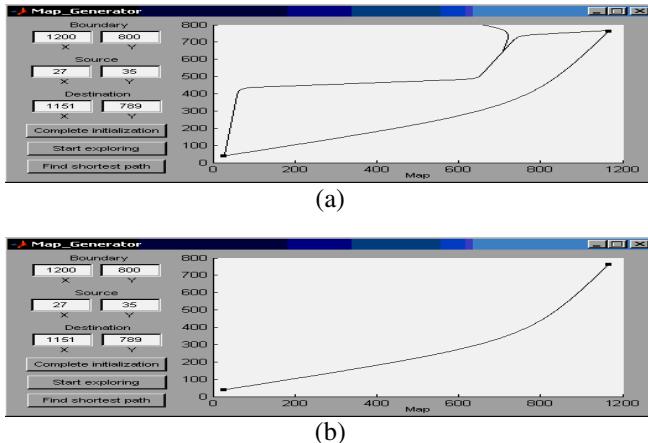


Fig. 8. (a)Map generation, (b) Shortest path calculation

8 Conclusion

In this paper we have applied trapezoidal approximation method for road map generation and a smallest road segment detection method has been devised and applied for the shortest path calculation. As the vehicle initially explores all possible road branches, so the complete map of the environment is generated. Later the shortest path is calculated for the vehicle to follow. During the whole process, the vehicle can operate independently avoiding all obstacles and practically it is implemented using MORPHEUS mobile robot test platform. Practically, the proposed algorithm generates map with 90% accurate road position and the proposed shortest path calculation method can always calculate correctly. Experimental results also validate the proposed SRS algorithm.

References

1. Zhu, Z., Yang, S., Xu, G., Lin, X., Shi, D.: Fast Road Classification and Orientation Estimation Using Omni- View Images and Neural Networks. *IEEE Trans. On Image Processing* 7(8), 1182–1197 (1998)
2. Bertozzi, M., Broggi, A.: GOLD: A parallel realtime stereo vision system for generic obstacle & lane detection. *IEEE Trans. on Image Proc.* 7(1), 62–81 (1998)

3. Sharma, U.K., Davis, L.S.: Road boundary detection in range imagery for an autonomous robot. *IEEE Journal of Robotics and Automation* 4(5), 515–523 (1988)
4. Tarel, J.P., Guichard, F.: Combined dynamic tracking and recognition of curves with application to road detection. In: *Proc. of IEEE Int. Conf. on Image Processing*, vol. 1, pp. 216–219 (September 2000)
5. Tang, G., Liu, X., Liu, X., Yang, H.: A road boundary detection method for autonomous land vehicle. In: *Proc. of the 4th World Congress on Intelligent Control and Automation*, Shanghai, June 2002, vol. 4, pp. 2949–2951 (2002)
6. Tabb, M., Ahuja, N.: Multiscale Image Segmentation by Integrated Edge and Region Detection. *IEEE Trans. on Image Processing* 6(5), 642–655 (1997)
7. Routray, A., Mohanty, K.B.: A Fast Edge Detection Algorithm for Road Boundary Extraction Under Nonuniform Light Condition. In: *Proc. of 10th International Conference on Information*, Rourkela, December 17–20, 2007, pp. 38–40 (2007)
8. Davis, L., Kushner, T.: Road boundary detection for autonomous vehicle navigation. In: *SPIE. Intelligent Robots and Computer Vision*, vol. 579 (1985)
9. Hong, T., Abrams, M., Chang, T., Shneier, M.O.: An Intelligent World Model for Autonomous Off-Road Driving. In: *Computer Vision and Image Understanding* (2000)
10. Rasmussen, C.: Combining Laser Range, Color, and Texture Cues for Autonomous Road Following. In: *Proceedings of the Int. Conference on Robotics and Automation* (2002)
11. Gregor, R., Lützeler, M., Dickmanns, E.D.: EMS-Vision: Combining on- and off-road driving. In: *Proc. SPIE Conf. on Unmanned Ground Vehicle Technology III*, AeroSense 2001, Orlando, FL, USA, April 16–17 (2001)
12. Lieb, D., Lookingbill, A., Thrun, S.: Adaptive Road Following using Self- Supervised Learning and Reverse Optical Flow. In: *Proc. Robotics Science and Systems*, Cambridge, MA, USA, June 8–11 (2005)
13. Stavens, D., Thrun, S.: A Self-Supervised Terrain Roughness Estimator for Off-Road Autonomous Driving. In: *Proc. Conference on Uncertainty in AI (UAI)*, Cambridge, MA, USA, July 13–16 (2006)
14. Hwang, H., Haddad, R.A.: Adaptive Median Filters: New Algorithms and Results. *IEEE Trans. on Image Processing* 4(4), 499–502 (1995)
15. Duda, R.O., Hart, E.P.: Use of the Hough Transformation to detect lines and curves in images. *Grapics and Image Processing* 3, 11–15 (1972)
16. Ausiello, G., Italiano, G.F., Spaccamela, A.M., Nanni, U.: Incremental algorithms for minimal length paths. In: *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, San Francisco, California, United States, January 22–24, 1990, pp. 12–21 (1990)
17. Jing, N., Huang, Y.-W., Rundensteiner, E.A.: Hierarchical optimization of optimal path finding for transportation applications. In: *Proceedings of the fifth international conference on Information and knowledge management*, Maryland, USA, pp. 261–268 (November 1996)
18. Hribar, M.R., Taylor, V.E.: Termination detection for parallel shortest path algorithms. *Journal of Parallel and Distributed Computing* 55(2) (December 1998)
19. Hribar, M.R., Taylor, V.E., Boyce, D.E.: Choosing a shortest path algorithm, Technical Report CSE-95-004, Comp. Sci. and Engg, EECS Department, Northern University (1995)
20. Shihavuddin, A.S.M., Fadlullah, N.M., Islam, K.K.: Micro-controller operated robot using photo sensor guiding system. In: *1st international conference on control, instrumentation and mechatronics*, CIM 2007, Johor, Malaysia (May 2007)
21. Hwang, H., Haddad, R.A.: Adaptive Median Filters: New Algorithms and Results. *IEEE Trans. on Image Processing* 4(4), 499–502 (1995)
22. Scharcanski, J., Venetsanopoulos, A.N.: Edge Detection of Color Images Using Directional Operators. *IEEE Trans. on Circuits and Systems for Video Technology* 7(2), 397–401 (1997)

Exploring Combinations of Ontological Features and Keywords for Text Retrieval

Tru H. Cao¹, Khanh C. Le², and Vuong M. Ngo¹

¹ Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology, Vietnam
{tru@cse.hcmut.edu.vn, vuong@cse.hcmut.edu.vn}
² TMA Solutions, Vietnam
{lckhanh@tma.com.vn}

Abstract. Named entities have been considered and combined with keywords to enhance information retrieval performance. However, there is not yet a formal and complete model that takes into account entity names, classes, and identifiers together. Our work explores various adaptations of the traditional Vector Space Model that combine different ontological features with keywords, and in different ways. It shows better performance of the proposed models as compared to the keyword-based Lucene, and their advantages for both text retrieval and representation of documents and queries.

1 Introduction

Information retrieval, in general, and text retrieval¹, in particular, is not a new area but still attracts much research effort, social and industrial interests. That is because, on the one hand, it is important for searching required information, especially on the explosive WWW, and on the other hand, there are still many open problems to be solved to enhance the existing methods or to propose new models. Retrieval precision and recall could be improved by developing appropriate models, typically as similarity-based ([5], [13]), probabilistic relevance ([15]), or probabilistic inference ([16]) ones. Semantic annotation, representation, and processing of documents and queries are another way to obtain better performance ([4], [6], [8], [17]).

Traditionally, text retrieval is only based on keywords (KW) occurring in documents and queries. Later on, word similarity and relationship are exploited to represent and match better documents to a query. However, keywords alone are not adequate, because in many domains and cases named entities (NE) constitute the user intention in a query and the main content of a document. Named entities are those that can be referred to by names, such as people, organizations, and locations ([14]). They are inherently different from words, as they represent individuals while words denote general concepts, such as types, properties, and relations. If named entities are marked up in texts then, for example, one can search for, and correctly obtain, web pages about *Saigon* as a city. Whereas current search engines like Google may return any page that contains the word *Saigon*, though it is the name of a river or a university.

¹ In this paper we use the terms *information retrieval*, *text retrieval*, and *document retrieval* interchangeably, though they are not quite the same.

There are different ontological features of named entities that can be of user interest and expressed in a query. First, the user may want to search for documents about exactly identified named entities, like the *Saigon City* in Viet Nam but not a city of the same name elsewhere. Second is the case when only the name and class of entities are of concern or available, as in searching for documents about people named *McCarthy*. Third, one may be interested in documents about entities of a certain class, like city capitals. Fourth, it is not uncommon that only entity names are the criterion of a search. In short, the possible distinct features of named entities in question are names, classes, joins of names and classes, and identifiers. Nevertheless, usually, a query cannot be completely specified without keywords, like “*economic growth of East Asian countries*”, where *East Asian countries* represents named entities while *economic* and *growth* are keywords.

Until now, to our knowledge, there is no information retrieval model that formally integrates and treats all above-mentioned named entity features in combination with keywords. Our work presented in this paper is to explore and analyse possible combinations of ontological features and keywords in the formal framework of the Vector Space Model (VSM) and its adaptation. Implementation and experiments are also carried out to evaluate and compare the performance of developed models themselves and to the traditional purely keyword-based VSM. Section 2 recalls the basic notion of the traditional VSM and system, and its adaptation for the named entity spaces. Section 3 presents alternative adapted VSMs that combine both named entities and keywords. Section 4 is for evaluation and discussion on experimental results. In Section 5, we review related works in comparison with our approach. Finally, Section 6 gives some concluding remarks.

2 Ontology-Based Multi-vector Space Models

Despite having known disadvantages, VSM is still a popular model and a basis to develop other models for information retrieval, because it is simple, fast, and its ranking method is in general either better or almost as good as a large variety of alternatives ([1]). We recall that, in the keyword-based VSM, each document is represented by a vector over the space of keywords of discourse. Conventionally, the weight corresponding to a term dimension of the vector is a function of the occurrence frequency of that term in the document, called *tf*, and the inverse occurrence frequency of the term across all the existing documents, called *idf*. The similarity degree between a document and a query is then defined as the cosine of their representing vectors.

Given a query, the retrieval process composes of two main stages, namely, document filtering and document ranking. The former selects those documents that satisfy the Boolean expression of keywords as specified in the query. For example, if the query is $k_1 \vee k_2$, and D_1 and D_2 are respectively the sets of documents that contain k_1 and k_2 , then $D_1 \cup D_2$ is the set of selected documents. In the latter, those selected documents are ranked by their similarity degrees to the query as calculated above.

With terms being keywords, the traditional VSM cannot satisfactorily represent the semantics of texts with respect to the named entities they contain, such as for the following queries:

Q_1 : Search for documents about *cities*.

Q_2 : Search for documents about *Saigon City*.

Q_3 : Search for documents about *Hanoi Tower*.

Q_4 : Search for documents about *Hanoi University of Technology*.

That is because, for Q_1 , a target document does not necessarily contain the keyword *city*, but only some named entities of the class *City*, i.e., real cities in the world. For Q_2 , a target document may mention about *Saigon City* by other names, i.e., the city's aliases, such as *Ho Chi Minh City*. On the other hand, documents containing entities named *Saigon* but not being cities, like *Saigon River*, are not target documents. For Q_3 , documents about *Hanoi* as a city or a university are not target documents at all, though containing the keyword *Hanoi*. Meanwhile, Q_4 targets at documents about a precisely identified named entity, i.e., *Hanoi University of Technology*, not other universities of similar names. Therefore, simple keyword looking up and matching may fail to give expected answers.

For formally representing documents (and queries) by named entity features, we define the triple (N, C, I) where N , C , and I are respectively the sets of names, classes, and identifiers of named entities in the ontology of discourse. Then:

1. Each document d is modelled as a subset of $(N \cup \{*\}) \times (C \cup \{*\}) \times (I \cup \{*\})$, where '*' denotes an unspecified name, class, or identifier of a named entity in d , and
2. d is represented by the quadruple $(\vec{d}_N, \vec{d}_C, \vec{d}_{NC}, \vec{d}_I)$, where \vec{d}_N , \vec{d}_C , \vec{d}_{NC} , and \vec{d}_I are respectively vectors over N , C , $N \times C$, and I .

A feature of a named entity could be unspecified due to the user intention expressed in a query, the incomplete information about that named entity in a document, or the inability of an employed NE recognition engine to fully recognize it. Each of the four component vectors introduced above for a document can be defined as a vector in the traditional *tf.idf* model on the corresponding space of entity names, classes, name-class pairs, or identifiers, instead of keywords. However, there are two following important differences with those ontological features of named entities in calculating their frequencies:

1. The frequency of a name also counts identical entity aliases. That is, if a document contains an entity having an alias identical to that name, then it is assumed as if the name occurred in the document. For example, if a document refers to *Saigon City*, then each occurrence of that entity in the document is counted as one occurrence of the name *Ho Chi Minh City*, because it is an alias of *Saigon City*.
2. The frequency of a class also counts occurrences of its subclasses. That is, if a document contains an entity whose class is a subclass of that class, then it is assumed as if the class occurred in the document. For example, if a document refers to *Saigon City*, then each occurrence of that entity in the document is counted as one occurrence of the class *Location*, because *City* is a subclass of *Location*.

The similarity degree of a document d and a query q is then defined to be, where $w_N + w_C + w_{NC} + w_I = 1$:

$$\text{sim}(\vec{d}, \vec{q}) = w_N.\text{cosine}(\vec{d}_N, \vec{q}_N) + w_C.\text{cosine}(\vec{d}_C, \vec{q}_C) + w_{NC}.\text{cosine}(\vec{d}_{NC}, \vec{q}_{NC}) + w_I.\text{cosine}(\vec{d}_I, \vec{q}_I) \quad (\text{Eq. 1})$$

We deliberately leave the weights in the sum unspecified, to be flexibly adjusted in applications, depending on user-defined relative significances of the four ontological features. We note that the join of \vec{d}_N and \vec{d}_C cannot replace \vec{d}_{NC} because the latter is concerned with entities of certain name-class pairs. Meanwhile, \vec{d}_{NC} cannot replace \vec{d}_I because there may be different entities of the same name and class (e.g. there are different cities named *Moscow* in the world). Also, since names and classes of an entity are derivable from its identifier, products of I with N or C are not included. In brief, here we generalize the notion of terms being keywords in the traditional VSM to be entity names, classes, name-class pairs, or identifiers, and use four vectors on those spaces to represent a document or a query for text retrieval.

There are still possible variations of this proposed ontology-based multi-vector space model that are worth exploring. Firstly, that is due to overlapping of those four types of generalized terms in a query, which all convey information about the documents that a user wants to search for. For example, given a query containing *Ho Chi Minh City*, this entity includes all the four terms, namely the identifier of the entity itself, the name-class pair (*Ho Chi Minh*, *City*), the class *City*, and the name *Ho Chi Minh*. We call these variations overlapped or non-overlapped models, respectively denoted by NEo or NEn, depending on whether term overlapping is taken into account or not. Figure 1 shows a query in the TIME test collection (available with [2]) and its corresponding sets of ontological terms that we extract for the two models, where *InternationalOrganization_T.17* is the identifier of *United Nations* in the knowledge base of discourse.

Query: “*Countries have newly joined the United Nations*”.

Overlapped ontological term set:

$\{(*/\text{Country}/*), (\text{United Nations}/*/*), (*/\text{InternationalOrganization}/*), (\text{United Nations}/\text{InternationalOrganization}/*), (\text{United Nations}/\text{InternationalOrganization}/\text{InternationalOrganization_T.17})\}$

Non-overlapped ontological term set:

$\{(*/\text{Country}/*), (\text{United Nations}/\text{InternationalOrganization}/\text{InternationalOrganization_T.17})\}$

Fig. 1. Overlapped and non-overlapped ontological terms extracted from a query

As in the traditional VSM retrieval process, after the Boolean document filtering stage, let D_N , D_C , D_{NC} , and D_I be the respective sets of obtained documents containing generalized terms of the four ontological features in a query. For the document ranking stage, we take the intersection of D_N , D_C , D_{NC} , and D_I in the overlapped model or their union in the non-overlapped model, respectively, as the set of documents to be ranked and returned for the query. This application of intersection or union operations can be justified as responding to the overlapping effect, which is supported by experimental results shown later.

3 Combining Named Entities and Keywords

Clearly, named entities alone are not adequate to represent a text. For example, in the query in Figure 1, *joined* is a keyword to be taken into account, and so are *Countries* and *United Nations*, which can be concurrently treated as both keywords and named entities. Therefore, a document can be represented by one vector on keywords and four vectors on ontological terms. Then, given a query, after the document filtering stage, one can take either the intersection or the union of the document set satisfying the Boolean expression of the keywords and the document set satisfying the Boolean expression of the named entities in the query.

Regarding also overlapping or non-overlapping of ontological terms as discussed in Section 2, one have four alternative models combining keywords and named entities, denoted by $KW \cap NEo$, $KW \cap NEn$, $KW \cup NEo$, and $KW \cup NEn$. The similarity degree of a document d and a query q is then defined as follows, where $w_N + w_C + w_{NC} + w_I = 1$, $\alpha \in [0, 1]$, and \vec{d}_{KW} and \vec{q}_{KW} are respectively the vectors representing the keyword features of d and q :

$$\text{sim}(\vec{d}, \vec{q}) = \alpha.[w_N.\text{cosine}(\vec{d}_N, \vec{q}_N) + w_C.\text{cosine}(\vec{d}_C, \vec{q}_C) + w_{NC}.\text{cosine}(\vec{d}_{NC}, \vec{q}_{NC}) + w_I.\text{cosine}(\vec{d}_I, \vec{q}_I)] + (1 - \alpha).\text{cosine}(\vec{d}_{KW}, \vec{q}_{KW}) \quad (\text{Eq. 2})$$

We now explore another adapted VSM that combines keywords and named entities. That is we unify and treat all of them as generalized terms, where a term is counted either as a keyword or a named entity but not both. Each document is then represented by a single vector over that generalized term space. Document vector representation, filtering, and ranking are performed as in the traditional VSM, except for taking into account entity aliases and class subsumption as presented in Section 2. We denote this model by KW+NE. Figure 2 show another query in the TIME test collection and its corresponding key term sets for the multi-vector space models and the KW+NE model.

Query: “U.N. team survey of public opinion in North Borneo and Sarawak on the question of joining the federation of Malaysia”.

Multi-vector space models ($KW \cap NEo$, $KW \cap NEn$, $KW \cup NEo$, $KW \cup NEn$):

Keywords = {*U.N.*, *opinion*, *North Borneo*, *Sarawak*, *join*, *federation*, *Malaysia*}

Onto-terms = {(*U.N./InternationalOrganization/InternationalOrganization_T.17*), (*North Borneo/Province_Province_T.2189*), (*Sarawak/Location/**), (*Malaysia/Country/Country_T.MY*)}

KW+NE model:

Generalized terms = {(*U.N./InternationalOrganization/InternationalOrganization_T.17*), *opinion*, (*North Borneo/Province_Province_T.2189*), (*Sarawak/Location/**), *join*, *federation*, (*Malaysia/Country/Country_T.MY*)}

Fig. 2. Keywords, ontological terms, and generalized terms extracted from a query

4 Implementation and Experimentation

We have implemented the above-adapted VSMs by employing and modifying Lucene, a general VSM-based open source for storing, indexing and searching documents ([7]).

We have evaluated and compared the new models in terms of precision-recall (P-R) curves and single F-measure values. For each query in a test collection, we adopt the common method in [11] to obtain the corresponding P-R curve. That is, the returned documents are examined from the top to the bottom, regarding their similarity degrees to the query. At each step, the precision and recall for the documents that have been examined are calculated, creating one point of the curve.

In order to obtain the average P-R curve over all the queries in the test collection, each query curve is interpolated to the eleven standard recall levels that are 0%, 10%, ..., 100%, as in [1]. The interpolated precision for the i -th query at the j -th standard recall level r_j ($j \in \{0, 1, \dots, 10\}$) is defined by $P_i(r_j) = \max_{r_j \leq r \leq r_{j+1}} P_i(r)$. Given N_q as the number of queries, the average precision at r_j over all the queries is then computed by $\bar{P}(r_j) = \sum_{i=1}^{N_q} \frac{P_i(r_j)}{N_q}$. Consequently, the interpolated F-measure value for the i -th query at r_j is $F_i(r_j) = \frac{2.P(r_j).r_j}{P(r_j) + r_j}$, and the average F-measure value at r_j over all the queries is $\bar{F}(r_j) = \sum_{i=1}^{N_q} \frac{F_i(r_j)}{N_q}$.

We have conducted experimentation on the TIME collection, containing 425 documents and 83 queries. The ontology and NE recognition engine of KIM ([10]) are employed to automatically annotate named entities in documents. For the queries, we manually extract and mark their named entities and keywords, to represent their meanings concisely and appropriately for document retrieval. In the experiments, we set the weights $w_N = w_C = w_{NC} = w_I = 0.25$ and $\alpha = 0.5$, assuming that the keyword and named entity dimensions are of equal importance.

Table 1 presents the average precisions of the keyword-based VSM by Lucene itself, the NE-based overlapped/non-overlapped models, and the KW-NE-based² models combining named entities and keywords, at each of the standard recall levels. Table 2 shows their average F-measure values. One can observe that, for all the models, the maximum F-measure values are achieved at the 50% recall level. The performances of the NEo and NEn models are quite similar (39.1 and 38.9), so are those of the KW-NE-based models (around 42.0). Therefore, we take the NEn model and the KW+NE model as representatives of these two groups, respectively. The similar performances of the models in each group justify our use of intersection or union on filtered document sets in accordance to overlapping or non-overlapping application on query terms.

Overall, KW+NE is better than NEn (42.46 versus 38.9), and both are better than the Lucene baseline (37.93). The difference would be larger on a test collection involving more named entities and ontological terms than in the TIME one. Figure 3 illustrates the average P-R and average F-R curves of the three models. We have also examined some typical queries for which KW+NE is better than, as good as, or worse than Lucene, as shown in Figure 4. Following are those queries and our analysis.

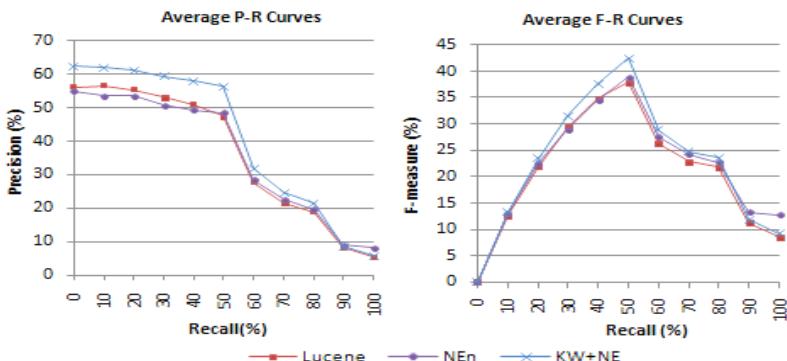
² We use *KW-NE-based* to refer to all proposed models combining keywords and named entities (i.e., $KW \cap NEo$, $KW \cap NEn$, $KW \cup NEo$, $KW \cup NEn$, and $KW+NE$).

Table 1. The average precisions at the eleven standard recall levels

	Recall (%)											Precision (%)
	0	1	2	3	4	5	6	7	8	9	1	
Lucene	5	5	5	5	5	4	2	2	1	8	5	
NEo	5	5	5	4	4	4	2	2	2	1	1	
NEn	5	5	5	5	4	4	2	2	1	8	8	
KW+NE	6	6	6	5	5	5	3	2	2	8	5	
KW \cup NEo	6	6	5	5	5	5	3	2	2	7	4	
KW \cap NEo	6	6	5	5	5	5	3	2	2	1	1	
KW \cup NEn	6	6	6	5	5	5	3	2	1	7	4	
KW \cap NEn	6	6	6	5	5	5	3	2	2	1	9	

Table 2. The average F-measure values at the eleven standard recall levels

	Recall (%)											F-measure (%)
	0	1	2	3	4	5	6	7	8	9	1	
Lucene	0	1	2	2	3	3	2	2	2	1	8	
NEo	0	1	2	2	3	3	2	2	2	1	1	
NEn	0	1	2	2	3	3	2	2	2	1	1	
KW+NE	0	1	2	3	3	4	2	2	2	1	9	
KW \cup NEo	0	1	2	3	3	4	2	2	2	1	7	
KW \cap NEo	0	1	2	3	3	4	3	2	2	1	1	
KW \cup NEn	0	1	2	3	3	4	2	2	2	9	7	
KW \cap NEn	0	1	2	3	3	4	3	2	2	1	1	

**Fig. 3.** Average P-R and F-R curves of Lucene, NEn, and KW+NE models

Query a. “*Kennedy administration pressure on Ngo Dinh Diem to stop suppressing the buddhists*”. For this query, our single vector model KW+NE performs better than Lucene because the latter fails to recognize aliases of the named entities in the query. There are two ontological terms here, namely, (*Kennedy/Person/**) and (*Ngo Dinh Diem/Person/**). In the document collection, *Kennedy* also occurs as *John Kennedy*, and *Ngo Dinh Diem* has two other aliases *NgoDinh Diem* and *Diem*. These aliases are used frequently in the documents, but keyword-based search sees them as different terms, which leads to a reduction in retrieval precision.

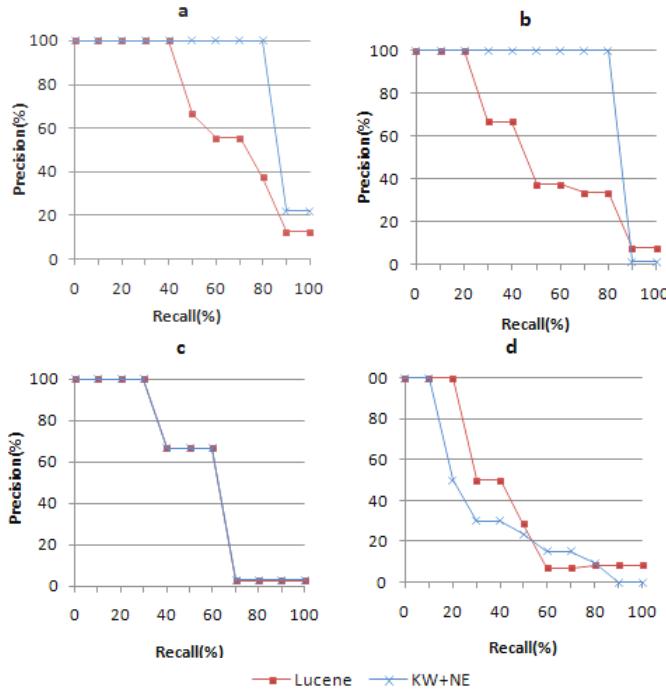


Fig. 4. Performances on typical queries of Lucene and KW+NE models

Query b. “*Persons involved in the Viet Nam war*”. For this query, KW+NE also outperforms Lucene. That can be explained by the fact that, while keyword-based search looks for documents explicitly containing the words *person* or *persons*, KW+NE recognizes and selects also those documents that contain named entities of the class *Person*. It boosts up the ranking values of relevant documents to be placed at the top of the returned document list.

Query c. “*Somalia is involved in border disputes with its neighbors what military aid is being supplied to Somalia by Russia*”. This is a case when KW+NE and Lucene have no performance difference. That is because there are no aliases of *Somalia* and *Russia* in the document collection. So, what actually happens is that KW+NE matches identifiers with identifiers whereas Lucene matches names with names, representing the two named entities. Without aliases, that obviously does not affect the results.

Query d. “*Indian fears of another Chinese invasion*”. For this query, Lucene performs slightly better than KW+NE. Here, two implicit named entities *India* and *China* are manually extracted from *Indian* and *Chinese*, respectively. However, KIM NE recognition engine could not detect named entities implicitly occurring in a document under the adjective form. So, with the KW+NE model, a document just containing the keywords *Indian* and *Chinese* is not considered as relevant to the query, while with Lucene they are. That explains the difference.

We note that the performance of any system relying on named entities to solve a particular problem partly depends on that of the NE recognition module in a preceding stage. However, in research for models or methods, the two problems should be separated. This paper is not about NE recognition and our experiments incur errors of the employed KIM engine, whose current average precision and recall are respectively 90% and 86%.

Among the KW+NE-based models, the KW+NE model is straightforward and simple, unifying keywords and named entities as generalized terms, while having comparable performance as the others. Meanwhile, the multi-vector models can be useful for clustering documents into a hierarchy via top-down phases each of which uses one of the four NE-based vectors presented above (cf. [3]). For example, given a set of geographical documents, one can first cluster them into groups of documents about rivers and mountains, i.e., clustering with respect to entity classes. Then, the documents in the river group can be clustered further into subgroups each of which is about a particular river, i.e., clustering with respect to entity identifiers. As another example of combination of clustering objectives, one can first make a group of documents about entities named *Saigon*, by clustering them with respect to entity names. Then, the documents within this group can be clustered further into subgroups for *Saigon City*, *Saigon River*, and *Saigon Market*, for instance, by clustering them with respect to entity classes. Another advantage of splitting document representation into four component vectors is that, searching and matching need to be performed only for those components that are relevant to a certain query.

5 Related Works

In [12], a probabilistic relevance model was introduced for searching passages about certain biomedical entity types (i.e., classes) only, such as genes, diseases, or drugs. Also in the biomedical domain, the similarity-based model in [18] considered concepts being genes and medical subject headings, such as *purification*, *HNF4*, or *hepatitis B virus*. Concept synonyms, hypernyms, and hyponyms were taken into account, which respectively corresponded to entity aliases, super-classes, and subclasses in our NE-based models. A document or query was represented by two component vectors, one of which was for concepts and the other for words. A document was defined as being more similar to a query than another document if the concept component of the former is closer to that of the query. If the two concept components were equally similar to that of the query, then the similarity between the word components of the two documents and that of the query would decide. However, as such, the word component was treated as only secondary in the model, and its domain was just limited within biomedicine. Recently, [9] researched and showed that NE normalization improved retrieval performance. The work however considered only entity names and that normalization issue was in fact what we call aliasing here.

Two closely related works to ours are [4] and [6]. In [4], the authors adapted the traditional VSM with vectors over the space of NE identifiers in the knowledge base of discourse. For each document or query, the authors also applied a linear combination of its NE-identifier-based vector and keyword-based vector with the equal weights of 0.5. The system was tested on the authors' own dataset. The main drawback was that

every query had to be posed using RDQL, a query language for RDF, to first look up in the system's knowledge base those named entities that satisfied the query, before its vector could be constructed. For example, given the query searching for documents about *Basketball Player*, its vector would be defined by the basketball players identified in the knowledge base. This step of retrieving NE identifiers was unnecessarily time consuming. Moreover, any knowledge base is usually incomplete, so documents containing certain basketball players not existing in the knowledge base would not be returned. In our proposed models, the query and document vectors on the entity class *Basketball Player* can be constructed and matched right away.

Meanwhile, the LRD (Latent Relation Discovery) model proposed in [6] used both keywords and named entities as terms for a single vector space. The essential of the model was that it enhanced the content description of a document by those terms that did not exist, but were related to existing terms, in the document. The relation strength between terms was based on their co-occurrence. The authors tested the model on 20 randomly chosen queries from 112 queries of the CISI dataset ([2]), achieving the maximum F-measure of 19.3. That low value might be due to the dataset containing few named entities. Anyway, the model's drawback as compared to our KW+NE model is that it used only entity names but not all ontological features. Consequently, it cannot support queries searching for documents about entities of particular classes, name-class pairs, or identifiers.

6 Conclusion

We have presented various adapted VSMs that take into account possible combinations of ontological features with keywords, which all yield nearly the same performance and are better than the keyword-based Lucene. Our consideration of entity name aliases and class subsumption is logically sound and empirically verified. We have shown that overlapping of ontological features if applied to a query can be compensated by taking intersection of the selected document sets with respect to each of the features. Also, retrieval performance is not sensitive to the choice of intersection or union of the selected documents satisfying the keyword expression and that for the named entity expression in the query.

For its uniformity and simplicity, we propose the single vector KW+NE model for text retrieval. Meanwhile, the multi-vector model is useful for document clustering with respect to various ontological features. These are the first basic models that formally accommodate all entity names, classes, joint names and classes, and identifiers. Within the scope of this paper, we have not considered similarity and relatedness of generalized terms of keywords and named entities. This is currently under our investigation expected to increase the overall performance of the proposed models.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Reading (1999)
2. Buckley, C.: Implementation of the SMART Information Retrieval System. Technical Report 85-686, Cornell University (1985)

3. Cao, T.H., Do, H.T., Hong, D.T., Quan, T.T.: Fuzzy Named Entity-Based Document Clustering. In: Proceedings of the 17th IEEE International Conference on Fuzzy Systems, pp. 2028–2034 (2008)
4. Castells, P., Vallet, D., Fernández, M.: An Adaptation of the Vector Space Model for Ontology-Based Information Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 261–272 (2006)
5. Dominich, S.: Paradox-Free Formal Foundation of Vector Space Model. In: Proceedings of the ACM SIGIR 2002 Workshop on Mathematical/Formal Methods in Information Retrieval, pp. 43–48 (2002)
6. Gonçalves, A., Zhu, J., Song, D., Uren, V., Pacheco, R.: LRD: Latent Relation Discovery for Vector Space Expansion and Information Retrieval. In: Proceedings of the 7th International Conference on Web-Age Information Management (2006)
7. Gospodnetic, O.: Parsing, Indexing, and Searching XML with Digester and Lucene. *Journal of IBM DeveloperWorks* (2003)
8. Guha, R., McCool, R., Miller, E.: Semantic Search. In: Proceedings of the 12th International Conference on World Wide Web, pp. 700–709 (2003)
9. Khalid, M.A., Jijkoun, V., de Rijke, M.: The Impact of Named Entity Normalization on Information Retrieval for Question Answering. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 705–710. Springer, Heidelberg (2008)
10. Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic Annotation, Indexing, and Retrieval. *Journal of Web Semantics* 2 (2005)
11. Lee, D.L., Chuang, H., Seamons, K.: Document Ranking and the Vector-Space Model. *IEEE Software* 14, 67–75 (1997)
12. Meij, E., Katrenko, S.: Bootstrapping Language Associated with Biomedical Entities. In: Proceedings of the 16th Text REtrieval Conference (2007)
13. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. *Communications of the ACM* 18, 613–620 (1975)
14. Sekine, S.: Named Entity: History and Future. *Proteus Project Report* (2004)
15. Sparck Jones, K., Walker, S., Robertson, S.E.: A Probabilistic Model of Information Retrieval: Development and Comparative Experiments – Part 1 and Part 2. *Information Processing and Management* 36, 779–808, 809–840 (2000)
16. van Rijbergen, C.J.: A Non-Classical Logic for Information Retrieval. *The Computer Journal* 29, 481–485 (1986)
17. Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E.G.M., Miliotis, E.E.: Semantic Similarity Methods in WordNet and Their Application to Information Retrieval on the Web. In: Proceedings of the 7th Annual ACM Intl Workshop on Web Information and Data Management, pp. 10–16 (2005)
18. Zhou, W., Yu, C.T., Torvik, V.I., Smalheiser, N.R.: A Concept-based Framework for Passage Retrieval in Genomics. In: Proceedings of the 15th Text REtrieval Conference (2006)

Instance Management Problems in the Role Model of Hozo

Kouji Kozaki, Satoshi Endo, and Riichiro Mizoguchi

The Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan
`{kozaki, endo, miz}@ei.sanken.osaka-u.ac.jp`

Abstract. For knowledge (instances model) representation based on ontology and its use, it is desirable to understand phenomena in the target world as precisely as possible. The theory of ontology should reflect the understanding of them and provide a fundamental framework to manage the behavior of instances adequately. Hozo is known as an ontology-development tool with an ability to deal with roles and their instances. Although Hozo allows users to represent roles better than other existing tools, the underlying theoretical foundations are still unclear and there is some room for improvement concerning the generality of how to deal with instances of roles. Especially, establishment of the instance management method of role concept based on the ontological theory is an important subject. It is mainly concerned with handling of appearance and disappearance of role instances. This article discusses a refinement of role theory through investigation of problems such as distinction between constituent roles and general roles and what kind of constituent roles are needed in what situations are also discussed in detail.

Keywords: Ontology, instance management, role theory, ontology modeling.

1 Introduction

Ontology has been used as the basis of knowledge systems in various domains, and its utility is recognized more widely day by day. An ontology provides “an explicit specification of conceptualization” [1] underlying any knowledge representation (an instance model), and it is one of the important roles to keep the consistency and reusability of knowledge by describing them based on the ontology. Many researchers study ontological theories intended to contribute to building a well-founded ontology. Especially, theory of roles is one of the critical topics. The world is full of roles. Roles have various characteristics such as anti-rigidity [2], dynamics [3], context dependency, and so on. For example, by dynamics, we mean that the role which a person plays can be change dynamically according to the focal context. (e.g., a man may be a teacher in a school and a husband in a married couple.) We have been investigating these characteristics of roles and how to deal with them on computer systems as accurately as possible. As a result, we have developed an ontology development/use tool, named Hozo, based on fundamental consideration of roles [4]. However, although

some researchers discuss ontological theories of roles, there remains some room for investigations of instance management problems such as the counting problem [5], appearance/disappearance of instances of roles, dynamic change of roles which players play, and so on. It is important to establish an ontological theory for instance management of roles so that we can capture their behavior and manage them in a sound manner. In addition, there is no tool which implements role theory besides Hozo in the world.

This paper discusses these instance management problems related to roles in a generic framework, and we refine our role model to build and manage instances of roles appropriately. Furthermore, we think that they are general problems in role theories and are not limited to the role model in Hozo. The next section summarizes three instance management issues of roles which are discussed in the following sections. Section 3 discusses our previous role model and dependency of an instance of a role on its context. Section 4 discusses the main issue we investigate in this paper, and we propose distinction between *constituent role* and *post role* to solve the instance management concerned with promotion. Section 5 gives some discussions about identities of roles under consideration. Related work is discussed in Section 6, followed by concluding remarks.

2 Instance Management Problem of Roles

A role is defined as “a name of entity which changes according to contexts” or “an entity that is played by another entity in a context” [4]. For example, when an instance of a man is playing a teacher role in a school, it means that he is a teacher. And the man would be regarded as a teacher in the school and as a husband in his marital relationship. In this example, *teacher* and *husband* are roles, and they differ from so-called natural types (e.g. human).

These characteristics of roles can explain various behaviors of an instance in the real world. The following shows typical ones:

(1) Dependency of an instance of a role on its context

We assume *an instance of human*, Taro, who is a *teacher* in Osaka high school. If the school is closed down, he stops being a teacher in the school while he is still an instance of human. This example says the existence of an instance of a *role* (e.g. teacher) is dependent on that of its *context* (e.g. school).

(2) Continuity of instances of roles

We assume Jiro is an associate professor of Osaka University. When he promotes from associate professor to full professor, his post changes while he remains to be a teaching staff in the university. It implies that some instances of roles continue to exist while others change when the player changes roles to play.

(3) Identity of instances of roles

When the Prime Minister of Japan changes from Abe to Fukuda, we can regard that they play the same role as the head of the Japanese Government. However, we also can recognize they play different roles (e.g. 90th and 91st Prime Ministers). The details of these issues and its solution are discussed in following sections.

3 Role Model

3.1 Fundamental Schema of Our Role Model

The fundamental scheme of our roles at the instance level is the following (see the lower diagram in Fig. 1.):

“In Osaka high school, Taro plays teacher role-1 and thereby becomes teacher-1”

This can be generalized to the class level (see the upper diagram in Fig. 1.)

“In schools, there are persons who play teacher roles and thereby become teachers.”

By **play**, we mean that something “acts as”, that is, it contingently acts as according to the role (role concept). By “**teacher**”, we mean a class of dependent entities which roughly correspond to persons who are playing teacher roles and which are often called *qua individuals* [5]. Here, we introduce a couple of important concepts to enable finer distinctions among role-related concepts: **role concept**, **role holder**, **potential player** and **role-playing thing**. In the above example, these terms are used as “*In a context, there are potential players who can play role concepts and thereby become role holders*”.

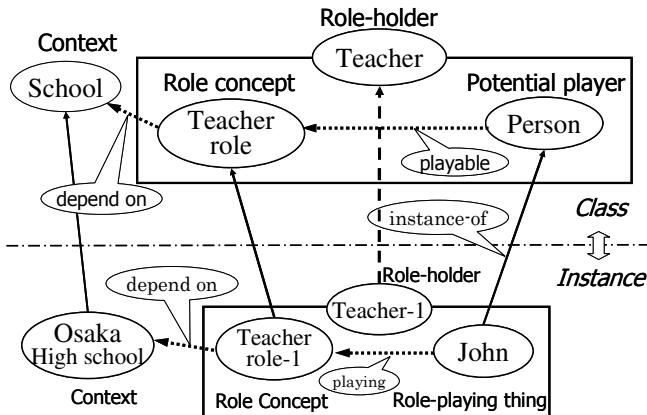


Fig. 1. Fundamental scheme of a role concept and a role holder

By **context**, we mean a class of things that should be considered as a whole. A context includes entities and relations. **Role concept** is defined as a concept whose entities are played by some entity within a context. So, it essentially depends on the context. By **potential player**, we mean a class of things which are able to play an instance of a role concept. In many cases, basic concepts (natural types) can be used to denote classes of **potential players**. When an instance of potential player is playing the instance of role concept, we call the instance a **role-playing thing**. In this example, we say a person can play an instance of a teacher role. In particular, Taro is actually playing a specific teacher role *teacher role-1*. By doing so, he/she is associated with the instance *teacher-1*, an individual teacher **role holder**. A role-holder class is a class of dependent entities like teacher-1. As such, it is neither a specialization of a potential player class (e.g., person) nor that of a role concept class (e.g., teacher role),

but an abstraction of a composition of a role-playing thing and an instance of role concept, as is shown in Figure 1, which is the heart of our Role model. The link from Teacher-1 to Teacher is a broken arrow rather than a solid one like instance-of link to show the relation is not completely same as instance-of relation in Fig. 1. Our model and tool do not allow people to directly instantiate role holder classes because the individual role holder as a dependent entity to be instantiated inherently requires first an instance of a potential player class and of a role concept class. Then, when the playing link is asserted, it virtually acquires the properties of the potential player and the role concept. This is why role holders are dependent entities.

All the concepts introduced here are core of our role model and contain rich implications which are elaborated in [4]. The above shows that we divide the conventional notion of “Role” into two kinds: role concept and role holder in our model. Therefore, our model of roles does not have the concept of “Role” explicitly. In particular, it is understood conventionally that a role existing at the instance level must be something being played by something, since people understand the role instantiation and the action of playing the role as happening at the same time. In contrast, in our model a role concept can exist at the instance level without being played, since it depends only on its context and not on its player. While the concept of role is the target of the ontological research on roles, at the same time, this term has been the source of confusion, since it hides the difference between role concept and role holder.

3.2 Hozo’s Representation Our Role Model

Figure 2 shows the correspondence between the model and the corresponding Hozo representation. Because Hozo is based on frames, the representation is rather straightforward. Additionally, we discussed theoretical solid foundation and formal definitions of our role theory in previous work [4]. In the paper, we discussed the solid foundation of role model and presented its semantics using OWL to clarify its formal definitions. The details of role representation model using OWL and SWRL are discussed in [6]. Hozo also can export ontologies in OWL.

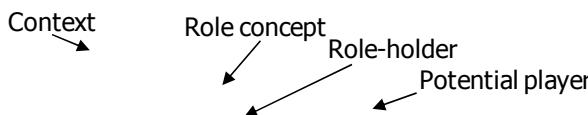


Fig. 2. Hozo’s representation our role model

Let us explain Hozo’s representation conventions by using the example shown in Fig. 2. In Hozo each concept defined as a class is represented in a rectangle like *School* and *Person*. Each class is defined by specifying its parts (denoted by “p/o”) and/or attributes (denoted by “a/o”) as slots. *School* is here defined as an entity composed of teachers and students where *teacher role* and *student role* are role concepts

played by individuals specified by the rectangle at the far right, instances of *Person* in this case. As shown in Figure 2, the key idea of class definition in Hozo is that all concepts, which can theoretically be parts of something, are defined independently of the possible wholes they belong to, and each class as a whole is defined by specifying the roles whose parts play. In other words, all the class definitions in Hozo are reciprocal, in the sense that a whole (*School*) is defined in terms of its parts (*Person*) playing their own roles, and at the same time, the roles (*teacher role*) played by the parts (*Person*) are defined there under the context of the whole (*School*).

Fig. 3. Representation of the hierarchy of role concepts

Is-a (super-sub) relations between basic concepts are represented by *is-a links* as shown Fig. 3. In this example, *University* is defined as a sub concept of *School*. Sub concepts inherit all role concepts from their super concepts, and sometimes they specialize inherited role concepts to define the role concepts in their context. Fig. 3 shows two role concepts (*full professor role* and *associate professor role*) which are defined in the context of *University*. They are sub concepts of *teacher role* in *School*. The relationships between these role concepts are represented by describing the super concepts on the right of role concepts with double angles “<<” as shown in Figure 3. It means the hierarchy of role concepts is analogue to the hierarchy of basic concepts because all role concepts are defined within the basic concepts as their contexts.

3.3 Instances of Role Concepts

The example of the teacher discussed in section 2-(1) can be elaborated and generalized in the following manner. Firstly, if *Osaka High School* does not exist, the instance of the teacher role never exists. In general, any instance of a role concept cannot exist without an instance of its context. This dependency applies to all types of role concepts. Secondly, a vacancy in a teacher post arises when the instance of the *teacher role* is not played. Such a vacancy supports the existence of the role concept. Furthermore, it means that the role concept has two states: played and not played. It can exist in the un-played state because some values of some properties including those of the essential properties of the role concept (for example, in the case of the *teacher role*, *subject*, *class*, and so on) can be determined independently of whether it is played or not. But *name* or *age* of the *teacher* cannot be determined until someone plays it. Thirdly, Taro is no longer a teacher when the teacher position he fills disappears, when he quits the *teacher role*, or when he dies. In general, an individual role holder disappears in the following cases: an instance of the role concept disappears, an instance of the player stops playing the role or an instance of the player disappears. This is understood because that an individual role holder is dependent on the

individuals of a role concept and of its player as far as the playing relation is valid as discussed in Section 3.1. This observation suggests that the identity (ID) of the individual of the role holder is a function of the IDs of the role concept (ID_{Role}) and of the player (ID_{Player}). That is, $ID_{Role\ holder} = f(ID_{Role}, ID_{Player})$ in which both arguments are mandatory for $ID_{Role\ holder}$, and in which "f" is bijective (surjective and injective).

Here, we generalize the characteristics of instances of role-related concepts. An instance model specifies the interdependencies between classes and individuals, especially concerning the appearance and extinction of individuals. It appears as indispensable for the concrete application of ontologies, and for a clarification of the nature of role instances. In the following, \mathbf{R} denotes a role concept, \mathbf{C}^1 the contexts it depends on, and \mathbf{P} is a concept considered as the potential player of \mathbf{R} .

(A) Dependence of instances of role concepts on their context

An instance of \mathbf{R} exists if (and only if) an instance of \mathbf{C} is instantiated. When the instance of \mathbf{R} ceases to exist, so does the instance of \mathbf{R} .

(B) Dependence of instances of role concepts on their players

An instance of \mathbf{R} is dealt with as a defective instance by itself. When the instance of \mathbf{R} as constituents of \mathbf{R} is played by an instance of \mathbf{P} , \mathbf{R} is concretized to be a complete instance corresponding to \mathbf{R} .

(C) Extinction of a role holder

A role holder of \mathbf{R} is composed of both instances of \mathbf{R} and \mathbf{P} by combining all of their slots. Let r and p denote instances of \mathbf{R} and \mathbf{P} , respectively. Then, there are three cases in which the individual role holder disappears: (1) p disappears, (2) r disappears and (3) p stops playing r .

4 Instance Management Problem Concerned with Promoting

4.1 Instance Management Concerned with Promotion

For example, we consider an *associate professor role* and a *full professor role* which are defined in a university. In this example, an associate professor Jiro in Osaka University is represented as follows:

Jiro plays an instance of the associate professor role and thereby becomes an associate professor (role holder).

Now, we assume a case where Jiro is promoted from associate professor to full professor. It means that Jiro stops to play the instance of the *associate professor role*, then plays the instance of the *full professor role* and thereby becomes a full professor (role holder). In this example, when Jiro plays the instance of associate professor role or full professor role then, he also plays *teaching staff role* in the university at the same time. Because "full professor role *is-a* teaching staff role" and "associate professor role *is-a* teaching staff role" as shown Figure 3), the semantics of *is-a* relation tells us that Jiro stops to play the instance of teaching staff role at the very moment when he changes the role to play. This is because one has to stop to play the current role

¹ Our role model also supports role concepts depending on multiple contexts [4].

when he/she starts to play the new role. In other words, the continuity of playing the *teaching staff role* is damaged. Obviously, this example model does not capture the behavior of the instances in the real world accurately at all. In the real world, Jiro has been the same teaching staff not only when he is the associate professor but also the full professor.

4.2 Solution

The problem discussed in the previous section is caused by the confusion of *teaching staff role* between *full professor/associate professor roles*. The former means that the player of the role belongs to the university as a staff member, and the latter means a role (post) that its player performs in the university.

Therefore, this problem can be solved by distinguishing between *teaching staff role* and *professor/associate professor role*, and defining them separately in the university as shown in Fig. 4.

Fig. 4. Class definition to solve section 2-(2)

In the school context, *teaching staff role* and *school post role* are defined. The former is a role concept which means the player belongs to the school as teaching staff. An instance of *person* plays the teaching role and thereby becomes a teaching staff role holder. And the latter means a post which a *teaching staff role holder* holds in the school. In the *university* context, which is a sub concept of *school*, *academic staff role* is defined as a subclass of *teaching staff role*, and *full professor/associate professor roles* are defined as subclasses of *school post role*. In comparison with Figure 3, the super concept of *full professor role* and *associate professor role* in Fig.4 is a *school post role*. And their potential player is a *university staff role holder* rather than *person*. This class definition can solve the instance management problem concerned with promotion discussed in section 3.1 as follows.

Firstly, when Jiro is an associate professor of Osaka University, he plays an instance of an *academic staff role* in Osaka University and thereby becomes an *academic staff role holder* (referred to as RH_i). Then, RH_i plays an instance of *associate professor role* and thereby becomes the associate professor (role holder). Next, when Jiro is promoted from associate professor to full professor, he (RH_i) stops to play the *associate professor role* and plays a *full professor role* while he does not stop to play the *academic staff role*, since he comes back to not an ordinary *person* but RH_i which is a role holder of *academic staff role* when he stops to play the *associate professor role*. This problem is thus resolved successfully.

4.3 Constituent Role and Post Role

The instance management problem concerned with post promotion can be generalized to a problem which occurs under following three conditions: 1) there are more than two role concepts in a context which has a common super class, 2) a player stops to play a role instance of one of them, then he/she plays a role instance of another, and 3) the player remains to participate in the same context during the change of roles. The solution to this problem is to distinguish between the two role concepts: a role concept which means its player participates in the context and role concepts which mean roles (posts) that their players occupy perform in the context. We call these two role concepts *constituent role* and *post role*, respectively. Role holders of *constituent roles* are used to denote classes of potential players of *post role*.

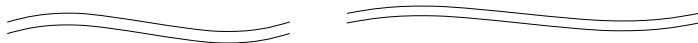


Fig. 5. Class definitions of Constituent role and Part role

In the example discussed section 4.2, *teaching staff role* in a school and *academic staff role* in a university are constituent roles which means their players are participating in the organization (school and university). And, *school post role* in a school, *associate professor role* and *full professor role* are *post roles* which mean roles (posts) are performed in the organization by role holders of the *constituent roles*.

We can find the same problem concerning artifacts. For example, suppose that a front wheel in a bicycle is replaced by a rear wheel in the same bicycle. The replaced wheels change its role while keeping participation in the context. To capture this case adequately, we should define *bicycle part role* as a constituent role and some post roles (e.g. *front wheel role*, *rear wheel role*) played by role holders of the *bicycle part roles*. Figure 5 shows examples of them discussed in this section.

4.4 Context Dependency of Constituent Roles

We can recognize constituent roles according to their contexts. In this section, we discuss contexts on which *constituent role* depends through some examples.

We consider two roles in a company: a *sales member role* in a sales department and *personnel officer role* in a personnel department. Here, we assume a case where Mr. Kimura moves from the sales department to the personnel department in a company. It means that the instance of *person* (Mr. Kimura) stops to play the instance of the *sales member role* then plays the instance of the *personal officer role*. In this example, because it is natural to regard that he has been an employee of the company, we should define an *employee role* as constituent role whose role holder can be potential players of the *sales member roles* and the *personnel officer roles* (Fig. 6.).

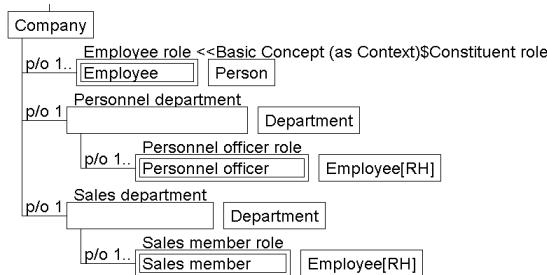


Fig. 6. Constituent roles in company

Furthermore, we assume a case where Mr. Kimura is promoted to the manager of the personnel department. It means that he stops to play the instance of the *personal officer role* then plays an instance of *manager role* in the personnel department while he remains to be a member of the department. To capture this change in an instance model accurately, *staff member roles* should be defined as constituent role and their role holders should be potential players of *personal officer roles* and the *manager role* in the personnel department (Fig. 7.).

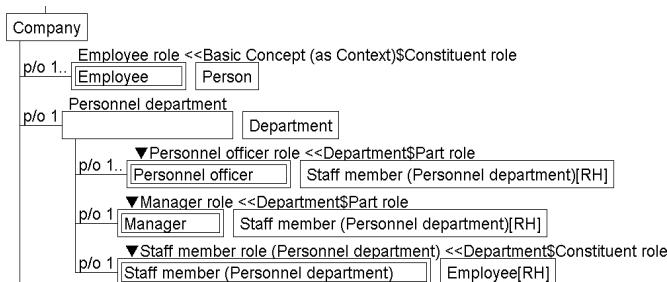


Fig. 7. Constituent roles in personnel department

These examples show that *constituent role* should be defined according to the context in which the player continues to participate in the context when he/she changes roles.

4.5 Ontological Consideration of Constituent Role and Post Role

In the previous section, we discussed that *constituent roles* and *post roles* should be distinguished to manage behavior of instances. The discussion was done from engineering point of view. From philosophical point of view, on the other hand, we can extend our theory of *constituent roles* and *post roles* to all type of entities which have unity. For example, we consider a table composed of a top board and four legs. *Component roles* of the table are defined as *constituent role*, and a *top board role* and *leg roles* are defined as *post roles*.

In the real world, just a set of one board and four sticks cannot be regarded as a table when the table is not constructed using them yet. It means they play only the instances of *component roles* (*constituent roles*) while the instances of the *top board role* and *leg roles* (*post roles*) are not played. In brief, such a status in which players play only the constitution roles of a context is considered just a set of players of the context. In order to make a thing as an independent whole, its properties and functions must be realized by its components, that is, *post roles* must be played by appropriate *component roles*.

In this example, when the table is constructed using the board and sticks, they play the *post roles* (e.g. the *top board role* and *leg roles*) and thereby its properties and functions as an independent whole are realized. Then, it is regarded as not a just set of components in an instance model but an independent whole which has properties and functions as a table.

On the basis of this consideration, we can understand that a members' club also has *constituent roles* and *post role* even if it has only one kind of membership. If the members of a club have only *constituent roles* without *post roles*, it means a just set of persons. To define members' club properly, *post roles* such as *membership roles* should be defined.

As mentioned above, all type of entities which have unity have *constituent roles* and *post roles* from a philosophical point of view. While we can omit to define constituent roles as an engineering approximation in the cases where instance management problems discussed in section 4.1 do not occur, we cannot omit definitions of *post roles* otherwise.

5 Identity of Roles Instances

When the *Prime Minister of Japan* changes from Abe to Fukuda, we can regard that they play the same role as the *head of the Japanese Government*. In our role model, it means the player of an instance of a *head role* in *Japanese Government* changes from Abe to Fukuda and thereby *Japanese Prime Minister role holder* also changes. In this example, while the instance of the *head role* has kept the same identity, the *Japanese Prime Minister role holder* Abe and the *Japanese Prime Minister role holder* Fukuda have different identities because the identity of role holders are generated using the

identity of the role concept and its player. It can explain the sameness and difference between the two Japanese Prime Ministers.

Next, we consider *Diet member roles* as another example. If the number of Diet seats is 480, we can regard there are 480 instances of *Diet member roles*. Whichever instance of the *Diet member roles* is played by a player, the identity of the *Diet member role holder* is not influenced because they have the same right in the *Diet*. However, when 480 persons play the instances of *Diet member roles*, these instances of roles should be distinguished. This suggests that the identities of *Diet member roles* have weaker identity than *head role* in *Japanese Government*.

Therefore, we are considering definitions of some kinds of identities such as ***strong identity*** and ***weak identity***. The strong identity can identify the uniqueness of an instance, and the weak identity only can argue a fact which an instance is different from others. While these definitions are under consideration, they would contribute to an instance management of roles.

6 Related Work

Guarino and his colleagues aim to establish a formal framework for dealing with roles [2,4,5]. Gangemi and Mika introduced an ontology for representing contexts and states of affairs, called D&S, and its application to roles [7]. Their research was concerned with formalities and axioms of an ontology. In contrast, rather than formalizing role concepts, our goal has been to develop a computer environment for building ontologies. Our notions of role concepts share a lot with their theory of roles, especially context-dependence and specialization of roles. According to their theory, our framework can be reinforced in terms of axioms.

Our notions differ from their work on other two points: the dynamics of a role, and the clear discrimination of a role from its player (role holder). Firstly, we focus on context-dependence of a role concept and its categories. So, time dependence of a role concept is treated implicitly in our framework because an entity changes its roles to play according to its aspect without time passing. As opposed to this, the framework by Guarino and colleagues deals with time-dependency explicitly. Secondly, we distinguish *role concepts* from *role holder concepts* [8,9]. On the basis of this distinction, we have developed a tool for properties and relations on roles. Masolo et al. introduced a new kind of entity, called qua-individuals, to solve the counting problem [5]. According to them, qua-individuals would be created each time an entity is classified by a role. So if a person plays two roles, the qua-individuals of the person would be created twice, and he/she would be counted three times as a person and the two roles. Qua-individuals seem to be slightly similar to role holder, but it is unclear how to create their instances and identities, while the notion of role holder does not produce such problems that qua-individual would cause.

7 Conclusions and Future Work

In this paper, we have discussed three problems related to instance management of roles. To solve these problems, we have refined our role model based on an ontological

theory of role. The main contribution is the decomposition of role concepts into *constituent roles* and *post roles*. It can explain the continuity and context dependency of instances of roles when the players change their roles. However, the problems and their solutions are never arbitrary. While the idea for solutions looks simple, its generality is high enough. The problem discussed in this paper appears whenever several role concepts are defined in a same context. The proposed theory can be applied to such cases independently of the role player is human being or things, and the unique solution is obtained. As the result, we explained our role model and its implementation in Hozo can solve these instance management problems appropriately. Because constitution roles and part roles which we introduced in this paper can be represented based on the role model using Hozo, we think their solid foundation and formal definitions are certified.

As future work, we plan to further investigate problems related to appearances/disappearances of roles, an investigation of kinds of identity, a version management of instances of role concepts and their players.

References

1. Gruber, T.: A translation approach to portable ontologyspecifications. In: Proc. of JKAW 1992, pp. 89–108 (1992)
2. Guarino, N.: Some Ontological Principles for Designing Upper Level Lexical Resources. In: Proceedings of the First International Conference on Language Resources and Evaluation, Granada, Spain, pp. 527–534 (1998)
3. Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gengami, A., Guarino, N.: Social Roles and their Descriptions. In: Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2004), pp. 267–277 (2004)
4. Mizoguchi, R., et al.: A Model of Roles within an Ontology Development Tool: Hozo. J. of Applied Ontology 2(2), 159–179 (2007)
5. Masolo, C., Guizzardi, G., Vieu, L., Bottazzi, E., Ferrario, R.: Relational roles and quaindividuals. In: Boella, G., Odell, J., van der Torre, L., Verhagen, H. (eds.) Proceedings of the 2005 AAAI Fall Symposium Roles, an Interdisciplinary Perspective: Ontologies, Languages, and Multiagent Systems, Arlington, VA, Technical Report FS-05-08, pp. 103–112. AAAI Press, Menlo Park (2005)
6. Kozaki, K., Sunagawa, E., Kitamura, Y., Mizoguchi, R.: Role Representation Model Using OWL and SWRL. In: Proc. of 2nd Workshop on Roles and Relationships in Object Oriented Programming, Multiagent Systems, and Ontologies, Berlin, July 30-31 (2007)
7. Gangemi, A., Mika, P.: Understanding the Semantic Web through Descriptions and Situations. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) CoopIS 2003, DOA 2003, and ODBASE 2003. LNCS, vol. 2888, pp. 689–706. Springer, Heidelberg (2003)
8. Kozaki, K., et al.: Development of an Environment for Building Ontologies which is based on a Fundamental Consideration of “Relationship” and “Role”. In: PKAW 2000, Sydney, Australia, pp. 205–221 (December 2000)
9. Kozaki, K., et al.: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of “Role” and “Relationship”. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 213–218. Springer, Heidelberg (2002)

Advancing Topic Ontology Learning through Term Extraction

Blaž Fortuna¹, Nada Lavrač^{1,2}, and Paola Velardi³

¹ Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia

{blaz.fortuna, nada.lavraca}@ijs.si

² University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia

³ Universita di Roma “La Sapienza”, 113 Via Salaria, Roma RM 00198, Italy
velardi@di.uniroma1.it

Abstract. This paper presents a novel methodology for topic ontology learning from text documents. The proposed methodology, named OntoTermExtraction (Term Extraction for Ontology learning), is based on OntoGen, a semi-automated tool for topic ontology construction, upgraded by using an advanced terminology extraction tool in an iterative, semi-automated ontology construction process. This process consists of (a) document clustering to find the nodes in the topic ontology, (b) term extraction from document clusters, (c) populating the term vocabulary and keyword extraction, and (d) choosing the concept names by comparing the best-ranked terms with the extracted keywords. The approach was successfully used for generating the ontology of topics in Inductive Logic Programming, learned semi-automatically from papers indexed in the ILPnet2 publications database.

Keywords: Topic ontology, ontology construction, term extraction.

1 Introduction

OntoGen [1, 2] is a semi-automated, data-driven ontology construction tool, focused on the construction and editing of topic ontologies. In a topic ontology, each node is a cluster of documents, represented by keywords (topics), and nodes are connected by relations (typically, the SubConcept-Of relation). The system combines text mining techniques with an efficient user interface aimed to reduce user’s time and the complexity of ontology construction. In this way, it presents a significant improvement in comparison with present manual and relatively complex ontology editing tools, such as Protégé [3], whose use is hindered by the lack of ontology engineering skills of domain experts constructing the ontology.

Concept naming suggestion (currently implemented through describing a document cluster by a set of relevant terms) plays one of the central parts of the OntoGen system. Concept naming helps the user to evaluate the clusters when organizing them hierarchically. This facility is provided by employing unsupervised and supervised methods for generating the naming suggestions. Despite the well-elaborated and user-friendly approach to concept naming, as currently provided by OntoGen, the approach

was until now limited to single-word keyword suggestions, and by the use of very basic text lemmatization in the OntoGen text preprocessing phase.

This paper proposes an improved ontology construction process, employing improved concept naming by using terminology extraction as implemented in the advanced TermExtractor tool [7,8]. The improved ontology construction process consists of the following steps:

- document clustering to find the nodes in the topic ontology,
- terminology extraction from document clusters,
- population of the terminology vocabulary and keyword extraction, and
- selection of concept names by comparing the best-ranked terms with the extracted keywords.

The proposed approach is illustrated on a case study analysis of the ILPnet2 publications database [4,5], a database of publications in the area of Inductive Logic Programming, extensively gathered for the period of about 20 years. The approach was successful in generating the ontology of topics in Inductive Logic Programming.

The paper is structured as follows. Section 2 describes the ILPnet2 domain used to illustrate the proposed approach to topic ontology construction. Section 3 describes the background technologies, as implemented in the OntoGen and TermExtractor tools. Section 4 presents the proposed methodology, through a detailed description of the individual steps of the advanced ontology construction process. In Section 5 the approach is illustrated by the results achieved in the analysis of the ILPnet2 database.

2 The ILPnet2 Database

The analyzed domain is the scientific publications database of the ILPnet2 Network of Excellence in Inductive Logic Programming [4]. ILPnet2 consisted of 37 project partners composed mainly of universities and research institutes. The entities for our analysis are ILP publications. The ILPnet2 database is publicly available on the Web and contains information about ILP publications between years 1971 and 2003. The data about publications is in the BibTeX format, available in files at <http://www.cs.bris.ac.uk/~ILPnet2/Tools/Reports/Bibtexs/2003,...>, (one file for each year 2003, 2002, ...).

The first stage of the data-driven ontology construction process is data acquisition and preprocessing. The data was acquired with the *wget* utility and converted into the XML format [5]. For this purpose a shell script was implemented. A part of the script that collects the data from the Web is as follows:

```
$ for((i=1971;i<2004;i++)); do
wget
http://www.cs.bris.ac.uk/~ILPnet2/Tools/Reports/Bibtexs/$;
done
```

For easier data management in exploratory analysis of the social network of authors of ILP publications [5], it was convenient to put the data into a relational database format, using the Microsoft SQL Sever. The resulting ILPnet2 database contains the following tables:

- **Authors:** ID (key), name of the author
- **AuthorOf** (relates authors to publications with a many-to-many relation): ID (key), ID of the author, ID of the publication
- **Publications:** ID (key), title, abstract, institution, year, month
- **KeywordIn** (related keywords to publications with a many-to-many relation): ID (key), ID of the keyword, ID of the publication
- **Keywords:** ID (key), keyword

One of the tasks accompanying the database population was the normalization of authors' names. While this was crucially needed for social network analysis (described in [5]), this step was not needed for the experiments in ontology construction described in this paper, as ontology construction uses only document titles and abstracts, preprocessed using a predefined list of stop-words and the Porter stemmer.

3 Background Technologies

This section describes the semi-automated topic ontology generation tool OntoGen, and the term extraction tool TermExtractor.

3.1 OntoGen

The two main design goals of the ontology construction tool OntoGen [1,2,6] are (1) the visualization and exploration of existing concepts from the ontology, and (2) addition of new concepts or modification of existing concepts using simple and straightforward machine learning and text mining algorithms. These design goals are supported by the following two main characteristics of the OntoGen system:

- *Semi-Automatic.* The system is an interactive tool that aids the user during the topic ontology construction process. It suggests concepts, relations between the concepts, and concept names, automatically assigns instances to the concepts, visualizes instances within a concept and provides a good overview of the ontology to the user through concept browsing and various kinds of visualizations. At the same time the user is always in full control of the system and can affect topic ontology construction by accepting or rejecting the system's suggestions or manually editing the ontology.
- *Data-Driven.* Most of the aid provided by the system is based on the underlying data provided by the user typically at the start of the ontology construction process. The data affects the structure of the domain for which the user is building the topic ontology. The data is usually a document corpus, where ontological instances are either documents themselves or named entities occurring in the documents. The system supports automated extraction of instances (used for learning concepts) and co-occurrences of instances (used for learning relations between the concepts) from the data.

The main window of the system (shown in Figure 1) provides multiple views on the ontology. A tree-based view on the ontology, which is intuitive for most users, presents a natural way to represent a topic ontology as a concept hierarchy. This view

is used to show the folder structure and as a visualization offering a one-glance view of the whole topic ontology. Each concept from the ontology is further described by the most informative keywords, automatically extracted (employing unsupervised and supervised learning methods) from the cluster of documents defining the concept.

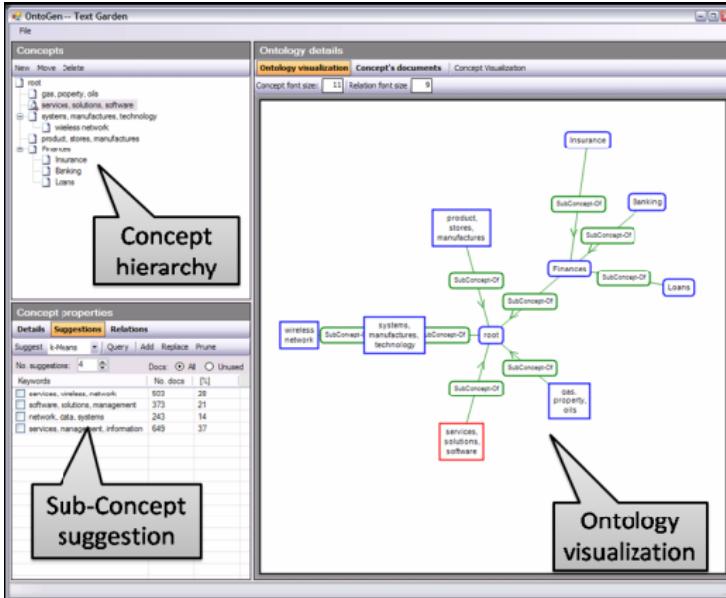


Fig. 1. The user gets suggestions for the sub-concepts of the selected concept (the left bottom part); the ontology is visualized as a tree-based concept hierarchy in a textual mode (the left upper part) and in a graphical mode (the right part)

A sample topic ontology in the form of a tree-based concept hierarchy, constructed from the ILPnet2 documents, is shown in Figure 2. Both the first and the second level of the concept hierarchy were constructed using the k-means clustering algorithm, where the first level was split into 7 concepts and each of these concepts was further split into three sub-concepts. The hierarchical structuring is user-triggered. At each single level, k-means is invoked for various user-defined values of k, then selecting the preferred k and dividing all the documents into k-subclusters, consequently.

While this procedure of ontology construction is elegant and simple for the user, quite some effort is needed to understand the contents and the meaning of the selected concepts. This is especially striking when comparing the second level concepts, for example the sub-concepts of the concept named *logic_program*, *program*, and *inductive_logic* in Figure 2 with the sub-concepts of the concept *logic program* in Figure 3, which shows the concept hierarchy developed by the novel concept naming methodology based on TermExtractor.

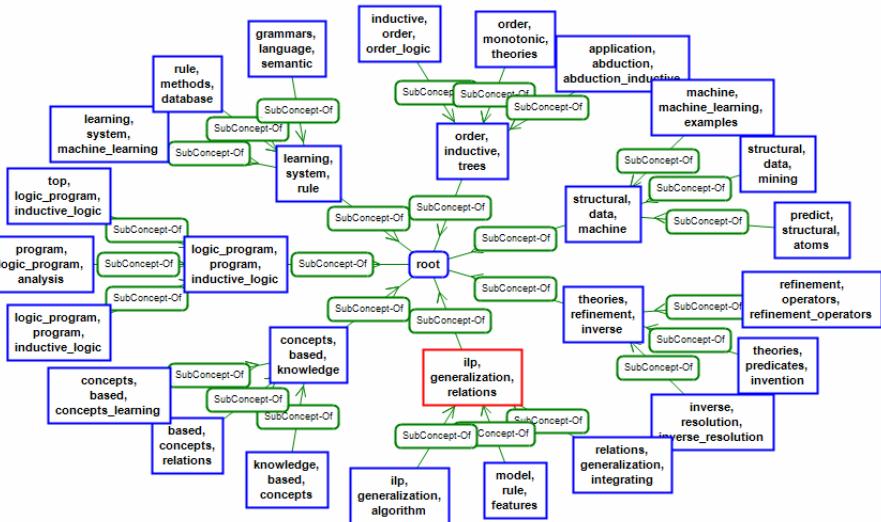


Fig. 2. Ontology constructed by the standard OntoGen approach, constructed from the ILPnet2 publications data, using the k-means clustering algorithm without using the pre-calculated vocabulary extracted by TermExtractor

3.2 TermExtractor

The TermExtractor tool [7,8] for automatic extraction of terms (possibly consisting of several words, as opposed to single keywords) from documents works as follows.

Given a collection of documents from the desired domain, TermExtractor first extracts a list of candidate terms (frequent multi-word expressions). In the second step it evaluates each of the candidate terms using several scores which are then combined and the candidates are ranked according to the combined score. The output is a set of candidates whose score excides a given threshold. Documents from contrast domains are used as extra input for term evaluation and serve as a control group for measuring the term significance. The following scores are used to evaluate candidate terms in the second step (normalized score values are in the $[0,1]$ interval):

- *Domain Relevance* is high if the term is significantly more frequent in the domain of interest than in other domains.
- *Domain consensus* is high if the term is used consistently across the documents from the domain.
- *Lexical cohesion* is high if the words composing the term are more frequently found with the term than alone in the documents.
- *Structural Relevance* is high for terms that are emphasized in the documents (e.g. appear in the title).
- *Miscellaneous* set of heuristics is used to remove generic modifiers (e.g. large knowledge base).

The combined score is a weighted convex combination of the individual scores.

4 The OntoTermExtraction Methodology

There are several ways in which a vocabulary can be acquired. In some domains there already exist established vocabularies (e.g. EUROVOC used for annotating European legislation, AGROVOC used for annotating agricultural documents, ASFA used within UN FAO, DMOZ created collaboratively to categorize web pages, etc.). Another option is automatic extraction of terms from documents, which is especially attractive for the domains where there is no established vocabulary.

Concept and concept name suggestions play a central part in every ontology construction system. OntoGen provides unsupervised and supervised methods for generating such suggestions [1,2,6]. Unsupervised learning methods automatically generate a list of sub-concepts for a currently selected concept by using k-means clustering and latent semantic indexing (LSI) techniques to generate a list of possible sub-concepts. On the other hand, supervised learning methods require the user to have a rough idea about a new topic¹ – this is identified through a query returning the documents. The system automatically identifies the documents that correspond to the topic and the selection can be further refined by the user-computer interaction through an active learning loop using a machine learning technique for semi-automatic acquisition of user's knowledge.

While OntoGen originally used only the input documents for proposing concept suggestions and term extraction techniques for providing help at naming the concepts, it should be noted that the whole process can be significantly improved by constructing a predefined vocabulary from the domain of the ontology under construction. The vocabulary can be used to support the user during hierarchical ordering of concepts, and to create concept descriptions, thus helping concept evaluation.

The rest of this section presents the proposed OntoTermExtraction methodology, through a detailed description of the individual steps of this ontology construction process.

4.1 Steps in the Proposed ontoTermExtraction Methodology for Concept Naming

The advanced ontology construction process, proposed in this paper, consists of the following steps:

- (a) document clustering to find the nodes in the ontology (described in Section 3.1),
- (b) terminology extraction from document clusters (described in Section 3.2), using TermExtractor
- (c) populating the term vocabulary and keyword extraction (described in Section 4.2),
- (d) choosing the concept name (topic) by comparing the best-ranked terms with the extracted keywords (described in the ILPnet2 application in Section 5).

¹ Hereafter we name *concepts* the document clusters generated by the k-means clustering algorithm, while a *topic* is a description of the concept, e.g. a term of a set of terms that best identify the document cluster.

4.2 Populating the Terms and Keyword Extraction

For each term from the vocabulary, a classification model is needed which can predict if the term is relevant for a given document cluster. In this paper we use a centroid-based nearest neighbor classifier [5] which was developed for fast classification of documents into taxonomies. We use this approach since it can scale well to larger collections of terms (hundreds of thousands of terms). A training set of documents is needed to generate a classification model. In some cases vocabularies already come with a set of documents annotated by the terms. In this case these documents can be used for training the term models. When no annotated documents are available, information retrieval can be applied for finding documents to populate the terms.

In this paper we propose using two different techniques to populate terms extracted by TermExtractor.

Let T be the set of terms automatically extracted from document clusters:

- The first technique used the ILPnet2 collection. Each term $t \in T$ was issued in turn as a query and the top ranked documents (according to cosine similarity, using TFIDF word weighting) were used to populate the term.
- The second technique did not use the ILPnet2 collection and relied on Google web search instead. A query was generated from each term t by taking its words and attaching an extra keyword "ILP" to limit the search to ILP related web pages. For example, if t is *inductive logic programming*, the query is *ILP inductive logic programming*. The query was then sent to Google and snippets of the returned search results were used to populate the term.

The ILP vocabulary prepared in this way was used as an extra input to OntoGen, besides the collection of the articles. We tried both approaches but in this paper we only show the results of the second technique, because the retrieval from the whole web turned out to be a richer resource than just the ILPnet2 collection. Details on how the vocabulary looked and how it was applied in the ILP ontology construction are described in Section 5.

5 ILPnet2 Vocabulary and Ontology Construction

In this section, the approach is illustrated by the results achieved in the analysis of the ILPnet2 publications database.

5.1 Vocabulary Extraction

As described in the previous section, we used TermExtractor to automatically extract the vocabulary for the ILP domain from the ILPnet2 collection of ILP publications. Table 1 shows the 11 top-ranked terms (out of 97) extracted from ILPnet2 documents.

All the terms were populated using Google web search. As an example, here are the top 5 snippets that were returned for the query "ILP predictive accuracy":

- Boosting Descriptive ILP for Predictive Learning in Bioinformatics -- general, this means that a higher predictive accuracy can be achieved. Thirdly, although some predictive ILP systems may produce multiple classification ...

Table 1. Top-ranked terms extracted by TermExtractor from ILPnet2 documents

Top-10 terms extracted from ILPnet2	Term Weight	Domain Relevance	Domain Consensus	Lexical Cohesion
inductive logic	0.928	1.000	0.968	0.557
logic programming	0.924	1.000	0.988	0.293
inductive logic programming	0.893	1.000	0.966	0.181
background knowledge	0.825	1.000	0.737	0.835
logic program	0.824	1.000	0.867	0.203
machine learning	0.785	1.000	0.777	0.221
data mining	0.776	1.000	0.691	0.672
refinement operator	0.757	1.000	0.572	1.000
decision tree	0.742	1.000	0.613	0.714
inverse resolution	0.722	1.000	0.557	0.894
experimental result	0.718	1.000	0.594	0.684

- Imperial College Computational Bioinformatics Laboratory (CBL) -- Results on scientific discovery applications of ILP are separated below ... Progol's predictive accuracy was equivalent to regression on the main set of 188 ...
- Evolving Logic Programs to Classify Chess-Endgame Positions -- indicate that in the cases where the ILP algorithm performs badly, the introduction of either union or crossover increases predictive accuracy.
- Estimating the Predictive Accuracy of a Classifier -- the predictive accuracy of a classifier. We present a scenario where meta- Workshop on Data Mining, Decision Support, Meta-Learning and ILP, 2000.
- -*.BibTeX ... -- An outline of the theory of ILP is given, together with a description of Golem Performance is measured using both predictive accuracy and a new cost ...

For each query the snippets of the first 1000 results were used. The snippets served as input for term modeling, described in Section 4.2. The models generated for each term, using this data, were then used for generating the concept suggestions and name suggestions in OntoGen.

5.2 Ontology Learning

First the ILPnet2 collection and vocabulary were loaded into the program. The collection was imported in OntoGen as a directory of files, where each document was a separate ASCII text file (File -> New ontology -> Folder). The vocabulary was loaded using the Tools -> Context menu.

After experimenting with different numbers and with the help of concept visualization, a partition into seven concepts using the k-means clustering algorithm was chosen. For all the seven concepts the first-ranked term suggested from the vocabulary suggested by TermExtractor was selected. This means that the term extraction and population have indeed succeeded to rank the terms in a meaningful way. This is

illustrated also by the following list of discovered concepts, with best-ranked concept names proposed by TermExtractor, followed by the second best-ranked concept name (in parentheses), and the list of most important keywords, as chosen originally by OntoGen:

- Learning system (learning algorithm) -- learning, system, rule, language, methods, machine_learning, machine, approach, ilp, grammars
- Decision tree (logical decision tree) -- order, inductive, trees, order_logic, discovery, decision, application, decision_trees, database, experiments
- Structured data (chemical structure) -- structural, data, machine, predict, examples, relations, machine_learning, mining, definitions, knowledge
- Clausal theory (theory revision) -- theories, refinement, inverse, resolution, predicates, operators, inverse_resolution, invention, refinement_operators, revision
- Relational database (inductive learning) -- ilp, generalization, relations, model, algorithm, constraints, integrating, rule, agent, evaluation

By checking the publication years of articles from different concepts it was possible to analyse the *evolution* of topics. For example, we can notice that most frequent years in concepts clausal theory, concept learning and logic program were around 1994, concepts structured data and learning system were most frequent around year 2000, and concepts decision tree and relational database appear to be most recent in years following 2000. Each of the concepts was further split into sub-concepts using suggestions from the vocabulary which resulted in the two-level taxonomy shown in Figure 3.

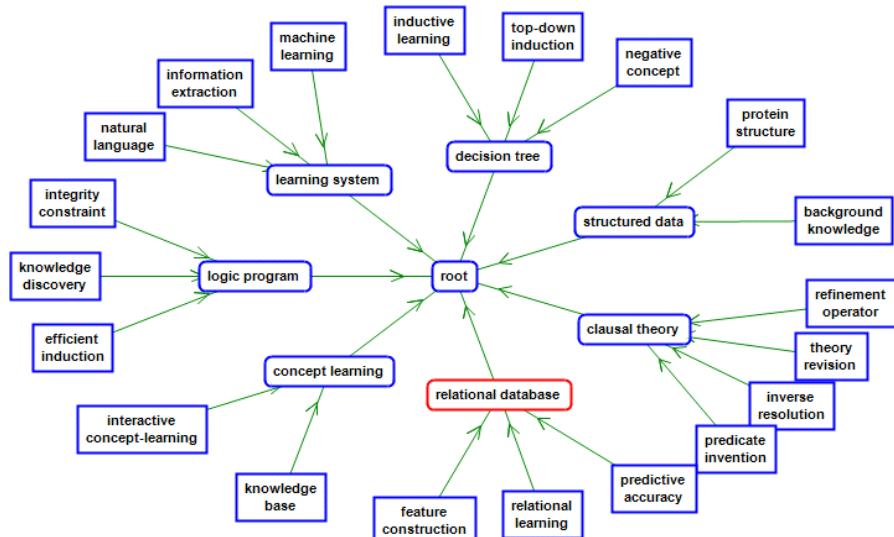


Fig. 3. Ontology constructed on top of ILPnet2 dataset using the pre-calculated terminology

6 Conclusions and Further Work

This paper presents a novel concept naming methodology applicable in semi-automated topic ontology construction, and illustrates the improved concept naming facility on the ontology of topics, extracted from the ILPnet2 scientific publications database. Concept naming supports the user in the task of concept discovery, concept naming and keeps the constructed topic ontology more consistent and aligned with the established terminology in the domain. Further work will be devoted to the evaluation of the constructed ontologies [9].

Acknowledgements

This work was supported by the Slovenian Ministry of Higher Education, Science and Technology project Knowledge Technologies, and the 7FP EU project Bisociation Networks for Creative Information Discovery (BISON). The authors are grateful to M. Grčar, S. Sabo, D.A. Fabjan and P. Ljubič who have transformed the ILPNet2 database into the XML format used in the experiments of this paper.

References

1. Fortuna, B., Mladenić, D., Grobelnik, M.: Semi-automatic construction of topic ontologies. In: Ackermann, M., et al. (eds.) EWMF 2005 and KDO 2005. LNCS (LNAI), vol. 4289, pp. 121–131. Springer, Heidelberg (2006)
2. Fortuna, B., Grobelnik, M., Mladenić, D.: Semi-automatic data-driven ontology construction system. In: Proceedings of the 9th International Multi-conference Information Society, Ljubljana, Slovenia, pp. 223–226 (2006)
3. The Protégé project (2000), <http://protege.stanford.edu>
4. ILPNet2 publications database, <http://www.cs.bris.ac.uk/~ILPnet2/>
5. Sabo, S., Grčar, M., Fabjan, D.A., Ljubič, P., Lavrač, N.: Exploratory analysis of the ILPnet2 social network. In: Proceedings of the 10th International Multi-conference Information Society, Ljubljana, Slovenia, pp. 223–227 (2007)
6. Grobelnik, M., Mladenić, D.: Simple classification into large topic ontology of web documents. In: Proceedings of the 27th International Conference Information Technology Interfaces, Dubrovnik, Croatia, pp. 188–193 (2005)
7. The TermExtractor tool, <http://lcl2.uniroma1.it/termextractor>
8. Sciano, F., Velardi, P.: TermExtractor: A Web application to learn the common terminology of interest groups and research communities. In: Proceedings of the 9th Conference on Terminology and Artificial Intelligence, Sophia Antipolis, France (2007)
9. Mladenić, D., Grobelnik, M.: Evaluation of semi-automatic ontology generation in real-world setting. In: Proceedings of the 29th International Conference Information Technology Interfaces, Dubrovnik, Croatia, pp. 547–551 (2007)

Handling Unknown and Imprecise Attribute Values in Propositional Rule Learning: A Feature-Based Approach

Dragan Gamberger¹, Nada Lavrač^{2,3}, and Johannes Fürnkranz^{4,*}

¹ Rudjer Bošković Institute, Bijenička 54, 10000 Zagreb, Croatia
dragan.gamberger@irb.hr

² Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
nada.lavrac@ijs.si

³ University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia

⁴ TU Darmstadt, D-64289 Darmstadt, Germany
juffi@ke.informatik.tu-darmstadt.de

Abstract. Rule learning systems use features as the main building blocks for rules. A feature can be a simple attribute-value test or a test of the validity of a complex domain knowledge relationship. Most existing concept learning systems generate features in the rule construction process. However, the separation of feature generation and rule construction processes has several theoretical and practical advantages. In particular, the proposed transformation from the attribute to the feature space motivates a novel, theoretically justified procedure for handling of unknown attribute values. This approach suggests also a novel procedure for handling imprecision of numerical attributes. The possibility of controlling the expected imprecision of numerical attributes during the induction process is a novel machine learning concept which has a high application potential for solving real world problems.

Keywords: rule learning, features, unknown attribute value, imprecision of attribute values.

1 Introduction

All real world applications of inductive learning systems are confronted with the problem of unknown attribute values in the training set. The simplest approach is to ignore examples with unknown attribute values. But this approach can be applied only when the number of unknown values is small and the number of

* This work was supported by Croatian Ministry of Science, Education and Sport project “Machine Learning Algorithms and Applications”, Slovenian Ministry of Higher Education, Science and Technology project “Knowledge Technologies”, German Science Foundation (DFG) project FU 580/2 “Towards a Synthesis of Local and Global Pattern Induction (GLocSyn)”, and EU FP6 project “Heartfaid: A knowledge based platform of services for supporting medical-clinical management of the heart failure within the elderly population”.

available examples is very large. A commonly used approach is to replace missing values with a default value in the data preparation phase. For example, CN2 [2] replaces unknown values of discrete attributes by most commonly occurring value (the *mode* value), and unknown values for continuous attributes by the average value (the *mean* value). It is also possible to substitute the unknown values by random values or to try to estimate their values from known values of other attributes in the same example. These and similar techniques have drawbacks, especially when the number of unknown attribute values is high. Extensive experiments presented in [1] demonstrate that none of the mentioned approaches is absolutely superior compared to the others.

An alternative approach suggested in [9] is to *reduce the apparent gain from testing attribute A by the proportion of cases with unknown values of A*. The rationale behind this approach is that testing attribute *A* will yield no information when it has unknown values, and when *A* has unknown values for all examples it is completely useless. The advantage of the approach is that it is theoretically sound. The problem is that its implementation is not a simple task. In decision tree induction we can easily appropriately reduce the information gain for the node testing attribute *A* that includes unknown values, but we do not have an effective method to partition examples with unknown values based on such a node. The consequence is that we can not expect optimal performance in nodes below the one that is based on an attribute that has many unknown values. In decision tree induction experiments reported in [9] the approach had similar prediction quality as those based on substituting unknown values by some known value in data preprocessing.

The approach based on reducing the gain from testing attributes with unknown values seems much more appropriate for covering rule induction approaches. The reason is that we do not have to partition the training set as in decision tree learning, but only to appropriately redefine the used heuristic evaluation measure so that the resulting value will get reduced by the proportion of unknown values. Although the principle is simple, there is a practical problem that, in contrast to decision tree induction, rules typically represent simultaneous decisions based on more than one attribute value and this makes the necessary computations very complex. To the best of our knowledge there is no rule inductive system that implements this approach. Even in [1] where very different approaches for handling unknown values were tested in the rule learning setting, the approach based on reducing the evaluation quality values for unknown attribute values was not mentioned.

In this paper we present a simple and straightforward approach to handling of unknown values in a rule learning setting using covering rule evaluation measures. It is based on a possibility to separate the feature construction process from the rule construction process. In the first phase we construct all potentially useful features and construct covering tables with true and false elements describing covering properties of features on all the training examples. In the second phase, representing the actual rule construction process, we use the information from these covering tables to find combinations of features with optimal covering properties.

The problem of unknown attribute values is solved during the covering table construction. Covering values true and false are set so that covering quality of features is reduced always when the corresponding attribute value is unknown. After that, the second phase is executed in the same way as if all attribute values had known values. It means that we do not need to modify the used heuristic evaluation measures at all. The consequence of appropriately set covering values for features based on unknown attribute values is that standard covering evaluation measures will result in the adequately reduced feature and rule quality. The approach is applicable regardless of the used covering evaluation measure but it is not effective for systems using information gain evaluation measures.

The organisation of the paper is as follows. In Section 2 we introduce the concept of features, we briefly describe an algorithm for the construction of simple features, and illustrate the covering tables construction process. Using this framework we present the novel approach for handling unknown attribute values in Section 3. In Section 4 we exploit the proposed unknown value handling methodology also in handling the imprecision of continuous (numerical) attributes. Finally we describe how the same approach can be applied on ordered nominal attributes with application in DNA gene expression data analysis.

2 Features in Propositional Rule Learning

Features describe properties of examples (instances). An example either has the property or it does not have this property. Thus, features are always Boolean-valued, i.e., either *true* or *false*. Features can be simple literals that test a value of a single attribute, like $A_i > 3$, or they can represent complex logical and numerical relations, integrating properties of multiple attributes, like $A_k < 2 \cdot (A_j - A_i)$, as illustrated in Table 1.

Table 1. Illustration of simple and complex features in a domain with three examples described by three continuous attributes

Attributes			Features	
A_i	A_j	A_k	$A_i > 3$	$A_k < 2 \cdot (A_j - A_i)$
7	1.5	2	<i>true</i>	<i>false</i>
4	3	-4	<i>true</i>	<i>true</i>
1.07	2	0	<i>false</i>	<i>true</i>

It is important to realize that features differ from the attributes that describe instances in the input data. Attributes can be numerical variables (with values like 7 or 1.5) or nominal or discrete variables (with values like *red* or *female*). In contrast with attributes, a feature cannot have a missing or unknown value. As a result, features are different from binary attributes even for binary-valued attributes that have values *true* and *false*.

Note that our use of the term *feature* is not fully aligned with the practice in the machine learning community where terms like *feature extraction*, *feature*

construction or *feature selection* are used for approaches that aim at finding a suitable set of descriptors for the training examples by including expert knowledge, increasing the quality of learning or the expressiveness of the hypothesis language. As most learning algorithms, such as decision tree learners, focus on attributes, the term *feature* is frequently used as a synonym for *attribute*. In this paper, we clearly distinguish between these two terms.

Rule learning algorithms are *feature-based*, because rule learning algorithms employ features as their basic building blocks, whereas, for example, decision tree learning algorithms are attribute-based, because decision trees are constructed from attributes. For many rule learning systems this is not obvious because the process of transformation of attributes into features is implicit and tightly integrated into the learning algorithm [2], [3]. In these systems, feature generation is part of the rule building process. The main reason for this strategy is the simplicity and especially the memory usage efficiency. Explicit usage of features requires that feature covering tables are constructed and these tables may be relatively large. Nevertheless, there are rule learning algorithms that explicitly construct covering tables before starting the rule construction process. A classical example is the LINUS system for converting relational learning problems into a propositional form [6]. In this framework, the concept of literal relevancy has been introduced by [7].

The generation of features for the given set of training examples is the first step in the rule learning process. It can also be viewed as the transformation from the attribute space into the space of features. Although generation of simple features is a straightforward and a rather simple task, generation of an appropriate set of sophisticated features is a hard task which is out of the scope of this paper.

2.1 Feature Generation

An algorithm enabling the generation of simple features from attributes for a two-class classification problem is presented in [4]. The algorithm generates features separately and independently for each attribute. If an attribute is discrete then all distinct values in positive examples are detected and for them features of the form $Att = value$ are generated. Also all distinct values for negative examples are detected and from them features of the form $Att \neq value$ are generated. The number of generated features for a discrete attribute is equal to the sum of distinct attribute values occurring in positive and negative examples.

For a continuous attribute, features are generated in the following way: We identify pairs of neighboring values, where neighboring means that there is no other value between them. From these pairs, we compute the mean of the two neighboring values (*mean_value*). If there are two neighboring values from different classes, then if the smaller of the two values is from the positive class we generate the feature $Att < mean_value$, while if the smaller of the values is from the negative class we generate the feature $Att \geq mean_value$. The number of features generated for a continuous attribute depends on the grouping of classes in the increasing value list but typically the number of generated features is proportional to the number of examples.

Table 2. A part of the covering table generated for the domain with five examples and three attributes. Included are truth values for five out of ten features generated for this domain by the algorithm described in Section 2.1.

Examples Ex.	Class Cl.	Attributes			Covering table					
		A_1	A_2	A_3	$A_1 = \text{red}$	$A_1 \neq \text{red}$	$A_2 \geq 0.9$	$A_2 < 1.7$	$A_3 \geq 1.5$...
p_1	\oplus	red	1.2	4	true	false	true	true	true	...
p_2	\oplus	blue	1.3	2	false	true	true	true	true	...
n_1	\ominus	green	2.1	3	false	true	true	false	true	...
n_2	\ominus	red	2.5	1	true	false	true	false	false	...
n_3	\ominus	green	0.6	1	false	true	false	true	false	...

2.2 Covering Tables

A *covering table* is a table which has examples as its rows and features as its columns. Table 2 presents a part of the covering table constructed for simple features generated for a small domain with only three attributes. It can be noticed that the covering table has much more columns than the corresponding table presenting the examples by the attributes. Conversion from the attribute to the feature space thus presents a significant increase of the space complexity.

The covering table has only values *true* and *false* as its elements. These truth values represent the covering properties of features on the given set of examples. Together with example classes, the covering table is the basic information necessary for the rule induction process. Rule construction from the available set of features can be done by any covering based rule induction algorithm. The actual attribute values are no longer needed for rule construction. The significant difference of explicit feature generation compared to standard approaches is only that we do not generate features from attribute values during rule construction. Instead, the possible features with corresponding covering properties are obtained from the pre-prepared covering tables.

The price of explicit feature generation is the increase of the space complexity of rule learning algorithms. However, the advantages of explicit introduction of features are: a possibility to use the covering tables directly for rule construction, a possibility to introduce feature relevancy and ensure that only most relevant features really enter the rule learning process (described in [8]), and a possibility to solve problems of handling unknown attribute values and imprecision of continuous attributes during feature covering table construction.

3 Unknown Attribute Values

Note that a good feature to be used in rule construction has the property of being true for many positive examples and false for many negative examples. It is possible to formalize this statement in a form of a theorem. To do so, we start by defining the concept of p/n pairs of examples.

Definition 1 (p/n pair). A p/n pair is a pair of training examples p_i/n_j where $p_i \in Pos$ and $n_j \in Neg$, where Pos is the set of positive and Neg the set of negative examples.

Definition 2 (Discriminating feature). Let F denote a set of features. Feature $f \in F$ discriminates a pair p_i/n_j iff feature f correctly classifies both examples, i.e., if feature f has value true for p_i and value false for n_j .

Theorem 1. For training set E and set of features F a complete and consistent hypothesis H can be found using only features from set F if and only if for each possible p/n pair from training set E there exists at least one feature $f \in F$ that discriminates the p/n pair.

The proof of the theorem can be found in [7].

This theorem indicates that a good feature for rule construction is the one that discriminates many p/n pairs. An ideal feature has a property to be true for all positive and false for all negative examples because it discriminates all the p/n pairs. If the feature discriminates no p/n pairs it is completely irrelevant and can be immediately eliminated from the further rule construction process. In the situation when we have an example with the unknown attribute value, and when because of that we can not determine the real feature truth values for the features that test this attribute value, we should set the truth values of these features so that the features do not discriminate all p/n pairs built from the example. In this way, by reducing their covering properties, we directly penalize the features for those and only those examples that have unknown attribute values. If we have an attribute with unknown values for all the examples then all the features that test the attribute will not be able to discriminate any p/n pair and all such features will be irrelevant.

Definition 2 states that a feature discriminates a p/n pair only if the feature is true for the positive and false for the negative example. It means that when we have a positive example for which we can not determine the real feature truth values then we should set them to *false*. In this way the features will be not able to discriminate all p/n pairs built with this positive example. Based on the same reasoning we see that for negative examples we should set feature truth

Table 3. A part of the covering table for the domain from Table 2 which includes a few unknown attribute values

Examples Ex.	Class Cl.	Attributes			Covering table					
		A_1	A_2	A_3	$A_1 = red$	$A_1 \neq red$	$A_2 \geq 0.9$	$A_2 < 1.7$	$A_3 \geq 1.5$...
p_1	\oplus	red	?	4	true	false	false	false	true	...
p_2	\oplus	blue	1.3	2	false	true	true	true	true	...
n_1	\ominus	?	2.1	3	true	true	true	false	true	...
n_2	\ominus	red	2.5	?	true	false	true	false	true	...
n_3	\ominus	green	0.6	1	false	true	false	true	false	...

values to value *true* when we do not know the real value. Only in this way we can ensure that the features will not discriminate all p/n pairs built with this negative example. The approach is illustrated in Table 3.

The consequence of the above procedure is that values *false* introduced in positive examples for unknown attribute values will be interpreted as false negative predictions by rule quality evaluation measures. In the same way, values *true* in negative examples will be interpreted as false positive predictions. Because every reasonable rule quality measure favours rules with many true positive and true negative classifications, the result is the degradation of the computed quality values for rules build from features that rely on attributes with unknown values.

The advantage of the proposed approach is that we do not have to change the rule induction process at all. Rules are constructed from completely specified features, rule quality measures may remain unchanged, and features can be combined in the rules in the same way as if all attribute values were known. Actually we can also incorporate the described approach in the rule induction algorithms that construct features during the rule construction process. But having explicit covering table is a good strategy not only because of handling unknown attribute values but also because it enables the detection and elimination of irrelevant features before the rule induction process starts.

At the end let us notice that the approach can be directly applied also for complex features. For example, feature $A_k < 2 \cdot (A_j - A_i)$ from Table 1 will have unknown value when any of attributes A_i , A_j , and A_k has unknown value. The principle is easily extendable to features of any complexity with included both numerical and logical operations.

4 Imprecision of Continuous Attributes

Another problem encountered in real world rule learning applications is the problem of imprecise values. Imprecision is inherent to most non-integer continuous attributes. There are two main reasons for the necessity of imprecision handling. The first is that some continuous attributes are not or can not be measured with high precision (like some biological properties) or their values are significantly fluctuating (e.g., human blood pressure). In such cases, if the example values are very near to the decision values used in the features then the actual feature truth values for such examples are unreliable. The second reason is that although some attributes like income or human age can be known very precisely, building rules from features that have supporting examples very near to the decision values used in the features may be a bad practice. The reasoning is that such features may lead to rules with significant overfitting, or, in case of descriptive induction, may result in non-intuitive rules like that 20 years old people have significantly different properties from those having 19 years and 11 months.

An approach to deal with inherent attribute imprecision, regardless of which type it is, is to treat attribute values near to the feature decision values as unknown values in order not to allow that such close values will affect feature quality. Such ‘soft’ unknown values are handled as standard unknown attribute

Table 4. A part of the covering table for the domain from Table 2 generated with the assumption of the expected attribute imprecision value $\delta = 0.35$

Examples Ex.	Class Cl.	Attributes A_1 A_2 A_3	Covering table						
			$A_1 = \text{red}$	$A_1 \neq \text{red}$	$A_2 \geq 0.9$	$A_2 < 1.7$	$A_3 \geq 1.5$...	
p_1	\oplus	red 1.2 4	true	false	false	true	true	...	
p_2	\oplus	blue 1.3 2	false	true	true	true	true	...	
n_1	\ominus	green 2.1 3	false	true	true	false	true	...	
n_2	\ominus	red 2.5 1	true	false	true	false	false	...	
n_3	\ominus	green 0.6 1	false	true	true	true	false	...	

Table 5. The example demonstrates the difference in feature covering properties for imprecision values 0.0 and 0.17 respectively. Attributes A_1 and A_2 are very similar but there are significant differences in feature covering properties when different expected attribute imprecision is taken into account.

Examples Ex.	Class Cl.	Attributes A_1 A_2	Features (with $\delta = 0.0$)		Features (with $\delta = 0.17$)	
			$A_1 < 1.95$	$A_2 < 1.95$	$A_1 < 1.95$	$A_2 < 1.95$
p_1	\oplus	1.5 1.5	true	true	true	true
p_2	\oplus	1.6 1.6	true	true	true	true
p_3	\oplus	1.7 1.65	true	true	true	true
p_4	\oplus	1.8 1.7	true	true	false	true
p_5	\oplus	1.9 1.8	true	true	false	false
n_1	\ominus	2.0 2.1	false	false	true	true
n_2	\ominus	2.1 2.2	false	false	true	false
n_3	\ominus	2.2 2.25	false	false	false	false
n_4	\ominus	2.3 2.3	false	false	false	false
n_5	\ominus	2.4 2.4	false	false	false	false

values described in Section 3. It must be noted, however, that the actual attribute value is known, therefore it may happen that for a feature with some decision value it should be treated as unknown, while for some other feature with a different decision value it may be treated as a regular known value. More precisely, appropriate dealing with continuous attributes is such that in case of a feature decision value d we have to treat all attribute values v in the range $d - \delta < v < d + \delta$ as unknown attribute values, for δ being the user-defined attribute imprecision boundary.

For illustration in Table 4 we have assumed during covering table generation that the expected attribute imprecision value is 0.35 ($\delta = 0.35$). This assumption has practical consequences for the feature $A_2 \geq 0.9$ only, resulting by two 'soft' unknown values when attribute A_2 has values 1.2 and 0.6 which are less than δ far from the feature decision value 0.9. It can be noticed that this does not affect the truth values of the feature $A_2 < 1.7$ which has a different decision value.

The example presented in Table 5 demonstrates an artificial situation with two very similar continuous attributes such that from both ideal features discriminating all 25 p/n pairs can be generated. But when the notion of imprecision of

continuous attributes is introduced, the situation may change significantly. The right part of the table presents covering properties of the same features but with imprecision of 0.17. Now none of the features is ideal. The first discriminates 9 and the second 16 p/n pairs. Although with assumed imprecision 0.0 both features seem equally good, with assumed imprecision 0.17 we have clear preference for the second feature. By analyzing the attribute values it really seems that it is better to use attribute A_2 because it may turn out to be a more reliable classifier in an imprecise environment.

It is important to notice that the presented approach to handling imprecision of continuous attributes can be applied also when we have ordered nominal attributes. A good example are biological gene expression domains with presence call (signal specificity) values A (absent), P (present), and M (marginal). Although in this situation attributes have three discrete values A , M , and P , practically they are a formalization of values low, medium, and high. In each decision we have a problem to define if medium (marginal) values will be treated as low or high values. A principally clean solution is a possibility to treat medium values as unknown values. The described methodology can be interpreted also as handling imprecision of numerical attributes by selecting the expected imprecision value so that medium value M is always in the undefined region near to the assumed decision point. The methodology and the results are presented in [5].

5 Conclusions

We have defined features as the basic rule building blocks, and described a simple algorithm for the generation of propositional features that provably generates only the potentially relevant features both for discrete and continuous attributes. Explicit feature generation and presentation of their covering properties in the covering table has several important advantages. The basic idea is that the complete rule construction process can be done using only the information from the covering tables, which are an appropriate representation format for various learning algorithms.

The most relevant consequence is a possibility of systematic handling of relevancy of features with the possibility to detect and eliminate irrelevant features from the process of classification rule learning. In this paper we have demonstrated that explicit definition of features is useful also for systematic handling of unknown attribute values. In contrast to existing approaches which try to substitute an unknown value with some good approximation, we have addressed the problem of handling of unknown values by appropriately defining feature truth values for such attribute values. The approach is simple, applicable also to complex features based on many attributes, and well justified.

The problem of imprecision of continuous attributes is seldom analyzed in the rule learning framework although it can be very relevant for real life domains. The problem is solved so that attribute values near to the feature decision value are treated as unknown attribute values. The approach relies on the user's estimation of the expected imprecision levels and the possibility to efficiently

handle unknown attribute values through covering properties of generated features. There is a possibility to apply the same approach also for nominal values representing classes of numerical values which can be ordered by magnitude.

Signal specificity attributes in gene expression domains with values absent, marginal, and present are a good example of such attribute. It is known that gene expression domains are very prone to overfitting due to a large number of attributes in domains with a very modest number of examples. Extensive experiments in these domains have demonstrated that using signal specificity instead of signal intensity values, especially in combination by handling marginal values as unknown values is a effective method for overfitting prevention [5]. Also, we have applied the described approach for handling imprecision of numerical attributes in different, especially medical domains [4]. By experimenting with various expected imprecision levels in all of these domains the approach has demonstrated to be able to help construct features and rules describing general concepts easily interpretable by humans.

References

1. Bruha, I., Franek, F.: Comparison of various routines for unknown attribute value processing: The covering paradigm. *International Journal of Pattern Recognition and Artificial Intelligence* 10(8), 939–955 (1996)
2. Clark, P., Niblett, T.: The CN2 induction algorithm. *Machine Learning* 3(4), 261–283 (1989)
3. Cohen, W.W.: Fast effective rule induction. In: Prieditis, A., Russell, S. (eds.) *Proceedings of the 12th International Conference on Machine Learning (ICML 1995)*, pp. 115–123. Morgan Kaufmann, San Francisco (1995)
4. Gamberger, D., Lavrač, N.: Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research* 17, 501–527 (2002)
5. Gamberger, D., Lavrač, N., Zelezny, F., Tolar, J.: Induction of comprehensible models for gene expression datasets by subgroup discovery methodology. *Journal of Biomedical Informatics* 37(4), 269–284 (2004)
6. Lavrač, N., Džeroski, S.: *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood (1994)
7. Lavrač, N., Gamberger, D., Jovanoski, V.: A study of relevance for learning in deductive databases. *Journal of Logic Programming* 40(2/3), 215–249 (1999)
8. Lavrač, N., Gamberger, D.: Relevancy in constraint-based subgroup discovery. In: Boulicaut, J.F., De Raedt, L., Mannila, H. (eds.) *Constraint-Based Mining and Inductive Databases*, pp. 243–266. Springer, Heidelberg (2005)
9. Quinlan, J.R.: Unknown Attribute Values in Induction. In: *Proceedings of the 6th International Machine Learning Workshop, ML-1989*, pp. 164–168 (1989)

Fuzzy Knowledge Discovery from Time Series Data for Events Prediction

Ehsanollah Gholami and Mohammadreza Matash Borujerdi

Department of Computer and IT Engineering
Amirkabir University of Technology, Tehran, Iran
egh1360@yahoo.com, borujerm@aut.ac.ir

Abstract. When the time dimension is added to datasets, time series data are obtained. Extracting knowledge from time series data requires special attention to the timing aspects of the data. An interesting activity in the field of knowledge discovery from time series data is predicting the timing of upcoming events. In this paper we present a method for mining fuzzy knowledge from time series data. In contrast to traditional time series analysis methods which largely focus on global models, our method is about the discovery of local patterns in time series. The extracted knowledge will be in the form of fuzzy association rules and it aims at predicting the approximate timing of upcoming events. The proposed method includes cleaning and filtering of time series data, segmenting time series, extracting important features for prediction, further cleaning on feature values, fuzzifying feature values, extracting fuzzy association rules, and pruning the discovered rules. We will show the efficiency of our approach on a stock market dataset.

Keywords: knowledge discovery, time series, fuzzy association Rules, pre-processing, post-processing.

1 Introduction

When the time dimension is added to datasets, time series data are obtained. Extracting knowledge from time series data requires special attention to the timing aspects of the data. Time series data occurs frequently in business applications and in science. Well-known examples are daily stock prices, daily temperature readings, etc.

According to [5], knowledge discovery in databases usually includes the stages of data cleaning and preprocessing, data mining (searching for useful patterns), and post-processing of discovered patterns. One of the interesting activities in knowledge discovery from time series is predicting the timing of upcoming events. In this paper we present a method for mining fuzzy knowledge from time series data. The extracted knowledge will be in the form of fuzzy association rules and it aims at predicting the approximate timing of upcoming events. Before outlining our method we first review some related work in the literature.

In [7], Han *et al.* developed a method for mining segment-wise periodic patterns in time series databases. Their method was based on combination of association rule

mining technique ([2]) and data cube structure, and allowed periodicity with certain confidence. In [8] algorithms for mining partial periodic patterns are presented; which are more frequent than full periodic patterns. The presented method was based on apriori algorithm for mining association rules.

Algorithms for mining inter-transaction association rules are presented in [11] and [16]. They used a fixed sliding window over time series, and the algorithm was able to extract association rules among events that fall in the same sliding window. In [11] experiments on the synthetic data and stock exchange data were done but no prediction results were reported.

Mannila *et al.* investigated the problem of discovering frequently occurring episodes in event sequences ([12]). They defined an episode as a collection of events that occur relatively close to each other in a given partial order. They presented algorithms for discovery of frequent episodes. The main idea was first finding small frequent episodes and then progressively looking for larger frequent episodes. The number of discovered episodes, which have an exponential search space, is limited by the width of the time window and the frequency threshold (both defined by the user).

Change points are an important type of events in time series. These are time points, where there is a change in the parameters of the underlying data model or even in the model itself. The problem of identifying the change points is studied in [6]. The change-point detection algorithm of [6] is based on the recursive binary partitioning of the time segment by using likelihood criteria. Linear regression is used as the underlying model for each segment.

In [10] a methodology for the entire process of knowledge discovery in time-series databases is presented. After cleaning data by signal processing techniques, they extracted association rules for predicting upcoming events by an information-theoretic connectionist network. The set of discovered rules is then reduced by fuzzification and aggregation. They demonstrated their method on two datasets, stock market, and weather data, and reported prediction accuracy of 64.9% on validation set for stock dataset, and 53.1% for weather dataset. But in the case of stock data, they earned only 2 rules after pruning, which both were expressing known facts about data and thus lacking the novelty (an important interest measure in knowledge discovery).

Das *et al.* considered adaptive methods for finding rules relating previously unknown patterns from time series data to other patterns in that series [4]. Their method was based on discretizing the sequence by methods resembling vector quantization by first forming subsequences by a sliding window through the time series and then clustering subsequences by using a measure of time series similarity. They used an algorithm of sequential rule mining for discovering rules, and J-measure as measure of informativeness on discovered rules. They did experiments on some real-life data sets to show that their method will find the interesting rules from the sequences and the method is robust to the change in parameters, but no result of prediction of rules reported.

Another algorithm, ITARM, for inter-transactional association rule mining is proposed in [13]. It uses a FP-tree based and divide-and-conquer approach. After the frequent 1-itemsets is produced, the algorithm separately uses them as constraint conditions to construct compact FP-tree and mine inter-transactional association rules. They defined inter-transactional association rules in multiple time series. They also did some experiments concerning the time and space complexity of the proposed method, but no experiment on the prediction power of extracted rules reported.

In this correspondence, we present a general method for the entire process of knowledge discovery in time-series databases. We first improve quality of time series data by using digital filters. Then we do segmentation on time series data to get a series of intervals. Then we extract relevant features from interval series. We use the method proposed in [10] for segmentation and also we use the same features as [10]. But we detected some problems in the extracted values of the features. In our experiments we found out that extracted features might have problems like outliers and skewness in data. We present solutions that address these problems. Next step is to build fuzzy membership functions. We propose a new method for doing this task. And then we fuzzify dataset using generated membership functions to get a fuzzy dataset. We then use an apriori based algorithm for discovering fuzzy association rules. For post-processing and pruning rules we propose a new measure based on adjustment of minimum confidence parameter in extracting fuzzy association rules.

Our contributions are detecting and modifying the problems of segmentation method of [10], proposing a new way for generating membership functions, and introducing a new interest measure for pruning the discovered rules.

The remainder of the paper is organized as follows. In Section 2, we present some definitions to formalize the problem to be solved. In Section 3, we present the activities that used for preprocessing of time-series data. In section 4 we explain data mining methods used to extract useful knowledge from preprocessed data, and our post-processing technique as well. In Section 5, the method is demonstrated on a data set of stock prices. We conclude our study in Section 6.

2 Problem Statement

In this section we present some definitions to formalize the problem to be solved. A time series is a set of records, such that each record contains some attributes and also a time value which indicates the time point in which observations of attributes are made. If we use a model for describing time series data, two choices are possible: a general model with global parameters for the whole time series, and a model that its parameters change in different time intervals in time series. We use the second choice. In that case, each time interval in time series data is associated with some parameters for the descriptor model.

For example if we describe a time series data by a piece-wise linear model, each interval $[l_i, h_i]$ will be associated with two parameters a_i and b_i so that:

$$T(i) = a_i t + b_i, \quad \forall t \in [l_i, h_i] \quad (1)$$

Where $T(i)$ is the value of time series at time point t in the i -th interval.

An event is defined as a change in the parameters of the descriptor model between two adjacent intervals. The time point that lies between two consecutive intervals is called a change point iff the values of model parameters for two intervals differ significantly. The problem of identifying the change points is known as the *event detection* problem [6].

For example, if in the interval [June 11, July 2] IBM stock value is approximated by $\phi(t) = \alpha_i t + \beta_i$ where $\alpha_i = 0.32$ and in the interval [July 2, August 14] it has

$\phi(t)$ of the same form, but with $\alpha_i = -0.7$, then we say that on July 2 there was an event in IBM stock value wherein the slope changed from 0.32 to -0.7.

3 Preprocessing of Time Series

The preprocessing of time series data usually includes the following steps:

- If necessary, clean the raw data by digital filters to remove additive noise.
- Perform segmentation on the time series data and get an interval series.
- Extract features from each interval, and detect events between pairs of adjacent intervals.
- Further data cleaning on extracted feature values.
- Generate fuzzy membership functions for each feature. Then convert the dataset to a fuzzy dataset using generated membership functions.

3.1 Data Cleaning

Time series data are usually noisy. For example, in the stock time series, the closing price of each day is influenced from daily random fluctuations as well as long-term trends. So we must preprocess the raw data to remove noise and produce cleaner data.

To clean the data, we can use a low-pass filter (LPF) operator that is operator that eliminates the high frequency waves (mostly noisy parts) and leaves only those with low frequency (mostly long-term signal part).

There are several LPF operators, in time and frequency domains. One famous kind of these filters is Exponential Moving Averages, which is defined as follows

$$\begin{aligned} EMA(i, m) &= p \times t(i) + (1 - p) \times EMA(i - 1, m - 1) \\ EMA(i, 1) &= t(i) \end{aligned} \tag{2}$$

Where p , a real number between 0 and 1 is called discount rate, m is the window size, and $t(i)$ is the value of time series at time point i . From the definition it is clear that the filter gives more weights to recent values. We can change the effect of history by setting discount rate. Simple Moving Averages (MA) is another digital filter, which in all coefficients of the filter are same. In our experiments we found out that EMA has an additional advantage over MA that it preserves most of the extrema (which play an important role in segmentation step) in time series.

3.2 Segmentation and Feature Extraction

Before extracting relevant features for prediction, we transform the cleaned time series into an interval series. There are several methods for time series segmentation. Here we use a bottom-up segmentation algorithm from [10]. In this method we will get a piece-wise linear approximation of the time series, in which each segment is associated with several features. Last *et al.* ([10]) introduced three features for each interval: slope, length, and signal to noise ratio (SNR). The segmentation process is as follows [10]:

We should first find extrems in time series data which implies the points where slope changes. Then we eliminate short intervals (less than the user-defined threshold d) by using linear interpolation.

If $t_{i+1} - t_i < d$, then remove t_i and t_{i+1} from T and insert $t_{i,i+1} = (t_i + t_{i+1})/2$ instead. After several passes, we get a minimum interval of over d time units, or

$$t_{i+1} - t_i > d, \forall t_i, t_{i+1} \in T \quad (3)$$

For each resulting interval, we take $\hat{a}(t)$ in its start and end. We can now derive the slope as

$$\alpha_i = \frac{\hat{a}(t_{i+1}) - \hat{a}(t_i)}{t_{i+1} - t_i} \quad (4)$$

And the offset term as

$$\beta_i = \hat{a}_i(t_i) - \alpha_i t_i \quad (5)$$

We now loop over the intervals and merge intervals with similar slope (e.g., slope with difference less than 0.05). That is, If $|\alpha_i - \alpha_{i+1}| < 0.05$, then remove t_{i+1} from T . Calculate each slope again. As a result, we get a set of intervals, each with an associated slope.

Now we get an interval series and we can extract features for each of the intervals. The length feature is defined easily as the length of each interval. The slope for each interval is computed as (4). The Signal to Noise Ratio (SNR) is another important feature in time series. This feature expresses the fluctuations of the series. A high SNR value indicates that the series is unstable and influenced by various parameters and factors. A low value indicates a stable series. We calculate the SNR in the following way over an interval $[t_i, t_{i+1}]$: ([10])

$$SNR_i = \sqrt{\frac{\int_{t_i}^{t_{i+1}} \epsilon^2(t)}{t_{i+1} - t_i}} \quad (6)$$

Where

$$\epsilon(t) = |a(t) - \alpha_i t - \beta_i| \quad (7)$$

Where $a(t)$ is the original function (before filtering), and α_i and β_i are the parameters of the slope function over interval i .

3.3 Further Cleaning on Feature Values

After extracting feature values, we should do some additional data cleaning before getting onto data mining step. In this part we will describe the additional cleaning activities and justifications for doing so.

The first task is to remove outliers, which are data items too different from most of the remaining. We remove outliers because we will use FCM algorithm in generating

membership functions and our experiments show that outliers will strongly mislead the FCM algorithm. We use a simple statistical test to detect outliers. We state that a value x_i is outlier, if it satisfies the following inequality:

$$|x_i - \bar{x}| > 2\sigma \quad (8)$$

Where \bar{x} is the mean of all x_i 's, and σ is the corresponding standard deviation. Our solution for problem of outliers is to simply remove all of them.

As mentioned before we used an algorithm from [10] for segmentation, but by investigating the values of features "length", and "SNR" we found out that there is some positive skewness in values of both features (that was not mentioned in [10]). In our opinion it comes from the segmentation algorithm, when we set a minimum length on each interval and minimum difference between slopes of the consecutive intervals. This skewness can lead to incorrect results when clustering data to make membership functions, and when data mining. So we take logarithm from features length and SNR (which both take only positive values and taking logarithm is possible) to remove this skewness. Of course when using the extracted rules we should not to forget to take logarithm from new data before applying rules on it.

3.4 Generating Fuzzy Membership Functions

To generate fuzzy membership functions, we propose a new method based on genetic algorithms. Proposed method has good efficiency in this solution, and because of using genetic algorithms it produces a near optimal solution.

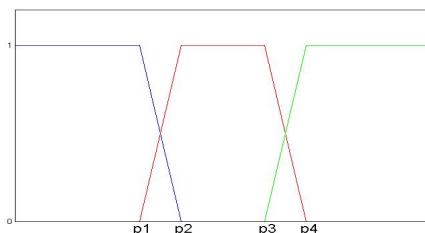


Fig. 1. For each feature we should adjust four parameters

First, we consider three membership functions for features length and SNR, named "Low", "Medium", and "High", and three membership functions for slope, named "Negative", "Zero", and "Positive". For all features we choose the first and last membership functions to be infinite trapezoids and the other to be complete trapezoid. After generating membership functions they will look like what shown in figure 1. From the figure it is obvious that we should adjust four parameters to completely specify membership functions for each feature.

For each feature, we run the FCM (Fuzzy C-Means) algorithm [3] once to cluster its data. In all runs we choose number of clusters to be three, and fuzziness degree to be two, as parameters of the FCM algorithm. After running FCM for a feature, we get a vector V of three elements which include cluster centers, and a matrix U of three

rows and N columns (N is the number of records in dataset). Element at column i and row j in matrix U indicates the grade of membership of data record i to j th cluster.

Now we present our evolutionary method for building membership functions. In our method each chromosome consists of four values that represent four parameters mentioned above. We define fitness function to be maximized as follows:

$$\frac{1}{\sum_{i=1}^N \sum_{j=1}^C (U_{ij} - M_{ij})^2} \quad (9)$$

Where N is the number of training samples, C is the number of clusters in clustering algorithm (3 in our experiments), U_{ij} is membership degree of sample i to cluster j , and M_{ij} is the membership value of sample i to fuzzy membership j corresponding to the parameters of this chromosome.

The intuition behind defining (9) is simple; we try to make membership values generated by target membership functions as close to as possible to membership values obtained from the FCM clustering algorithm. After running genetic algorithm for each feature, we will get membership functions for that feature, and then we use them to fuzzify the dataset. Now we have an Interval series with fuzzy values, and fuzzy sets corresponding to each feature.

In first look we might think that this method is inefficient because of many passes on data. But it works very fast in our problem for two reasons: first, we have only three features; second, the size of dataset is reduced drastically in smoothing and segmentation steps. In experiments we made membership functions for all features in a few seconds.

4 Data Mining and Post-processing

The goal of data mining step in knowledge discovery process is to search the dataset in order to discover interesting patterns from data. Association rules are an important type of such patterns. They were first introduced by Agrawal *et al.* [1] for analyzing market basket data, and finding items frequently bought together in buying transactions. An example of such rules could be that "90% of customers, who buy milk, also buy bread at the same transaction".

In this paper we extract knowledge from time series in the form of fuzzy association rules. So we will first introduce them and then we'll explain how we utilized them for time series prediction.

4.1 Fuzzy Association Rules

The early goal of association rules was to analyze customer transactions in order to discover important associations among items so that the presence of some items in a transaction will imply the presence of some other items. To achieve this goal, Agrawal and his co-workers proposed apriori algorithm ([2]). Apriori algorithm was just able to discover Boolean association rules, which implies only the presence of items together. But in real world problems, quantities of items are important too. In [14] an algorithm for discovering quantitative association rules was presented. The

idea was to divide the domain of quantities for each item into intervals and discretize dataset, and then finding associations between intervals. But this solution had some drawbacks. First, assigning proper meanings to these intervals is very difficult. Second, it may be problems for values lie at the interval boundaries, for example if we define that people younger than 40 years to be young, then a person having 39 years old has no difference with a 20 years old person.

Fuzzy association rules came up to address these problems. In this case one can establish fuzzy membership functions for each attribute which are more meaningful than intervals because of using linguistic terms; furthermore the problem of boundaries does not arise because of the smooth behavior of the membership functions at the boundaries.

There are many algorithms for discovering fuzzy association rules. We used FTDA [9] which is apriori based in nature, but regarding to the fact that it uses for each item, only the linguistic term with maximum scalar cardinality in later mining process, and we don't want this, so we modified it a little to use all linguistic terms for each item in mining process.

4.2 Data Mining on Time Series

After data preprocessing we get to an interval series and features associated to each interval. Now we should extract rules that imply associations between features of consecutive intervals. An example of such rule will be "If the stock price was increasing for a long time and then it begins to fall and the SNR value for this new time interval is high, it is most probable that the stock price rise again after a short time". This kind of rule differs from traditional association rules that were just able to discover associations among items in the same transaction. This kind of rule is called inter-transaction association rule in literature ([11, 16]). In order to discover these rules from time series data, we utilize a sliding window on the consecutive intervals. Intervals lying in the same window will compose a single transaction. So we get an augmented dataset which holds not only feature values but the ordering information as well. Then we can apply any of the algorithms of discovering ordinary association rules on this transformed dataset to find association rules. As mentioned earlier we've used a modified version of the FTDA algorithm for this purpose.

4.3 Post-processing

We propose a new measure for pruning extracted association rules. The proposed method is based on adjusting the minimum confidence parameter, which is used in discovering association rules from large item-sets.

In our experiments we found out that decreasing the minimum confidence will result in a larger number of rules and increase in prediction accuracy of rules on test data set, until we get to a special value for minimum confidence that if we continue decreasing the minimum confidence below it, the number of rules increases but the prediction accuracy will remain fixed. It means that the additional rules generated for minimum confidence values below this special value, are redundant since they don't improve the prediction accuracy. So we get to an optimum value for minimum confidence parameter, which using it in making rules will implicitly prune the redundant

rules. To get this optimum value for minimum confidence, we first set aside a small fraction of dataset as validation set. Other steps are outlined in figure 2.

```

set best_pred = 0 (best prediction accuracy so far)
set best_minConf = 1
for minConf = 1 downto 0 step ΔminConf (minimum confidence value)
begin
    Extract association rules from the set of large itemsets for minConf as minimum confidence.
    set pred = prediction accuracy of extracted rules on the validation data set.
    if best_pred < pred then begin
        set best_pred = pred
        set best_minConf = minConf
    end
    else return best_minConf as optimum value of minimum confidence.
End

```

Fig. 2. Pseudo code for computing optimum minimum confidence

The proposed measure has good efficiency, because making rules from large itemsets for a given minimum confidence is done very fast (we have passed the time consuming stage of the extracting large itemsets already). And also applying rules on validation data is fast, because the validation data is usually a small fraction of the entire data. In our experiments we were able to get the optimum value for minimum confidence just in a few seconds.

5 Experiments

We demonstrate our method on a stock market dataset. The dataset includes closing prices for the stock for each day. Stock prices are noisy and influenced daily by many factors, So Data cleaning must be done prior to any data mining process.

Our main interest is to predict the approximate timing of future events, i.e., the points at which the stock will change the direction of its slope. In this section we describe the process of knowledge discovery in the database of the daily value of stocks from Standard & Poor's index [15], which is traded in NYSE and NASDAQ.

We held out one third of the data as validation set for estimating the prediction accuracy of proposed method, and for post-processing of rules as mentioned in 4.3. The knowledge discovery is performed as follows:

- 1) Data Cleaning: As mentioned above, stock prices are noisy and influenced daily by many factors. So we should first clean the initial dataset by a low-pass filter. We tested Simple Moving Averages (MA) and Exponential Moving Averages (EMA) on the dataset. We chose window size of 21 time points for both filters and discount rate of 0.5 for EMA. We got these values after several trials. Results show that EMA has an advantage over MA that it preserves the extrema much better than MA, and regarding to the key role of extrema in segmentation stage, EMA will suit better to our solution.

2) Segmentation and Feature Extraction: Now we can do segmentation on time series and extract features. We did this by method described in section 3.2.

3) Further data cleaning: as we mentioned earlier in section 3.3 we should do additional data cleaning in order to remove outliers and modify skewness in data. The outlier removal is done by statistical measure introduced in section 3.3 (formula (8)).

In our experiments we realized that there is somewhat positive skewness in extracted features "length" and "SNR". To come up with this problem we take logarithm from the values of these two features. We chose the base of logarithm to be 10 for both features.

4) Generating fuzzy membership functions: Now the data set is ready for generating fuzzy membership functions. We run the FCM algorithm for each feature and by the use of evolutionary method introduced in section 3.4; we can find fuzzy membership functions for each feature. Now we use these membership functions to get a fuzzy dataset which consists of fuzzy transactions.

5) Data Mining: We have a set of fuzzy transactions that should be searched for finding association rules. We use an apriori based algorithm as mentioned in section 4.1. So we get fuzzy association rules describing local associations in time windows. We chose the value of the minimum support parameter of the association rule mining algorithm to be 0.15. The value of the minimum confidence parameter will be used in the next step for pruning redundant rules. Some of the extracted rules are shown in figure 3; for example at the first line we see a rule that its confidence is 0.84 and indicates that if at the first interval (two intervals ago) SNR be low, and at the third interval (current interval) SNR be high and slope be negative (fall of data), then we expect that current condition last for a long time (high length for the current interval).

High SNR(3) & Low SNR(1)	& Negative Slope(3) → High Length(3)	0.84
High SNR(3) & Mid Length(2)	& Negative Slope(3) → High Length(3)	0.89
High SNR(3) & Negative Slope(1) & Positive Slope(3)	→ High Length(3)	0.85
Low Length(1) & Low Length(2)	& Mid SNR(1) → Low Length(3)	0.84

Fig. 3. Some of the extracted rules and their confidence

6) Post-processing: In this step we use our proposed method in section 4.3 for pruning redundant rules. To demonstrate our method we decrease the value of minimum confidence gradually starting from 1 down to 0.3, and count number of rules, and then compute the prediction accuracy for each minimum confidence value by applying rules on the validation set. By applying this procedure in the case of stock market data, the number of rules and the prediction accuracy for each value of minimum confidence is shown in figures 4.a and 4.b respectively. From the figures it is obvious that the optimum value for minimum confidence in this dataset is about 0.6, since there we have the maximum prediction accuracy and minimum number of rules. In other words extra rules discovered for minimum confidence values smaller than 0.6 are all redundant, because they don't improve our prediction accuracy.

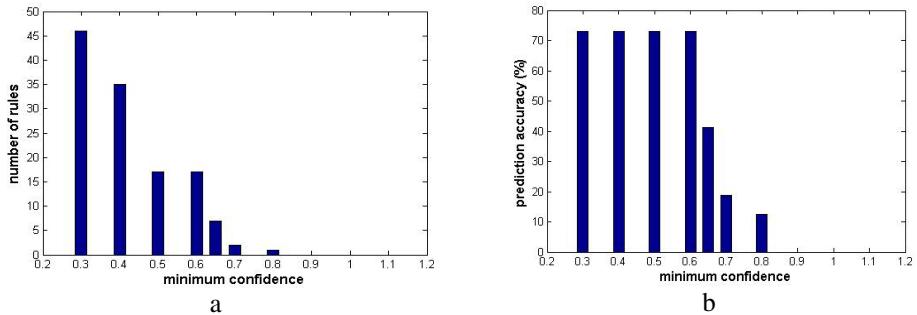


Fig. 4. a. number of rules for each given value of minimum confidence b. prediction accuracy for each given value of minimum confidence

As mentioned before, we held out one third of the data as validation set for estimating the prediction accuracy. We get prediction accuracy of 73.09% on validation data in stock data, which is better than that of [10] on the same dataset which was 64.9%. Furthermore Last *et. al.* in [10] presented only 2 rules as the knowledge discovery result, both expressing the known aspects of the dataset and thus lacking the novelty (an important interest measure). In the case of our method, there are 17 rules that include all informative patterns in dataset due to our interest measure.

6 Discussion and Conclusions

In this paper we introduced a methodology for knowledge discovery in time series data. We did some major enhancements with respect to previous methods. We discovered some probable problems that might arise in time series data as the result of segmentation, such as outliers and skewness, and methods to come up with them.

We proposed a new method for generating fuzzy membership functions that has good efficiency in the case of our solution for time series mining. We used an apriori based algorithm for discovering association rules as knowledge concerning given time series. We proposed a new method for pruning discovered rules, which is based on adjustment of minimum confidence to an optimum value.

But there are many other aspects of the time series mining left. Mining knowledge from multi dimensional time series would be a valuable activity that we didn't addressed it here. Some data mining methods incorporate domain experts into mining process to make knowledge discovery a more subjective task. Doing so in the case of different applications e.g. incorporating a stock expert's prior knowledge about the domain will make the process an interactive one, probably result in higher quality in extracted knowledge.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD Conference on Management of Data, Washington, D.C (1993)

2. Agrawal, R., Srikant, R.: Fast Algorithms for mining association rules. In: Proceedings of the VLDB Conference, Santiago, Chile (1994)
3. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1981)
4. Das, G., Lin, K.I., Mannila, H., Renganathan, G., Smyth, P.: Rule discovery from time Series. In: International Conference on Knowledge Discovery and Data Mining (KDD 1998), pp. 16–22. AAAI Press, Menlo Park (1998)
5. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery: An overview. In: Advances in Knowledge Discovery and Data Mining, pp. 1–30. AAAI/MIT Press, Cambridge (1996)
6. Guralnik, V., Srivastava, J.: Event detection from time series data. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 33–42 (1999)
7. Han, J., Gong, W., Yin, Y.: Mining segment-wise periodic patterns in time-related databases. In: Proceedings of the 1998 International Conference on Knowledge Discovery and Data Mining (KDD 1998), New York, pp. 214–218 (1998)
8. Han, J., Dong, G., Yin, Y.: Efficient mining of partial periodic patterns in time series database. In: Proceedings of the 1999 International Conference on Data Engineering (ICDE 1999), Sydney, Australia (1999)
9. Hong, T.P., Kuo, C.S., Chi, S.C.: Mining association rules from quantitative data. Intelligent Data Analysis 3, 363–376 (1999)
10. Last, M., Klein, Y., Kandel, A.: Knowledge discovery in time series databases. IEEE Transactions on Systems, Man, and Cybernetics 31(1), 160–169 (2001)
11. Lu, H., Han, J., Feng, L.: Stock movement prediction and N-dimensional inter-transaction association rules. In: Proceedings of the 1998 SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD 1998), Seattle, pp. 12:1–12:7 (1998)
12. Manilla, H., Toivonen, H., Verkamo, A.I.: Discovery of frequent episodes in event sequences. Data Mining Knowledge Discovery 1(3), 259–289 (1997)
13. Qin, L.X., Shi, Z.Z.: Efficiently mining association rules from time series. International Journal of Information Technology 12(4) (2006)
14. Srikant, R., Agrawal, R.: Mining quantitative association rules in large relational tables. In: Proceedings of the ACM-SIGMOD 1996 Conference on Management of Data, Montreal, Canada (1996)
15. Standard & Poor's Index, <http://www.spglobal.com/>
16. Tung, A.K.H., Lu, H., Han, J., Feng, L.: Breaking the barrier of transactions: Mining inter-transaction association rules. In: Proceedings of the KDD 1999, pp. 297–301. ACM, New York (1999)

Evolution of Migration Behavior with Multi-agent Simulation

Hideki Hashizume, Atsuko Mutoh, Shohei Kato, and Hidenori Itoh

Dept. of Computer Science and Engineering, Graduate School of Engineering,
Nagoya Institute of Technology,
Gokiso-cho Showa-ku Nagoya 466-8555 Japan
`{hasizume,atsuko,shohey,itoh}@juno.ics.nitech.ac.jp`

Abstract. We describe an artificial ecosystem consisting of five areas, evolving artificial creatures (called agents), and foods for the agents. The ecosystem was constructed for an analysis of migration behavior of the Monarch butterfly. We also report our simulation results on the emergence of the migration biology pertaining to the Monarch butterfly. We use real temperature data as an environmental parameter to define the environment in the field. To adapt to the environment, the agents have temperature sensors. We conducted two experiments using this ecosystem. The results show that the biology of the Monarch butterfly has been well modeled by the ecosystem and our evolutionary method.

1 Introduction

An evolution-based approach to ecological modeling and the simulation-based analysis of multiagent behaviors are an important means to understanding the origins and evolutionary processes of real creatures. This is because a simulation enables us to rapidly and repeatedly experiment with a virtual world. However, a real creature is too complex to program on computers. This naturally demands that we grasp the essence of a real creature, simplify it, and program it as a virtual creature that we call an agent [1]. In this approach, even if an agent initially has only a simple mechanism, it can evolve and obtain complex behaviors through the evolutionary process [2]. This kind of synthetic method could give us possible answers as to why or how real creatures obtain complex behaviors. Many studies have reported on the biology and behavior of real creatures with this approach. For example, from the aspect of animal behavior, foodforaging [3] and herding [4] are well-researched. Other examples are studies of specific creatures, such as egrets [5], magicicadas [6], and the Monarch butterfly [7,8].

The Monarch butterfly (*Danaus plexippus* L., Nymphalidae, Lepidoptera) is a good target for study [9,10]. It has an interesting ecology. The Monarch butterfly is a migratory butterfly that requires three or four generations per annual migration. We will describe the ecology of the butterfly in what follows [11]. The Monarch butterfly mainly lives in Central and North America. As winter ends and spring begins, spring populations of the Monarch butterfly in Mexico prepare for migration and start to migrate north. The migrating females lay eggs

Table 1. Five areas

Name	Model place		
<i>area₄</i>	N 52°	Saskatoon	(the southern part of Canada)
<i>area₃</i>	N 45°	Minneapolis	(the northern part of U.S.A)
<i>area₂</i>	N 39°	Kansas City	(the central part of U.S.A)
<i>area₁</i>	N 30°	Austin	(the southern part of U.S.A)
<i>area₀</i>	N 20°	Mexico City	(Mexico)

and repeat the alternation of generations. In the beginning of summer, some of them reach the southern part of Canada. Fall populations, which are born at the beginning of fall, are known to be biologically and behaviorally different from spring populations. Fall populations migrate south back to Mexico in only one generation. This travel is more than 3500 km. This means that the fall populations fly 50 km per day. In addition, the Monarch butterfly repeats their migration the following year. The butterfly migrates as described and its migration is pretty different from that of a bird. We stated before that the Monarch butterfly requires three or four generations per annual migration. This means that the migration route cannot be taught by the parents. Therefore, we can infer that migration behavior has been caused through the evolution process. It is believed that one of the selection pressures that has driven this species' evolution was food shortage. The non-migratory butterfly slowly evolved and adapted to the environmental change and then became a migratory butterfly over time.

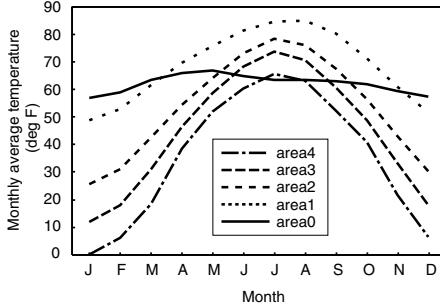
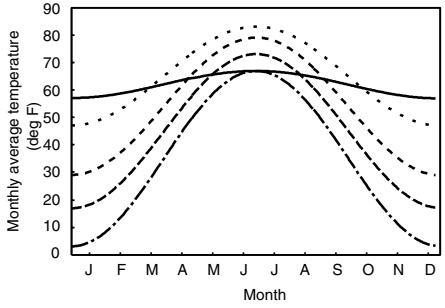
In this paper, we consider the Monarch butterfly as our agent. The agents have an environmental adaptation scale and an action decision table as their genetic components. These genetic components change through evolution and we assume the agents will be flexible to a dynamic environment. The term "dynamic environment" here is both short-term change: seasonal temperature changes and existence of food that is for these agents. Within this environment, we observed the agent's evolutionary process and evolved behavior, and finally, we introduce some speculations about the agent's behavior.

2 The Ecosystem

The ecosystem consists of five areas, plants, and agents. The agent performs one action in one day, which is a unit time. In addition, one year is defined as a certain number of fixed days. Let *DAY* represent these certain days.

2.1 The Area

Definition. An area consists of a two-dimensional 50×50 grid of square locations. Our ecosystem has five areas. These five areas are *area₀*, *area₁*, ... , and *area₄* and are located from south to north in sequence. These areas are modeled after Central and North America (Table 1). The area has plants, agents, and temperature as an environmental parameter. We express *area_i* (*i* : identifier) as

**Fig. 1.** Real temperature data [12]**Fig. 2.** Approximate temperature data

$$area_i(\mathbf{Agent}_i, \mathbf{Plant}_i, tmpr_i). \quad (1)$$

\mathbf{Agent}_i is a set of agents, \mathbf{Plant}_i is a set of plants (we will describe the agents and plants later), and $tmpr_i$ is the temperature.

Seasonal Temperature Change. The temperature, $tmpr_i$, changes periodically for short-term like seasonal change. We call it short-term change. For a short-term change, we use real temperature data from Central and North America. Figure 1 shows the real temperature data, but it is an average of the monthly data. We wanted averaged daily data, so we approximated the real temperature data into an approximate average daily data using a sin function (Figure 2). The temperature: $tmpr_i(d)$ in $area_i$ for day d is determined by

$$tmpr_i(d) = \alpha_i \sin(2\pi d/DAY) + \beta_i, \quad (2)$$

where α_i and β_i are constant numbers in each area.

Air Current. The ecosystem has an air current from $area_4$ to $area_0$. The air current helps an agent to move in the same direction. Therefore, if the agent starts migrating south from $area_1$ - $area_4$, it directly lands on $area_0$.

2.2 The Plant

Definition. A plant p_j (j : identifier) is expressed by

$$p_j(age_j), \quad (3)$$

where age_j is the number of days for which the plant has existed. The plant is the energy source for the agent.

Appearance & Disappearance. The birth number of plants: N_i^{plant} in $area_i$ is determined by

$$N_i^{plant} = M_i^{plant} \times (1 - \Delta_{suit_tmpr^{plant}}/S_t), \quad (4)$$

$$\Delta_{suit_tmpr^{plant}} = |tmpr_i(d) - suit_tmpr^{plant}|, \quad (5)$$

Table 2. Sensory information

Info	Question	Alternatives
Internal information	Enough energy?	$X_0 = \text{Yes/No}$
External information	Find plants?	$X_1 = \text{Yes/No}$
	Find other agents?	$X_2 = \text{Yes/No}$
	Temperature?	$X_3 = \text{Hot/Cold/Suitable}$

where M_i^{plant} is the maximum birth number of plants in $area_i$ in one day, S_t is a constant, and $suit_tmp^{plant}$ is the most suitable temperature for the plant. As you know from Eq 4, the temperature in the focused area determines the birth number of plants. In our simulation the upper limit of the birth number in each area is different. The birth number in $area_0$ is smaller than the others¹. After the birth number is determined, each plant is set in a random grid.

If the plant suffers either of the following conditions, it is removed from the simulation. T_d^{plant} is the maximum lifetime of plants.

$$\text{"An agent eats the plant"} \quad \text{or} \quad age_j > T_d^{plant} \quad (6)$$

2.3 The Agent

Definition. An agent a_k (k : identifier) is expressed by

$$a_k((ea_k, st_table_k), (age_k, in_k)), \quad (7)$$

where ea_k is the environmental adaptation scale, st_table_k is the action decision table, age_k is the number of days for which the agent has existed, and in_k is the energy level. The first two elements, ea_k and st_table_k , are inherited. It should be noted that ea_k and st_table_k are encoded into different genes. However, the last two elements, age_k and in_k , are not inherited. age_k and in_k are initialized when the agent is born.

Environmental Adaptation Scale. The environmental adaptation scale (abbreviated EA): ea_k is an integer fulfilling $0 \leq ea_k \leq M^{ea}$ (let M^{ea} be the maximum number of EA). We use the EA to represent the physical features of the agent, especially the “thickness of the skin.” If the EA is large, it means that the agent can stand a large temperature variation, and it also means that the agent is heavy and its actions consume a lot of energy. If the EA is small, it conversely means that the agent cannot stand a large temperature variation and nevertheless the agent is light. We use the EA to implement for the agent a sensory system that enables it to feel temperature.

Senses. The agent, a_k , senses the information (Table 2). The information is the internal information of its own energy level, in_k , and the external information,

¹ It is because the shortage of food or milkweed in the southern area is thought to have caused the migration of the Monarch butterfly.

ex_k . The internal information is on whether the agent has the energy level in the condition of “ $in_k > I_e$.” I_e is a certain energy level. The external information is on “finding an agent,” “finding a plant,” and “how the agent feels about the temperature.” To sense all of the information, the agent has visibility around constant grids to find other agents and plants. The agent also has a temperature sensor to feel whether $tmpr_-()$ is “hot,” “cold,” or “suitable.” $suit_tmpr_zone_k^{agent}$ categorizes the temperature:

$$S_b - (K_s/M_{ea}) \times eak \leq suit_tmpr_zone_k^{agent} \leq S_b + (K_s/M_{ea}) \times eak, \quad (8)$$

where S_b and K_s are constant values to construct $suit_tmpr_zone_k^{agent}$. If $tmpr_-()$ is within the range of $suit_tmpr_zone_k^{agent}$, the agent feels the area is “suitable.” If $tmpr_-()$ exceeds the maximum temperature of $suit_tmpr_zone_k^{agent}$, it feels the area is “hot.” Also, if $tmpr_-()$ is below the minimum temperature of $suit_tmpr_zone_k^{agent}$, it feels the area is “cold.”

Action Decision Table. Five actions: “eat (E)”, “reproduce (R)”, “migrate north (Mn)”, “migrate south (Ms)”, and “do nothing (N)” can be performed by the agent. The agent, a_k , decides which action to perform by using the following:

$$act_k = st_table_k(X_0, X_1, X_2, X_3), \quad (9)$$

where st_table_k is the action decision table and X_- is a sensory information. An example of action decision table is shown in Table 3.

Energy Level Update. After the action at day d , $in_k(d)$ is updated by

$$in_k(d) = in_k(d-1) + f(act_k, eak, \Delta suit_tmpr^{agent}), \quad (10)$$

where $\Delta suit_tmpr^{agent}$ is the difference in temperature between $tmpr_-()$ and the edge of $suit_tmpr_zone_k^{agent}$ given by

$$\Delta suit_tmpr^{agent} = \begin{cases} tmpr_-() - \max(suit_tmpr_zone_k^{agent}), & \text{“Hot”} \\ \min(suit_tmpr_zone_k^{agent}) - tmpr_-(), & \text{“Cold”,} \\ 0, & \text{“Suitable”} \end{cases} \quad (11)$$

where function f is the update function of the energy level. If the agent with eak performs act_k under the condition of $\Delta suit_tmpr^{agent}$, the function outputs the necessary amount of change in the energy level; (a) decreasing a certain amount of energy: “reproduce,” (b) decreasing its energy level in proportion to eak : “migrate north/south” and “do nothing,” and (c) increasing a certain amount of energy: “eat.”

Reproduction. Two agents, a_{p1}, a_{p2} , reproduce an offspring agent a_k with

$$\begin{aligned} a_k((eak, st_table_k), (0, I_f)), \\ eak = mu_{ea}(cr_{ea}(ea_{p1}, ea_{p2})), \\ st_table_k = mu_{st}(cr_{st}(st_table_{p1}, st_table_{p2})), \end{aligned} \quad (12)$$

where cr_{ea} and cr_{st} are the crossover functions for eak and st_table_k , respectively. mu_{ea} and mu_{st} are the mutation functions for eak and st_table_k , respectively. age_k is initialized by 0, and in_k is initialized by I_f which is initial energy level.

Table 3. An example of action decision table. The agent performs a checked action. Key: Y=Yes, N=No, H=Hot, C=Cold, S=Suitable.

		1	2	3	4	5	...	24
Sensory Information	X_0	Y	Y	Y	Y	Y	...	N
	X_1	Y	Y	Y	Y	Y	...	N
	X_2	Y	Y	Y	N	N	...	N
	X_3	H	C	S	H	C	...	S
Actions	E			✓			...	
	R			✓			...	
	Mn	✓					...	
	Ms		✓				...	
	N				✓	...	✓	

Table 4. Parameters setting

Symbol	Value
$DAY(1year)$	300
$(\alpha_i, \beta_i); i = 4$	(32,35)
$; i = 3$	(28,45)
$; i = 2$	(25,54)
$; i = 1$	(18,65)
$; i = 0$	(5,62)
T_d^{plant}	30
T_d^{agent}	100
M_0^{plant}	7
$M_{1,2,3,4}^{plant}$	30

Death. If the agent suffers either of the following conditions, it dies and is removed from the simulation. T_d^{agent} is the maximum lifetime.

$$in_k \leq 0 \quad \text{or} \quad age_k > T_d^{agent} \quad (13)$$

3 Experiment

In this section, we present the details of an experiment carried out using our defined ecosystem. The purpose of this experiment is to observe how the agents evolve and what adaptive behaviors the agents obtain in an environment that has short-term change and is locally bias of food distribution. The parameter setting is listed in Table 4.

3.1 Result

The results are shown in Figure 3. We executed a simulation using our defined ecosystem for a period of 2000 years. It is difficult to describe all the results for the entire period. So in Figure 3, we extracted 2 years from the 2000 years, and now are providing a detailed analysis. Figures 3(b)-(f) show the population changes and the number of migrate actors. As is evident from these figures, the agents obtained three emergent behaviors and adapted to the environment.

Stay in $area_0$. Figure 3(f) shows that some agents stayed in $area_0$ throughout the year. This means that a certain number of agents did not move to other areas, but remained in $area_0$. From the aspect of temperature, $area_0$ is the most suitable area. However, $area_0$ had a food shortage problem. So, other behaviors were observed.

Migration between $area_0$ and $area_1$. We focus on Figures 3(e) and (f) in this subsection. We confirmed the agents moved to $area_1$ from $area_0$ by selecting the “migrate north” action between year 1983 day 260 and year 1984 day 10. Then, between year 1984 day 0 and year 1984 day 40, some agents selected and executed a “migrate south” action and went back to $area_0$. These agents migrated between $area_0$ and $area_1$ for only 40-80 days. We can easily assume

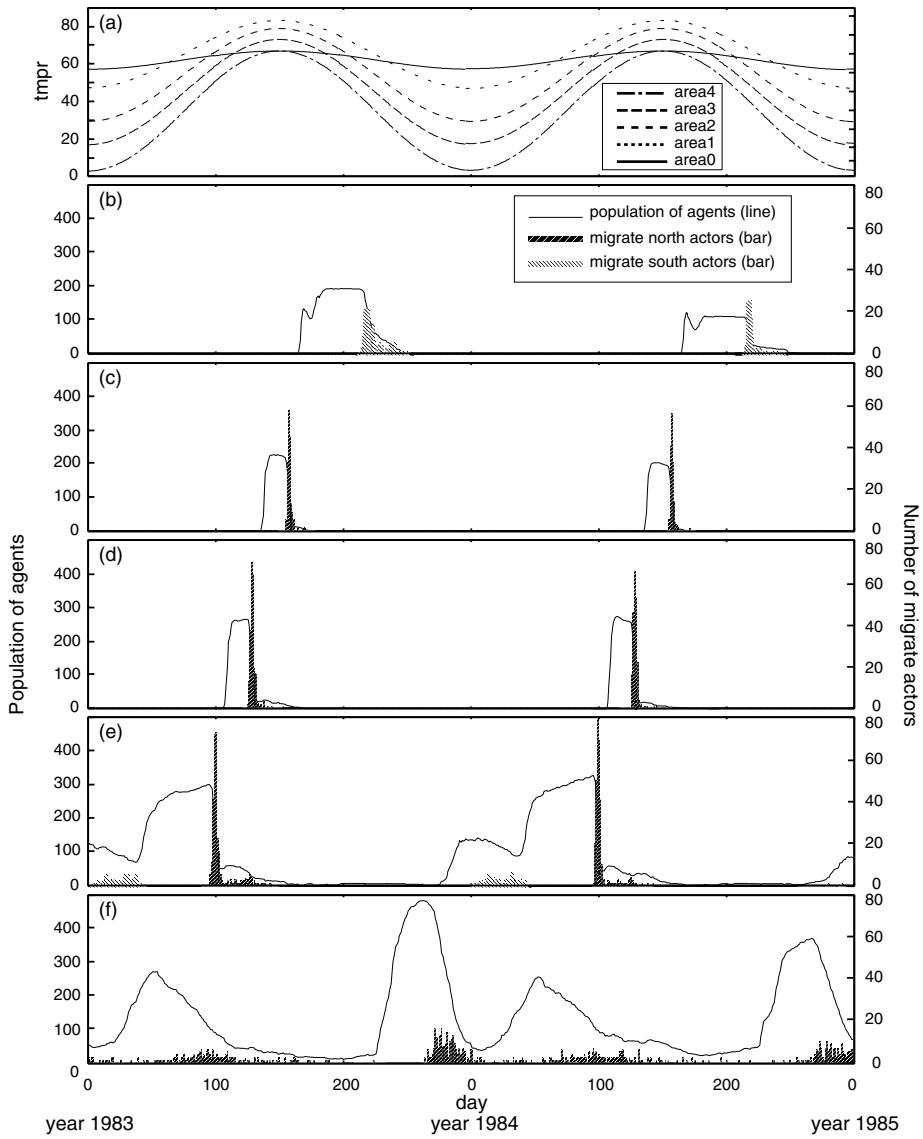


Fig. 3. (a) Temperature for 2 years. (b) Population of agents and number of migrate actors which behave “migrate north/south” in *area*₄. (c) is in *area*₃. (d) is in *area*₂. (e) is in *area*₁. (f) is in *area*₀.

one reason for the agents to behave like this, a food shortage problem in *area*₀. Around year 1983 day 270, *area*₀ held the biggest number of agents. This caused a food shortage. The number of plants in *area*₀ decreased to 9 (in this 2 year period, the maximum number of plants in *area*₀ was 93.). When *area*₀ reached this condition, the agents selected to go to *area*₁. Stated another way, the action

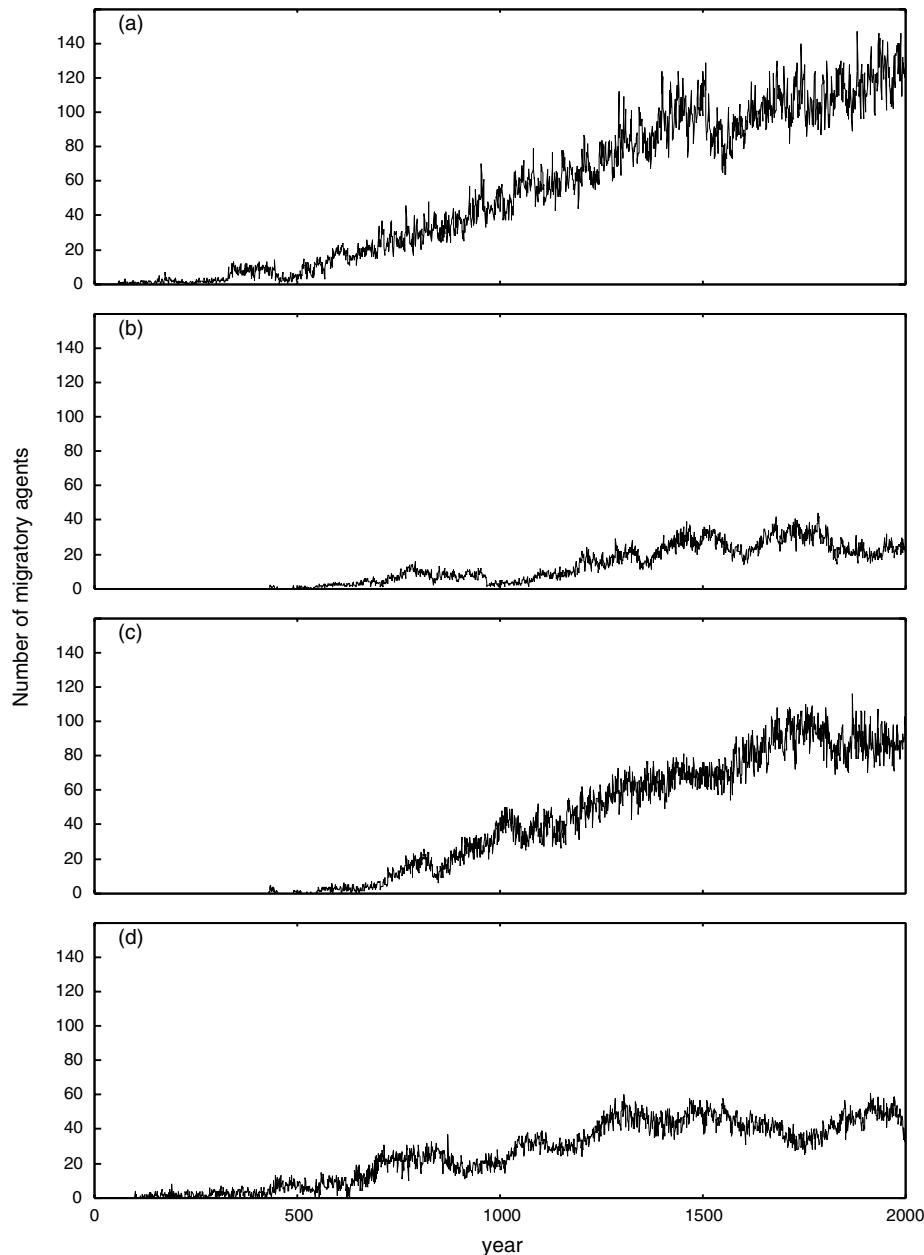


Fig. 4. (a) Number of migratory agents between $area_0$ and $area_1$. (b) is between $area_0$ and $area_2$, (c) is between $area_0$ and $area_3$, (d) is between $area_0$ and $area_4$. All points are average of 50 trials.

decision table of the agents evolved to selecting “migrate north” when the agent could not find any plants. However, it was not necessarily the best behavior from the aspect of temperature. The movement north around year 1983 day 260 - year 1984 day 10 meant that the agent left the most suitable area based on temperature. $area_1$ was a little colder at this time. Therefore, the agents soon returned to $area_0$.

Migration between $area_0$ and $area_4$. As in Figures 3(b) - (f), this migration behavior can be confirmed. First, we talk about the northward migration. A small number of the agents moved to $area_1$ from $area_0$ between year 1983 day 260 and year 1984 day 10. The number of agents in $area_1$ increased and around year 1984 day 100 moved to $area_2$. Moving to $area_3$ from $area_2$ was around year 1984 day 130, and moving to $area_4$ from $area_3$ was around year 1984 day 160. The agents stayed in $area_4$ for 60 days and then started back to $area_0$. This was around year 1984 day 220. As stated above, the agents established a migration behavior between $area_0$ and $area_4$. The reason why the agents migrated over the areas is attributed to the agents’ adaptation to both short-term change and a food shortage.

We have other results that helped us determine when the migration began and how many agents migrated. Before showing these results we will define the four migration behaviors that our ecosystem can achieve: migration between $area_0$ and $area_1$, migration between $area_0$ and $area_2$, migration between $area_0$ and $area_3$, and migration between $area_0$ and $area_4$. Figure 4 presents the results. It shows the development of migratory agents for a 2000 year period. Each migratory agent increased year by year. At the end of the experiment, all migration behaviors became apparent. This phenomenon is quite similar to the Monarch butterfly’s migration in the real world. We found from this result that short-term change and food shortage are possible reasons for the emergence of the migration behaviors of the Monarch butterfly.

4 Conclusion

We provided our definition of five areas, plants and agents, and the simulation of the emergence of the Monarch butterfly’s migration behavior. We used real temperature data from Central and North America as an environmental parameter. The results of the experiment showed that we succeeded in determining the agent’s migration behavior. We found from the second results that the agents established many kinds of migration behaviors. As future work we propose enhancing the agent’s functions. We want to determine the hibernation behavior of the Monarch butterfly.

Acknowledgments

This work was supported in part by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research under grant #17074003 and #20700199, and by the Tatematsu Foundation.

References

1. Todd, P., Wilson, S.: Environment structure and adaptive behavior from the ground up. In: From Animals to Animats 2: Proceedings of the 2nd International Conference on Simulation of Adaptive Behavior, pp. 11–20 (1993)
2. Langton, C.G.: Studying artificial life with cellular automata. *Physica. D* 2, 135–149 (1986)
3. Koza, J.R., Roughgarden, J., Rice, J.P.: Evolution of food-foraging strategies for the caribbean anolis lizard using genetic programming. *Adaptive Behavior* 1(2), 171–199 (1992)
4. Oboshi, T., Kato, S., Mutoh, A., Itoh, H.: Collective or scattering: Evolving schooling behaviors to escape from predator. In: Proceedings of the 8th International Conference on the Simulation and Synthesis of Living Systems: Artificial Life VIII, pp. 386–389 (2003)
5. Toquenaga, Y., Kajitani, I., Hoshino, T.: Egrets of a feather flock together. In: Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems: Artificial Life IV, vol. 1(4), pp. 391–411 (1994)
6. Marco Remondino, A.C.: An evolutionary selection model based on a biological phenomenon: The periodical magicicadas. In: Nolfi, S., Baldassarre, G., Calabretta, R., Hallam, J.C.T., Marocco, D., Meyer, J.-A., Miglino, O., Parisi, D. (eds.) SAB 2006. LNCS (LNAI), vol. 4095, pp. 485–497. Springer, Heidelberg (2006)
7. Sawada, T., Mutoh, A., Kato, S., Itoh, H.: A model of biological differentiation in adaptogenesis to the environment. In: Proceedings of the 8th International Conference on the Simulation and Synthesis of Living Systems: Artificial Life VIII, pp. 93–96 (2002)
8. Hashizume, H., Mutoh, A., Kato, S., Itoh, H., Kunitachi, T.: Multi-agent simulations of adaptive behavior with temperature-sensing agents. In: IEEE SMC International Conference on Distributed Human-Machine Systems, pp. 109–114 (2008)
9. Alerstam, T., Hedenstrom, A., Akesson, S.: Long-distance migration: evolution and determinants. *Oikos* 103(2), 247–260 (2003)
10. Walker, T.J.: Butterfly migrations in florida: Seasonal patterns and long-term changes. *Environmental Entomology* 30(6), 1052–1060 (2001)
11. The University of Kansas Entomology Program: Monarch watch (2008), <http://monarchwatch.org/>
12. National Oceanic and Atmospheric Administration: National Weather Service (2008), <http://www.nws.noaa.gov/>

Constraint Relaxation Approach for Over-Constrained Agent Interaction

Mohd Fadzil Hassan¹ and Dave Robertson²

¹ Computer and Information Sciences Department,
Universiti Teknologi PETRONAS, Bandar Seri Iskandar
31750 Tronoh, Perak, Malaysia

mfadzil_hassan@petronas.com.my

² Center for Intelligent Systems and their Applications (CISA),
School of Informatics, University of Edinburgh, Scotland, UK
dr@inf.ed.ac.uk

Abstract. The interactions among agents in a multi-agent system for coordinating a distributed, problem solving task can be complex, as the distinct sub-problems of the individual agents are interdependent. A distributed protocol provides the necessary framework for specifying these interactions. In a model of interactions where the agents' social norms are expressed as the message passing behaviours associated with roles, the dependencies among agents can be specified as constraints. The constraints are associated with roles to be adopted by agents as dictated by the protocol. These constraints are commonly handled using a conventional constraint solving system that only allows two satisfactory states to be achieved – completely satisfied or failed. Agent interactions then become brittle as the occurrence of an over-constrained state can cause the interaction between agents to break prematurely, even though the interacting agents could, in principle, reach an agreement. Assuming that the agents are capable of relaxing their individual constraints to reach a common goal, the main issue addressed by this research work is how the agents could communicate and coordinate the constraint relaxation process. The interaction mechanism for this is obtained by reinterpreting a technique borrowed from the constraint satisfaction field (i.e. distributed partial Constraint Satisfaction Problem), deployed and computed at the protocol level.

Keywords: Over-constrained agent interaction, brittle agent protocol, Distributed Partial CSP and agent protocol.

1 Introduction

In a collaborative problem solving task, multiple agents are involved in a joint decision, given that the sub-problems handled by each individual agents are interdependent and overlapping [1]. Each agent brings its own private constraints to bear on the decision, yet the agents must come to an agreement. The intra-agent constraints of each agent may be varied in terms of constraint density. Since agents are distributed in different locations or in different processes, each agent only knows the partial problem associated with those constraints in which it has variables. A global solution then

consists of a complete set of the overlapping partial solution of each agent. The solution is an instantiation of all variables that satisfy the intra-agent and inter-agent constraints.

The means of communicating and coordinating the problem solving efforts given the distinct sub-problems of the agents can be provided through an interaction protocol, as abstractly described in figure 1.

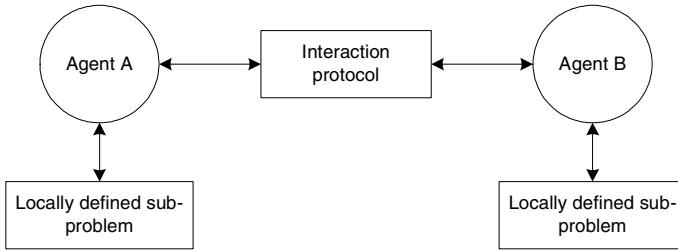


Fig. 1. Conceptual model of agent interaction in distributed problem solving

The interaction protocol provides roles that could be assumed by the interacting agents for reconciling their distinct sub-problems in finding mutually acceptable values for all variables of the problem to be jointly solved. The interactive states of the agents communicating through this protocol are dependent on the satisfiability of the constraints associated with the variables of the problem. The computation performed on an interaction protocol might involve the execution of the roles contained in the protocol across different machines or agents, therefore satisfaction of constraints by an agent associated with a particular role in an interaction protocol is done in ignorance of constraints imposed by other agents in the interaction. Hence, for a successful termination of the interaction protocol in coordinating the agents to achieve the intended objective of finding an agreeable solution, we require all constraints associated with the agents' roles to be solvable. For instance, the agents are in conflict if no compatibility is found between the corresponding variables values defined by the interacting agents. This conflict may lead to a failure in the reconciliation process, preventing the agents' progressions in their respective prescribed roles of the interaction protocol for achieving a solvable state. This inconsistent local view of interacting agents, which causes interaction failure, can be perceived as an over-constrained problem.

As such, interaction protocols are considered brittle, in a sense that the constraints imposed on the roles contained in the protocols must either succeed or fail, and if they fail the entire protocols may fail to achieve the objective of adequately resolving the interdependence among the agents' sub-problems. Consequently, protocol failure can cause the interaction between agents to break prematurely, even though the interacting agents could in principle reach an agreement. Given that the agents are capable of relaxing their individual constraints to accommodate the constraints of others in order to reach a common goal, the study presented in this paper is concerned with how existing approaches used to address over-constrained problems in the constraint satisfaction field can be integrated and adapted within a distributed interaction protocol framework to have a more flexible means of constraint

handling during agent interactions. For this purpose, we focus on the distributed partial Constraint Satisfaction Problem (CSP) scheme [2, 3].

2 Over-Constrained Problems and Distributed Partial CSPs

As described in [4], the use of constraint satisfaction techniques in multi-agent systems (MAS) is not new as they have been utilised either as a part of the agents' problem solving apparatus or coordination formalisms as reported in [5, 6]. However, in these works, the focus is strictly on the conventional formalisms of constraint satisfaction which require all constraints to be satisfied and do not address over-constrained problems. The few approaches that do attempt to integrate the currently available constraint satisfaction techniques for over-constrained problem with MAS include that of [7], which proposed a fuzzy constraint-based model for bilateral multi-issue negotiations in MAS. Our approach, on the other hand, considers integrating distributed partial CSP as part of the constraint handling feature of the distributed interaction protocol system. This is a novel way of providing a more flexible approach for handling constraints during the interactions of heterogeneous and autonomous agents participating in a distributed problem solving task.

A CSP consists of a finite number of variables, each having a finite and discrete set of possible values, and a set of constraints over these variables. A solution to a CSP is an instantiation of all variables for which all the constraints are satisfied. Though powerful, the CSP schema presents some limitations. In particular, all constraints are considered mandatory and need to be fully satisfied. However, in many real-world problems, it is often the case that there exists no consistent instantiation of variables that satisfies all constraints. This leads to unsolved problems. These problems are said to be over-constrained: any complete assignment of variables violates some defined constraint of the CSP [8]. In practice however, it is sometimes the case that certain constraints can be violated occasionally, or weakened to some degree. As conventional CSP techniques lack the mechanisms to accommodate such a notion of constraint handling, this gives rise to the establishment of a niche research area within the constraint satisfaction research field focusing on approaches to solve over-constrained problems, which include partial CSPs and distributed partial CSPs.

A partial CSP can be formally described as a triple [2]:

$\langle (P, U), (PS, \leq), (M, (\text{Necs}, \text{Suff})) \rangle$, where

- P is an original CSP, U is a set of ‘universes’, i.e., a set of potential values for each variable in P
- (PS, \leq) is a problem space, where PS is a set of CSPs (including P), and \leq is a partial order over PS
- M is a distance function over the problem space, and (Necs,Suff) are necessary and sufficient bounds on the distance between P and some solvable member of PS

A *solution* to a partial CSP is a soluble problem P' from the problem space and its solution, where the distance between P and P' is less than Necs. Any solution will suffice if the distance between P and P' is not more than Suff, and all search can

terminate when such a solution is found. An *optimal* solution to a partial CSP is a solution in which the distance between P and P' is minimal, and this minimal distance is called the optimal distance. The partial CSP scheme has been extended in [3, 9] for distributed environments and is known as distributed partial CSP.

A distributed partial CSP consists of:

- A set of agents (problem solvers), $1, 2, \dots, m$
- $\langle (P_i, U_i), (PS_i, \leq), M_i \rangle$ for each agent i
- $(G, (\text{Necs}, \text{Suff}))$, where

For each agent i , P_i is an original CSP (a part of an original distributed CSP), and U_i is a set of *universes*, i.e. a set of potential values for each variable in P_i . Furthermore, (PS_i, \leq) is called a *problem space*, where PS_i is a set of (relaxed) CSPs including P_i , and \leq is a partial order over PS_i . Also, M_i is a locally-defined *distance function* over the problem space. G is a *global distance function* over distributed problem spaces, and $(\text{Necs}, \text{Suff})$ are necessary and sufficient bounds on the global distance between an original distributed CSP (a set of P_i s of all agents) and some solvable distributed CSP (a set of solvable CSPs of all agents, each of which comes from PS_i).

A solution to a distributed partial CSP is a solvable distributed CSP and its solution, where the global distance between an original distributed CSP and the solvable distributed CSP is less than Necs . Any solution to a distributed partial CSP will suffice if the global distance between an original distributed CSP and the solvable distributed CSP is not more than Suff , and all search can terminate when such a solution is found.

3 Constraint Relaxation Approach

As described in [10], the coordination of a conflict resolution task among autonomous and heterogeneous agents requires the specification of the following two components. First, a protocol, or rules of interaction that coordinate the agents at an asocial level (i.e. synchronicity of messages) and social level (i.e. protocols that force the selection of a solution that satisfies some criteria). Second, the agent's strategy set, which can be specified as the preferred choices of the individual in how to i) generate solutions to the local/global problem and ii) how to evaluate proposals submitted by the other interacting agents in resolving the conflicts.

Within our proposed constraint relaxation approach, the protocol is derived from the interpretation of the distributed partial CSP scheme, which encapsulates both the asocial and social levels. It provides the interacting agents with the mechanism for constraint relaxation at both the intra-agent and inter-agent stages. At the intra-agent stage, it specifies the computational behaviour that can be assumed by the agents in determining the current state of the constraint relaxation process. At the inter-agent stage, the synchronisation of message-passing behaviour among agents is established. The design and working aspects of the approach are described in detail in the remainder of this section. The agent's strategy set is regarded as a 'black box', defined privately by each individual agent's designer, and is beyond the scope of this research.

Given a set of agents $X = \{1, \dots, n\}$, currently participating in the interaction to solve a distributed, problem solving task, then;

- For each agent $i \in X$, P_i is a solvable sub-problem defined by the agent concerning its part of the problem at the pre-interaction stage.
- As the problem is progressively solved by X , a set of variables, V of the problem is incrementally instantiated with mutually agreed set of solution values, as each agent $i \in X$ propagates its P_i that is part of the problem concerning V via an interaction protocol. The problem is said to be over-constrained if it consists of a set of variables, V , of which:
 - $V_S \subseteq V$, is a subset of variables that is fully solvable, in which all $i \in X$ agreed on the value assignments to V_S . That is, given $i \in X$, the value assignments to V_S is derivable from P_i . It is also possible for V_S to be empty, which means the agents cannot agree on the value assignments for any of the variables. In our work, this set of variables is specified in the necessary bound, Necs.
 - $V_F \subseteq V$, is a set of variables that is partially solvable, in which given $j \in X$, the value assignments to V_F is derivable from P_j , where agent j has already completed its part as prescribed in the interaction protocol concerning the solving of V_F . However, there is agent $k \in X$ that cannot complete its part in the interaction protocol to solve V_F , as its constraints as specified in P_k concerning V_F cannot be satisfied. In our work, this set of variables is specified in the sufficient bound, Suff.
- For each agent $i \in X$ involved in the constraint relaxation process, problem spaces, PS_i are made up of a number of possible weakened sub-problems, generated and provided by the agents during a particular constraint relaxation cycle, in which agents relax their original sub-problems (i.e. P_i) by applying constraint relaxation strategies privately held by the agents.
- For each agent $i \in X$, a solution subset distance metric, adapted from [3], is applied to compute the distance between each relaxed sub-problem, P'_i , selected from the problem space, PS_i , of agent i (i.e. $P'_i \in PS_i$), with its original, P_i , defined at the pre-interaction stage. Using this metric we identify N_i , the set of solutions not shared between the two problems, P_i and P'_i . N_i is derived by computing the union of the following two components; 1) a set of additional solutions introduced due to the selection of P'_i , and 2) a set of existing solutions of the original problem P_i , that is eliminated due to the selection of P'_i . The number of solutions identified by this union is computed as $d_i = |N_i|$, where d_i is the cardinality of N_i .
- The relaxation process involves agents $k, j \in X$ assuming their roles in relaxing their P_k and P_j respectively for attaining a solvable state. A solvable state of the problem is said to be achieved if any of the following is satisfied:
 - Agent k fully relaxes its original local sub-problem P_k , and produces a relaxed sub-problem, P'_k , which satisfies the necessary bound, $sols(P'_k) \supseteq sols(Necs)$. There exists at least a solution, N_k , from the set of

solutions derivable from P'_k , $N_k \in \text{sols}(P'_k)$, which is consistent with the existing solutions derivable from the original local sub-problem of agent j , $\text{sols}(P_j)$. That is, $N_k \cap \text{sols}(P_j)$. Attainment of this state indicates the satisfaction of sufficient bound, Suff. Alternatively, a similar result is achieved by agent j performing a constraint relaxation that meets the described requirements.

- o Both agents $k, j \in X$ partially relax their original sub-problems P_k and P_j respectively, and produce the respective relaxed sub-problems P'_k and P'_j . Both relaxed sub-problems satisfy the necessary bound, $\text{sols}(P'_k) \supseteq \text{sols}(\text{Necs})$ and $\text{sols}(P'_j) \supseteq \text{sols}(\text{Necs})$, and their combined constraint relaxations introduce new solutions N_k and N_j , where $N_k \cap N_j$. Attainment of this state indicates the satisfaction of sufficient bound, Suff.
- o If no combination of relaxed sub-problems, P'_k and P'_j , that produces a solvable state is found after an exhaustive search has been performed on the problem spaces, PS_k and PS_j , of the agents $k, j \in X$ respectively, then the constraint relaxation process involving the agents k and j is terminated. This indicates that the agents cannot reach an agreement in reconciling their differences.
- Obtaining a solvable state with the least number of constraint relaxations performed over agents $k, j \in X$ requires a search for the combinations of $P'_{k,j} \in PS_{k,j}$ which results in a solvable state to be achieved with a minimal $\sum(d_{k,j})$. Given that the search produces a number of equally ranked possible solutions, the solution with the minimal $\max(d_{k,j})$ is selected.

4 Encoding of Constraint Relaxation Approach as Agent Interaction Protocol

A number of existing approaches for agent interaction protocols include [11, 12], however as described in [13, 14], Lightweight Coordination Calculus (LCC) is considered more developed since it is readily available in an executable form and can be directly utilized for the work presented in this paper. This does not necessarily mean that our work is solely dependent on LCC. It is portable to any agent interaction protocol platform that has the same features as LCC.

LCC borrows the notion of role from agent systems that enforce social norms (i.e. Electronic Institutions [15]), but reinterprets this in a formalism based on process calculus. As LCC is a role-based language, it is necessary for our developed constraint relaxation approach described in detail in the previous section to be defined within the context of roles. As discussed in [16], there generally exist two distinct roles in any agent interaction protocol: that of *initiator* and that of *responder*. Both agents know when their portion of conversation is over because they had this notion of whether they initiated or responded to the conversation. For a smooth ongoing interaction between the agents participating in the constraint relaxation task, they are required to assume the designated roles as specified in the protocol. Each role in the interaction is modeled to encapsulate a set of conversation rules and behaviors applicable to the agents assuming the role. A role defines on how an agent in a given

state receives a message of specified type, performs local actions, sends out messages, and switches to another state. The descriptions on the intra-agent and inter-agent interactions between the agents' major roles are given in figure 2.

The agent faced with an over-constrained problem needs to assume the role of *initiator* to begin the constraint relaxation process. Contained within this initial role are three major roles namely *relaxation_initiation*, *relaxation_progression* and *relaxation_completion* that reflects the stages involved in the overall constraint relaxation process. These major roles are incrementally expanded in a sequential order as illustrated by the direction of the intra-agent arrows highlighted in figure 2. In the *relaxation_initiation* and *relaxation_progression* agent roles, we define the following two kinds of capabilities – message passing behaviours and constraint relaxation computations. For the message-passing behaviours, we allow inter-agent interactions concerning the sending and receiving of constraint relaxation related messages between the agents to be established, maintained and coordinated. This part is depicted as dashed arrows in figure 2. The constraint relaxation computations ensure that local actions like performing a solution subset distance given a relaxed and original sub-problems, searching for a solvable relaxed sub-problem with a minimal distance or revising the sufficient bound after the completion of a constraint relaxation cycle, are made available and accessible to the relevant agents. This allows the involved agents to effectively participate in the constraint relaxation process. Eventually, the *relaxation_completion* role, marks the end of the constraint relaxation task. It allows smooth termination of the protocol that guarantees a revised set of constrained variables is properly returned if a solvable relaxed state is achieved or a null value is returned if there exists none.

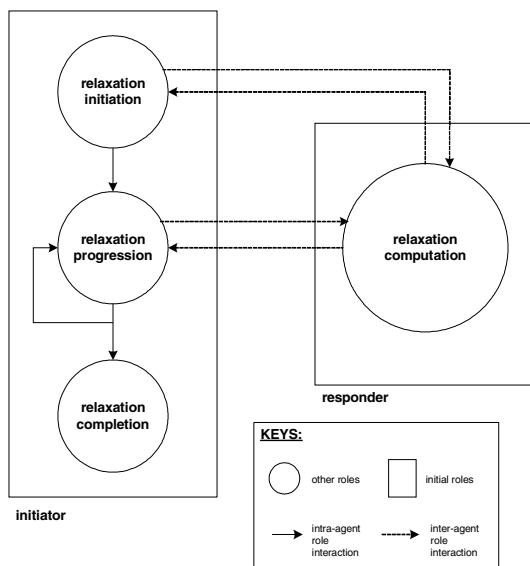


Fig. 2. Interaction between agent roles

An agent needs to assume the role of a *responder* to become the recipient of a request message to relax its part of the over-constrained problem. Upon receipt of the message, contained within the necessary and sufficient bounds, the *responder* assumes the *relaxation_computation* role. Within this role, the necessary computational process of finding a solvable relaxed sub-problem with a minimal distance given the original problem is performed. The inter-agent interactions between this role and the other roles of the *initiator* are illustrated as dashed arrows in figure 2.

5 Implementation and Evaluation

An important contribution of this work is not only developing the ideas of integrating distributed partial CSP with LCC, but also providing a practical and executable solution. In order to achieve this, our approach, which consists of inference procedures for performing constraint relaxation computations, needs to be implemented in a high level declarative language. In the LCC framework, the protocol language and the expansion engine are written in SICStus Prolog [17] and the message passing system is implemented in Linda [18]. Therefore, we choose to implement our approach in SICStus Prolog to take advantage of the existing code for the LCC basic framework and expansion engine, and ensure smooth interfacing with these components. In addition, a finite-domain constraint solver available in SICStus Prolog (i.e. clp(FD)) is used to accommodate the computations on the solution subset distance, necessary and sufficient bounds for the set of problems contained in the agents' problem spaces.

As the execution of the constraint relaxation protocol can be divided into a sequence of *cycles*, the time-based measurement is performed by analyzing the number of cycles taken by the agents to complete their respective parts in the protocol. A *cycle* is defined as one unit of protocol progress in which all agents, in their respective roles as specified in the constraint relaxation protocol, enacted the following three behaviors:

- i. Agents receive messages sent to them from the neighboring agents to whom the constraints on the over-constrained problem are shared;
- ii. Agents generate the necessary problem space contained within a set of relaxed sub-problem(s) and perform the necessary computation for finding a relaxed sub-problem with a minimal solution subset distance;
- iii. Agents send messages to the corresponding neighboring agents together with the solvable values the meet the distance specification, if there exist one.

For the experimental test bed, a set of over-constrained Multi-Agent Agreement Problems (MAPs) [19] with different levels of hardness are generated to be tested against the protocol. The results have shown that a harder problem generally requires a higher number of cycles for reaching a completion state.

6 Conclusion

The work reported in this paper has shown that our primary goal is fulfilled; to address the brittleness of protocol-led agent interaction for solving distributed problems. As the distinct sub-problems of the individual agents are interdependent, the existence

of an over-constrained state becomes the source of this brittleness. We have shown how a constraint relaxation approach can be adopted, by realizing the distributed partial CSP as an interaction protocol using the LCC. This allows heterogeneous agents, assumed to have the cognitive capability of relaxing their individual constraints, to take part in the interaction and coordination of distributed constraint relaxation process for obtaining a solvable state, if there exists one.

In addition, the research reported in the paper has bridged the gap between established works from two separate research disciplines; the constraint satisfaction and distributed protocol for multi-agent systems. It has shown on how we could utilize the available technique in one research field to solve the problem of another. It benefits both disciplines in the following two general aspects.

- i. For the constraint satisfaction research field, it makes the available techniques to address over-constrained problem relevant for the peer-to-peer agent environment.
- ii. For the multi-agent system research field, particularly the distributed agent protocol, it addresses the brittleness problem commonly faced by problem solving agents during their interactions for finding a solution.

Though this work is far from complete, it will pave a way for the integration of other available constraint satisfaction techniques based on fuzzy or probabilistic with the interaction protocol framework of MAS (e.g. LCC) to allow agents to have flexible interactions in solving distributed, constrained problem.

References

1. Decker, K.S., Durfee, E.H., Lesser, V.: Evaluating research in cooperative distributed problem solving, Computer Science Technical Report 88-99, University of Massachusetts at Amherst (1988)
2. Freuder, E.C., Wallace, R.J.: Partial constraint satisfaction. *Artificial Intelligence* 58, 21–70 (1992)
3. Yokoo, M.: Distributed constraint satisfaction: foundations of cooperation in multi-agent systems. Springer, Heidelberg (2001)
4. Sycara, K.P.: Multiagent systems. *AI Magazine* 19, 79–92 (1998)
5. Aldea, A., Lopez, B., Moreno, A., et al.: A multi-agent system for organ transplant coordination. In: Proceedings of the VIII European Conference on AI in Medicine, Carcias, Portugal (2001)
6. Macho-Gonzales, S., Torrens, M., Faltings, B.: A multi-agent recommender system for planning meetings. In: Proceedings of the Workshop on Agent-based Recommender Systems (WARS 2000), Barcelona, Spain (2000)
7. Luo, X., Jennings, N.R., Shadbolt, N., et al.: A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments. *Artificial Intelligence* 148, 53–102 (2003)
8. Meseguer, P., Bouhmala, N., Bouzoubaa, T., et al.: Current approaches for solving over-constrained problems. *Constraints* 8, 9–39 (2003)
9. Hirayama, K., Yokoo, M.: Distributed partial constraint satisfaction problem. In: Proceedings of the Third International Conference on Principles and Practice of Constraint Programming (CP 1997), Linz, Austria (1997)

10. Faratin, P., Klein, M.: Automated contract negotiation and execution as a system of constraints. In: Proceedings of the Workshop on Distributed Constraint Reasoning, at IJCAI 2001, Seattle, USA (2001)
11. de Silva, L.P.: Extending agents by transmitting protocols in open systems. RMIT University, Melbourne (2002)
12. Freire, J., Botelho, L.: Executing explicitly represented protocols. In: Proceedings of the Workshop on Challenges in Open Systems at AAMAS 2002, Bologna, Italy (2002)
13. McGinnis, J.P.: On the mutability of protocols, in CISA. University of Edinburgh, School of Informatics (2006)
14. Robertson, D.: A lightweight coordination calculus for agent social norms. In: Proceedings of the Declarative Agent Languages and Technologies at AAMAS 2004, New York, USA (2004)
15. Esteva, M., Padget, J., Sierra, C.: Formalizing a language for institutions and norms. In: Meyer, J.-J.C., Tambe, M. (eds.) ATAL 2001. LNCS (LNAI), vol. 2333, pp. 348–366. Springer, Heidelberg (2002)
16. Cabri, G., Ferrari, L., Zambonelli, F.: Role-based approaches for engineering interactions in large-scale multi-agent systems. In: Lucena, C., Garcia, A., Romanovsky, A., Castro, J., Alencar, P.S.C. (eds.) SELMAS 2003. LNCS, vol. 2940, pp. 243–263. Springer, Heidelberg (2004)
17. SICS, SICStus Prolog User's Manual. Stockholm: Swedish Institute of Computer Science (SICS) (1999), <http://www.sics.se/sicstus.html>
18. Carriero, N., Gelernter, D.: Linda in context. Communications of the ACM 32, 444–458 (1989)
19. Modi, P.J., Veloso, M.: Bumping strategies for the multiagent agreement problem. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), Utrecht, The Netherlands (2005)

Structure Extraction from Presentation Slide Information

Tessai Hayama¹, Hidetsugu Nanba², and Susumu Kunifugi¹

¹ Graduate School of Knowledge Science
Japan Advanced Institute of Science and Technology

1-1, Nomishi, Ishikawa, Japan

{t-hayama, kuni}@jaist.ac.jp

² Faculty of Information Sciences
Hiroshima City University

3-4-1, Ozukahigashi, Asaminamiku, Hiroshima, Japan
nanba@its.hiroshima-cu.ac.jp

Abstract. Electronic presentations are used in numerous scenarios, such as lectures and meetings. In recent years, the widespread use of electronic presentations means that presentation slide data is increasing as one of industry's most important information resources. Therefore, it is necessary to develop a practical usage method for the reutilisation of the data on slides. An approach to achieve this is to focus on visual structure information within a slide, because visual structure information is one of the most valuable, easy to understand methods for humans. However, since visual structure information is not explicitly defined in the slide data itself, computers have difficulty comprehending structure information directly. In this paper, we propose a method of extracting structure information from slide information. The proposed method is composed of two steps: organising objects within the slide as units, such as title, body text, figure and table, and structuring the units as a hierarchy tree based on a top-down approach.

Keywords: Information Extraction, Presentation Slide, Visual Layout, Web Data.

1 Introduction

The widespread use of electronic presentations is increasing the number of slides that businesses accumulate. Since used slides are often stored and reused as e-Learning or Web content, the data stored on slides is rapidly becoming one of industry's most important information resources. Therefore, it is necessary to develop a practical usage method for the reutilisation of the data stored on slides. One approach to preparing a slide data reutilisation system is to use visual structure information within a slide. Most systems currently handling slide data convert the slide data into simplified text data, and then users access the data by using a linear sequence of words. Although a slide's visual structure information, such as visual form and layout, is valuable to easily understanding the context of

the data, current slide data reutilisation systems always ignore such information. If a system could handle the text data with its structural information, then the system would be able to facilitate the intelligent processing of slide data. Structure information representing the relationships among the data entities is not explicitly defined in the slide however. Therefore, it is not easy to manually add the definition to all existing slide data. Thus, we need to develop technology to automatically extract structure information from the slide data.

Several methods for extracting structure information from documents have been proposed[8][7][1]. Rosenfeld et al.[6] and Zhai et al.[9] suggested a structure extraction method for PDF and Web documents using probabilistic approaches, such as the machine-learning and tree-graph-matching algorithms, respectively. These approaches need to prepare a large amount of annotated data, and the models made from the data are dependent on the data. Although it is useful to adapt target data containing a few types of information structure, it is difficult to adapt target data containing various types of information structure, such as on slides. Nanno et al.[5] proposed a method of extracting structure information from Web pages using the repetition of elements within the Web page. However, the method is inapplicable for slide structure extraction because slide data does not include an explicit regular element, such as an HTML tag. Ishihara et al.[3] proposed an extraction method based on close distances among the objects within a slide to analyze structure information focusing on diagrams within a slide. Because the objects within the slide can be created freely and then manually allocated on each slide, slides sometimes include objects in incorrectly overlapped positions. Thus, Ishihara et al.'s method cannot analyze the structure information appropriately. Although the previous methods can effectively extract structure information from a document with formal formatting, they cannot extract structure information from the information on a slide that has a varied layout structure and incorrectly placed objects. In this paper, we propose a method of extracting structure information from the information on slides. The proposed method is composed of two steps: organising primitive objects within the slide as units, such as title, body text, figure and table, and structuring the units as a hierarchy tree based on a top-down approach. Knowledge of slide structure is useful in various applications. For example, the knowledge of structure represents the visual structure that the current slide readers cannot utilise, so that it allows blind users to understand presentation documents more easily. In addition, a system that presents slides on hand-held devices with small display screens can use the structure for segmentation and assign a human-like layout to the segmented slide information.

2 Definition and Problems for Structure Extraction

2.1 Slide Information and Its Structure

Slide information is composed of one or more primitive objects, such as texts, pictures, lines and basic diagrams. Each object is recognised as a functional attribute, such as title, body text, figure, table and decoration. For example, as

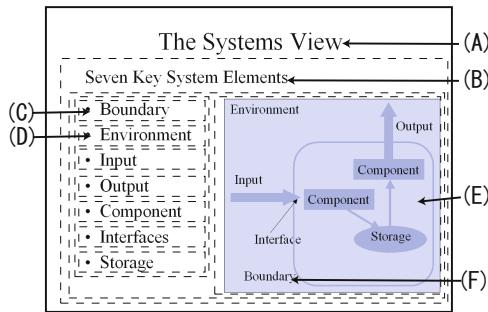


Fig. 1. An example of slide information and its structure. (Object(s) boxed by a dashed line represent a unit able to function as an attribute.)

shown in Fig. 1, (A), (C) and (F) are primitive text-type objects and have the functional attributes of title, body text and figure, respectively. In such a case, (F) is recognised with (E) as a unit of figure attribute. Thus, even if the objects are similar in type, as in the example, they may function either as different attributes or as a group of two or more objects, but not as a single object.

The structure of information within a slide can be represented as a hierarchy tree of the units in which the object(s) can function as an attribute: a set of similar units. To detect the relationships between the units, cues such as the slide's visual form and layout can be used. For example, as shown in Fig. 1, object (A) with a title attribute can be assigned to the root node. Objects ((B) and (C)), related by indent, and objects ((C) and (D)), itemised in same level, can be assigned to the parent-child nodes and sibling nodes, respectively. In addition, a text box can be represented by a partial tree composing the objects within the box. Thus, the visual cues within a slide provide the relationships between the units to detect the tree structure.

2.2 Extracting Structure Information from Slide Information

As described in the previous section, structure information within a slide is built on primitive objects by the following steps: (1) Organising primitive objects within a slide into units that can function as an attribute and (2) structuring the units as a hierarchy tree.

In step 1, to detect the units that are able to have an attribute, the close distance and the overlap between the objects can be used as described in [3]. However, since the objects are freely created and then manually allocated on the slide, incorrect object placement is inevitable and thus may fail to lead to the detection of the separations between the units. To detect the separations between the units, not only the distance relationships but also the functional relationships between the objects can be used. For example, if text- and diagram-type objects overlap, the text-type object can be properly identified as a body-text attribute using a bullet point list. Therefore, inappropriate organisation of the slide can be eliminated.

In step 2, the structuring of the visual layout is often used as an approach based on the regularity of the unit relationships as well as the matching of template layouts. Since it is difficult to prepare a large collection of layout templates as in the latter approach, we will apply the former approach to structure the units. Although it is useful to use the visual cues, such as indents and bullet point lists, to detect the regularity of the relationship of the units, it is inappropriate to use them alone. For instance, if a figure object is allocated in a large area of the slide, the regularity with the visual layout might be disturbed. To compensate the regular disturbance, the system can also use attribute information with the units. Even if the figure object disturbs the layout regularity, we can maintain layout regularity using the regularity of units of the object.

3 Proposed Method

We propose a method of extracting structure information from the information within a slide. The method consists of the following stages. First, organising primitive objects within a slide into units using function relationships among the objects. Second, structuring the units based on a top-down approach.

3.1 Organising Information in a Slide into Units Using Functional Relationships

To identify an uncertain attribute within an object, we assumed that the attribute could be determined by the functional relationships between that object and the other objects on the slide with more certain attributes. Therefore, this proposed method assigns the likely attribute within each object, and then determines the attribute of the object in order of the object with the most obvious attribute, which also affects the determination of the attribute(s) of the functionally related object(s). The organisation can be achieved with the following procedures.

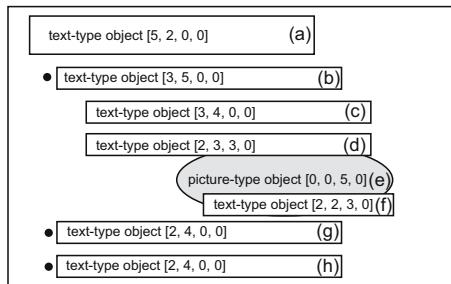
1) Assigning the likelihood of each attribute within each object: The score of each attribute within each object is assigned using a score sheet, as shown in Table 1. The score sheet is made based on the type, position and size of an object with the distinction of each attribute. The points within the sheet are scored according to the following rules: an object with the properties that indicate the likelihood of each attribute is given points for each attribute. If the type of the object influences the scoring for each attribute to a greater degree than the object's position and size, an object functionally related as the likelihood of each attribute is given points for each attribute.

The score is calculated by the number of items that are matched with the object. For example, as shown in Fig. 2, object (a) has attribute scores of title, body text, figure and table as 5, 2, 0 and 0, respectively. In addition to the assignment, the functional relationship of each attribute within each object is listed if the scoring of each attribute is used as a relation to the other object(s), such as the items underlined in the score sheet.

Table 1. Score sheet of attribute based on the likelihood of the attributes

Items for title attribute		Items for body-text attribute	
With font size $> \text{Threshold}_{(\text{fontsize1})}$	+1	With bullet symbol	+1
With position from the top $> \text{Threshold}_{(y\text{-axis_position})}$	+1	Existing text-type object(s) with similar format and at the same left-position.	+1
With the nearest top-position in the slide	+1	Existing the other text type object on the position of upper left/lower right of it	+1
With the largest font size in the slide	+1	With font size $> \text{Threshold}_{(\text{fontsize2})}$	+1
With number of characters $> \text{Threshold}_{(\text{number_of_characters})}$	+1	With number of characters $> \text{Threshold}_{(\text{number_of_characters})}$	+1
Items for figure attribute		Items for table attribute	
Graph/Picture object	5	With number of data more than half of cells within table	5
Complete overlapping G/P_Obj	4	With number of data less than half of cells within table	4
Partially overlapping G/P_Obj	4	Complete overlapping cell area within table	4
Overlapping G/P_Obj indirectly	3	Partially overlapping cell area within table	4
Text-type object at top/down position among a group overlapped G/P_Obj directly/indirectly	-1	Overlapping cell area within table	3
Diagram object with no text	4	indirectly	
With number of characters $< \text{Threshold}_{(\text{number_of_characters})}$	+1		

$\text{Threshold}_{(\text{fontsize1})}$, $\text{Threshold}_{(\text{fontsize2})}$, $\text{Threshold}_{(Y\text{axis_position})}$ and $\text{Threshold}_{(\text{number_of_characters})}$ are represented parameters of font size, font size, distance from the top-position and number of characters, respectively, G/P_Obj and underlined items indicate graph/picture object and the scoring using relationships among other object(s), respectively.

**Fig. 2.** An example of a slide including attributes scores (The numbers within square brackets indicate the attribute scores of title, body text, figure and table.)

(2) Identifying the attributes of the objects: By detecting an object with a maximum likelihood of an attribute, the attribute of the object is determined and then the other object(s) functionally related to this other object is affected. The process consists of the following three steps: 2.1, 2.2 and 2.3.

- **(2.1) Detecting an object with maximum likelihood of an attribute among the objects with non-determined attributes:** First, each object with a non-determined attribute is set as a candidate attribute (*attri_candidate*), which is one with the largest scores among the four attributes. The likelihood of the candidate attribute contains not only the likelihood degree of the candidate attribute but also the unlikelihood degree of the other attributes. Thus, the likelihood of the candidate attribute is defined and its value (*Li_Attri*) is given by equations (1) and (2). Here, *attri*, *Attri_Val*_(*attri*) and *MaxScore*_(*attri*) indicate an attribute, its scores assigned in step 1 and the max score for each attribute¹, respectively.

$$Ev_{(attri)} = \begin{cases} Attrи_Val_{(attri)} & (\text{if } attri_candidate == attri) \\ MaxScore_{(attri)} - Attrи_Val_{(attri)} & (\text{otherwise}) \end{cases} \quad (1)$$

$$Li_Attri = Ev_{('title')} * Ev_{('body-text')} * Ev_{('figure')} * Ev_{('table')}. \quad (2)$$

For example, as shown in Fig. 2, the likelihood attribute values of objects (b) and (g) are set by 375 and 300, respectively, so that (b) is more likely to be identified with the body-text attribute than (g). Finally, an object with the largest attribute value is detected and determined with the candidate attribute.

- **(2.2) Changing each attribute score with each object(s) related to the object, which is determined in step 2.1:** The rules are as follows:
 - If the object is identified as a title attribute, each object, including the relationship lists but excluding title, is set by each attribute score, which is then subtracted by 1. Also, the title attributes of all objects are set by 0.
 - If the object is identified as an attribute, excluding title, each object included in the relationships lists of the attribute is set by the attribute score, which is subtracted by 1.
- **(2.3) Repeating steps 2.1 and 2.2 until the attributes of all objects are determined:** An object with a more certain attribute is preferentially assigned the attribute and can affect the determination of an attribute within the functionally related object(s).

(3) Organising the objects into units using close distance and object overlap: After determining the attributes of all objects within the slide, the objects with figure attributes are organised based on the objects' figure relationships list.

(4) Assigning a decoration attribute: After the units with an attribute are detected, a diagram-type object including unit(s) with a body-text attribute and a non-organised arrow-shape type object are reassigned as decoration attributes.

¹ In the score sheet of Table 1, the max score for each attribute is 5.

3.2 Structuring the Units Based on a Top-Down Approach

By detecting regularity in a slide's visual structure, the system builds the units as a tree structure. The structuring is based on a top-down regional dividing approach; step by step, the block region including the units is divided into more blocks so that every additional dividing step creates a hierarchical structure by defining parent-child relationships between the units.

The procedures of the structuring are as follows:

(1) Setting initial state: The initial block and a root node are set based on the unit with the title attribute. If a unit with a title attribute is included, the unit is assigned to the root node and the initial block is created to contain the region, including the units below the unit with the title attribute. Otherwise, the root node is created as a blank, and the block is created by the region that includes all the units.

(2) Dividing block(s) vertically: If block(s) have vertical blank space(s), then these block(s) are divided into more blocks according to each space. Otherwise, this step is skipped.

(3) Dividing block(s) horizontally: The step consists of four stages: In the first stage, large horizontal blank space(s) are sought from block(s). If horizontal blank space(s) larger than the threshold is found in the block(s), then each block is divided into more blocks according to the space. Then, the system proceeds to step 4.

In the second stage, the units' attribute sequences in the block(s) are checked. If one of the sequences in each block is matched with the following heuristic rules, which are based on the attribute relationship between a body text and a figure/table, the block is divided into two blocks according to the matched rule. Then, the system goes to step 4.

- (i) If 'an attribute within *TopObj*' == 'body-text' and an object with a figure attribute to the left from *TopObj* is included, then the block is divided into two blocks by the top position of the object with a figure attribute. Only a bullet point list with the *TopObj* is not included in the object with the figure attribute.
- (ii) If 'an attribute within *TopObj*' == 'figure/table', then the block is divided into two blocks by the bottom position of the object at the top of the block.

where *TopObj* presents an object at the top of the block.

In the third stage, the unit at the top position of each block is checked. If the unit is identified as a body-text attribute and is included in a bullet point list, then the block is divided into more blocks by the top position of each of the bullet points and then goes to step 4.

In the final stage, the unit at the top position of each block is checked. If the unit is identified as a body-text attribute, then the block is divided into the unit and other units within the block.

(4) Repeating steps 2 and 3 until every block includes one unit.

4 Experimental Evaluation

4.1 Method and Preparation

The experiment we conducted mainly focused on two points: (1) whether it was effective for the organising process to apply functional relationships among primitive objects within a slide, and (2) whether it was appropriate for the structuring process to apply a top-down approach, which provides attribute information within the units. At present, no structuring methods of slide information and standardised data sets are available for evaluation. Therefore, to evaluate this proposed method, it is necessary to prepare a comparable method.

To compare the proposed method, we used two methods as follows. The first method is set as a standard method that is organised by the objects based only on information derived from the objects' distance relationships. For example, an object with a figure/table type and object(s) overlapping or closely allocated with the object are organised into a unit with a figure/table attribute. The second method is set as a proposal method without the functional relationship ('*Func_rel*'). For instance, an attribute within each object is assigned as an attribute with the highest score in step 2.1 of the organisation, and steps 2.2 and 2.3 are cancelled. In addition, we examined the accuracy of the structuring process using each unit data, which was produced in the organising process. To compare these two methods, we used recall, precision and F-measure. These performance measures are calculated with the following formulas.

$$\text{Recall} = \frac{\text{number of detected units matched with correct}}{\text{total number of correct}} \quad (3)$$

$$\text{Precision} = \frac{\text{number of detected units matched with correct}}{\text{total number of detected units}} \quad (4)$$

$$F - \text{measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad (5)$$

We created a slide data set for the evaluation. The data set included 30 slides randomly selected from a research paper database[4], which included papers and more than 10,000 slides from the Internet. The average number of pages in slides was 16.4. The structure data of these slides was created manually by manipulating a specially developed system interface. We implemented a system that could automatically generate the structure of information in a slide from the slide file. The system was developed in Microsoft Visual Studio C#, which accepts its input data from a Microsoft PowerPoint (PPT) file and outputs an XML file containing structure information.

4.2 Results and Discussion

The results of the organising and structuring processes are summarised in Tables 2 and 3, respectively. Table 2 shows that the proposed method can organise slide information into units with an attribute better than other methods. In particular, this method can also detect a unit as a figure attribute. The organisation of

Table 2. Accuracy for each attribute results in the organising process

Attribute (Number of its unit)		Title (602)	Body text (2267)	Figure (430)	Table (12)	Decoration (393)
Proposed Method	Recall	0.98	0.98	0.90	1.00	0.90
	Precision	1.00	0.91	0.83	1.00	0.62
	F-measure	0.99	0.94	0.86	1.00	0.73
Proposed Method without using <i>Func_Rel</i>	Recall	0.97	0.94	0.87	1.00	0.86
	Precision	0.98	0.92	0.80	1.00	0.63
	F-measure	0.98	0.93	0.84	1.00	0.73
Standard Method	Recall	0.93	0.84	0.57	1.00	0.86
	Precision	0.98	0.90	0.59	1.00	0.63
	F-measure	0.95	0.87	0.58	1.00	0.73

Table 3. Ratio in pages for each correct ratio of results in the structuring process

Ratio of correct link number within a page	1.00	0.99-0.80	0.79-0.60	0.59-0.00	N/A
Proposed Method	0.75	0.05	0.07	0.10	0.04
Proposed Method without using <i>Func_Rel</i>	0.62	0.06	0.08	0.20	0.04
Standard Method	0.60	0.04	0.05	0.28	0.04

units with a figure attribute is susceptible to the effect of the distance between the objects. Thus, the proposed method providing the functional relationship can eliminate the incorrect placement of slide objects. Table 3 shows that the proposed method is able to identify an attribute more correctly and also that the proposed method can structure the information within a slide better than any other methods. The proposed method used attribute information within units in the structuring process to compensate for the regularity of the visual layout. If units and their attributes were identified more correctly, the proposed method would be able to function more effectively. Thus, attribute information with the units is important for extracting slide structure information; therefore, our approach is useful for extracting information.

We also checked the errors caused by the proposed method in the experiment. One of the problems is that the relationships between the slide's objects are defined by text content, not by the slide's visual layout. This makes it necessary, therefore, to apply a text analysis technique to detect the relationships among the objects.

5 Conclusion and Future Work

In this paper, we proposed a technique, involving organizing and structuring processes, to extract structure information from the information within a slide. The organising process used functional relationships between the objects, and not only the information derived from the close distances between the objects, to eliminate potentially inappropriate organising. In the structuring process,

attribute information within the units, as well as the visual cues on the slide, was used to detect how the regularity of the layout structure could be improved.

Although our current system still needs some modifications, our experimental result shows that the proposed method can extract structure information from slide information. In our future work, we are planning to develop slide applications using the structure data extracted by this technique and the slide information processing technique[2] that we have developed.

References

1. Anjewierden, A.: AIDAS: Incremental Logical Structure Discovery in PDF Documents. In: Procs. the 6th International Conference on Document Analysis and Recognition, pp. 374–378 (2001)
2. Hayama, T., Nanba, H., Kunifushi, S.: Alignment between a Technical Paper and Presentation Sheets Using a Hidden Markov Model. In: Proc. Active Media Technology 2005, pp. 102–106 (2005)
3. Ishihara, T., Takagi, H., Itoh, T., Asakawa, C.: Analyzing Visual Layout for a Non-Visual Presentation-Document Interface. In: Proc. the 8th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 165–172 (2006)
4. Nanba, H., Abekawa, T., Okumura, M., Saito, S.: Bilingual presri: Integration of multiple research paper databases. In: Proc. the 7th RIAO Conference: Coupling approaches, coupling media and coupling languages for information retrieval, pp. 195–211 (2004)
5. Nanno, T., Saito, S., Okumura, M.: Structuring Web Pages Based on Repetition of Elements. In: Proc. the 2nd International Workshop on Web Document Analysis, pp. 58–60 (2003)
6. Rosenfeld, B., Feldman, R., Aumann, Y.: Structural extraction from visual layout of documents. In: Procs. the 11th International Conference on Information and Knowledge Management, pp. 203–210 (2002)
7. Watanabe, T., Luo, Q., Sugie, N.: Layout Recognition of Multi-Kinds of Table-Form Documents, IEEE Transactions on Pattern Analysis and Machine Intelligence 17(4), 432–445 (1995)
8. Yang, Y., Zhang, H.: HTML Page Analysis Based on Visual Cues. In: Procs. the 6th International Conference on Document Analysis and Recognition, pp. 859–864, 10–13 (2001)
9. Zhai, Y., Liu, B.: Structured Data Extraction from the Web Based on Partial Tree Alignment. IEEE Transactions on Knowledge and Data Engineering 18(12), 1614–1628 (2006)

Combining Local and Global Resources for Constructing an Error-Minimized Opinion Word Dictionary

Linh Hoang, Jung-Tae Lee, Young-In Song, and Hae-Chang Rim

Dept. of Computer and Radio Communications Engineering
Korea University, Seoul, Korea
`{linh,jtle, song, rim}@nlp.korea.ac.kr`

Abstract. A lexical dictionary consisting of opinion words and their polar orientations plays a crucial contribution to opinion mining tasks (e.g., sentiment classification). Previous works on automatic construction of such dictionary have a problem of generating errors (i.e., incorrect identification of polar orientations of words in dictionary). To address the problem, this paper proposes an Error Minimization Algorithm for reducing errors caused by automatic compiling process to construct a reasonable opinion word dictionary. The proposed algorithm combines global and local resources for extracting and refining the dictionary with minimum errors. Empirical results show that our proposed approach is effective for enhancing the performance of the sentiment classification task.

1 Introduction

Over the past few years, opinion mining has been receiving increasing research attentions not only because of itself being a new topic but also due to the effects it would bring on real world business. Web services today provide users an easy and accessible way to express their opinion about almost everything. Since the number of user-generated documents is increasing, opinion mining has practical meaning to exploit such kinds of valuable information.

In the fields of opinion mining, one of the most early and challenging tasks addressed so far includes sentiment classification. Dealing with this task, many researches preferred lexicon-based approaches to identify the polar orientation (*positive|negative*) expressed in a given text [1,2,3,5,6,8,12]. In these approaches, a dictionary consisting of opinion words is crucial.

To date, there have been remarkable efforts to compile a list of opinion words as the initial step for sentiment classification. Early studies [4,12,14] tried to find opinion words based on constraints or associations of patterns in large corpora (typically the Web). However, these approaches sometimes skip opinion words that do not occur frequently in the corpus. Most recent works [2,5,7,8] mined an available thesaurus (usually the WordNet) to acquire opinion words with their polar orientations. Starting with a small amount of annotated seed words, these approaches look for synonyms and/or antonyms of seeds, identify their polar

Table 1. Examples of errors generation

<i>fantastic</i> (positive) → <i>unreal</i> → <i>fake</i> (positive)
<i>violence</i> (negative) → <i>upheaval</i> → <i>excitement</i> → <i>joy</i> (negative)

orientation according to the orientation of associated seed word, add new found words to the seed list, and iterate this bootstrapping process until convergence.

While analyzing the latter approaches, we found the bootstrapping process may generate errors of which the polar orientation of a word is incorrectly identified. Let's consider the aforementioned process when we grow a word list starting from two seeds “*violence*” (negative orientation) and “*fantastic*” (positive orientation). After several iterations, errors may be generated, as shown on Table 1. When opinion words are iteratively extracted, number of errors are increased and may consequently harm the performance of final sentiment classification.

The shortcoming of previous studies suggests us to combine local and global resources (here are the WordNet and the Web respectively) for constructing an opinion word dictionary (OWD). Our proposed method consists of the following two separate stages: extracting opinion words and refining their polar orientation. At the extracting stage, we mine the WordNet’s *synsets* (sets of synonyms) to initially expand a list of words by using rather similar method to [5,8]. At the refining stage, we use an Error Minimization Algorithm to reduce errors generated from the extracting stage. This algorithm corresponds to our intuition to refine a reasonable OWD which can be a contributive source for sentiment classification task afterwards.

The rest of this paper is organized as follows. Some related works and the proposed method are presented in section 2 and 3 respectively. In section 4, the experiments for verifying the proposed approach are described in details. Section 5 discusses about empirical results with our justified hypothesis. Section 6 summarizes the paper and suggests several potential works in the future.

2 Related Work

So far, there have been three different ways to construct OWD. The easiest way is to select opinion words and decide their polar orientation manually. Intuitively, this approach requires much time, effort and experience of the annotators. Due to this reason, manual approach is no longer used as a preferred option for building OWD.

The second way, namely corpus-based approach, is to use syntactic or co-occurrence information of patterns in large corpora to mine opinion words. Based on the assumption that conjoined adjectives often have the same orientation, [4] used constraints on semantic orientation to classify adjective into *positive* or *negative*. The idea of using connectives or constraints in constructing OWD is recently utilized in [2,6,13]. Coming up with this approach, [12] might be a

conceptually simplest work for mining orientation of consecutive phrases on the Web corpora; it employed *pointwise mutual information* to estimate the semantic orientation for every extracted phrase co-occurred with two polar seed words, “*excellent*” and “*poor*”. The coverage is the cost of corpus-based approaches since they can hardly be applied to infrequent words that can be possible indicators of sentiment. Opinion words that rarely occur in the corpus can be skipped or miss-classified when frequency information is used to decide their semantic orientation.

The third way called dictionary-based approach often starts with a small set of seed words and expands it by iteratively searching for synonyms and/or antonyms in the thesaurus. Based on the relationships with corresponding seed words, new words are identified and added to the seed set after each iteration. This bootstrapping process ends when either the expanded set is big enough or there is no more new words to be added. Most of the dictionary-based works [2,5,7,8] used the WordNet as the thesaurus resource. As we mentioned earlier, the main shortcoming of these approaches is the errors caused in the iterative process. A manual inspection can be applied afterwards to filter out errors, but this requires much human effort.

3 Proposed Method

Our approach for constructing OWD consists of two stages. In the first stage, we extract opinion words from a local thesaurus (WordNet) to build an initial dictionary. In the second stage, we utilize both semantic information from WordNet and statistical information from global corpora (Web) to re-determine the orientation of the extracted words. The second stage of refining OWD is the main difference between our approach and previous ones.

3.1 Extracting Opinion Words

At the extracting stage, we obtain opinion words by looking for synonyms of seed words in the WordNet.¹ Our method of expanding OWD is similar to [5,8] except that we heuristically select our own seed set and only look for synonyms. Our starting seed list consists of 26 positive and 22 negative words within 4 part-of-speech (POS) categories (*noun*, *verb*, *adjective*, *adverb*) (See Table 2 for examples of seed words). Consequently, we only look for synonyms that have the same POS category with those of original seed words. After each iteration, all synonyms of *positive* or *negative* seed words are added to *positive* or *negative* seed list respectively. Duplicate words appeared in both of two seed lists are automatically removed. In our case, the iterative process finally produced 6,138 positive words and 5,647 negative ones.

¹ WordNet3.0 - <http://wordnet.princeton.edu/obtain>.

Table 2. Examples of seed words

Positive seed words	Negative seed words
great, perfect, excellent, fun, beauty, enjoy, love, nicely, wisely, ...	bad, terrible, poor, sadness, violence, avoid, hate, poorly, badly, ...

3.2 Refining Word Orientation

In this stage, we use an Error Minimization Algorithm (EMA) to reduce errors in the initial OWD obtained from the first stage. This algorithm is designed based on two following intuitions:

- The more synonyms of a word w occur in a polar class c (either *positive* or *negative*), the more likely w belongs to c .
- The more frequent a word w co-occurs with seed s (*positive* | *negative*), the more likely w expresses orientation of s .

Ideally, EMA tries to linearly combine the semantic information obtained from the WordNet and statistical information captured from the Web to reassign the polar orientation of those words that violate these two intuitions.

Word Error Score

Before going further, we first define several related concepts as follows. Let $synset(w_i)$ be the synonym set of a word w_i , we estimate the Polar Error (PE) for each word w_i by the log-likelihood ratio:

$$PE(w_i) = \log_2 \left(\frac{count[synset_p(w_i)] + \alpha}{count[synset_n(w_i)] + \alpha} \right) \quad (1)$$

where $count[synset_p(w_i)]$ is the number of w_i 's synonyms belonging to the positive seed list and $count[synset_n(w_i)]$ is the number of w_i 's synonyms in the negative seed list. α is the smoothing constant (On default, we choose $\alpha = 0.1$).

In order to utilize the Web corpora, we compute the Semantic Orientation (SO) for each w_i by the same method proposed in [12]:

$$SO(w_i) = \log_2 \left[\frac{hits(w_i, "excellent")hits("poor")}{hits(w_i, "poor")hits("excellent")} \right] \quad (2)$$

Equation (2) is formally a log-odds ratio that estimates semantic orientation of w_i by issuing queries to search engine and counting the hits of returned documents. Here $hits(w_i, "seed\ words")$ denotes the number of documents in which w_i and “seed words” co-occur. In this work, we selected “*excellent*” and “*poor*” as the seed words and chose AltaVista² as the search engine. AltaVista is a

² <http://www.altavista.com>

Table 3. Examples of errors detected by EMA

Positive Errors	WES	Negative Errors	WES
Specialty	+3.138	Half-hearted	-3.815
Superlative	+2.133	Garble	-3.607
Joy	+1.949	Oppressive	-3.383
Effectiveness	+1.595	Crime	-3.142
Perfection	+1.278	Selfish	-3.099
Soulful	+0.971	Fake	-2.329
Mind-blowing	+0.659	Suffer	-1.862

preferred search engine because it provides NEAR operator that restricts the co-occurrence hits within *n-word* window³.

Equation (1) interprets the first intuition that PE is positive when a word has more synonyms belong to positive list and negative when a word has more synonyms belong to negative list. Equation (2) corresponding to the second intuition indicates that SO is positive when a word is more associated with “*excellent*” and negative when a word is more associated with “*poor*”. We define the Word Error Score (WES) for each word w_i by linearly combining the two scores as follows:

$$WES(w_i) = \lambda PE(w_i) + (1 - \lambda) SO(w_i) \quad (3)$$

where λ is the linear parameter ranged in [0,1].

Error Minimization Algorithm

Based on equation (3), we consider a positive word as an error if it has negative WES ($WES < 0$). Similarly, a negative word is tracked as an error if it has positive WES ($WES > 0$). Returning to the earlier examples, “*fake*” and “*joy*” will be considered as errors because “*fake*” is originally identified as positive words but received negative WES ($WES(\text{"fake"}) = -2.695$ with $\lambda = 0.6$) whereas “*joy*” is negatively identified but got positive WES ($WES(\text{"joy"}) = +1.949$ with $\lambda = 0.6$). (Some other errors are shown in Table 3)

Our proposed algorithm starts with the two extracted lists (positive and negative list) derived from the first stage. EMA then computes WES for every word and detects errors in each list. The errors in a positive list are reassigned with negative orientation and added to the negative list. In contrary, the errors in the negative list are reassigned and added to the positive list. Errors in the two lists are summed up in order to assess the total errors in the initial dictionary. The aim of EMA is to minimize total errors by iteratively reassigning orientation for every error in OWD. This iterative algorithm stops when the total errors in the OWD is minimized (converged). We believe that at the last position of

³ In [11], NEAR operator was shown to be more effective than just considering the co-occurrences in the whole document.

```

loop (until total error  $TE <$  threshold  $t$  (converged))
  Set  $pl = \{w_i \mid w_i \in$  positive list $\}$ 
  Set  $nl = \{w_i \mid w_i \in$  negative list $\}$ 
  for each ( $w_i$  in  $pl \mid nl$ )
    Compute  $PE(w_i)$  and  $SO(w_i)$ 
    Compute  $WES(w_i) = \lambda \cdot PE(w_i) + (1 - \lambda) \cdot SO(w_i)$ 
    if ( $w_i \in pl$  and  $WES(w_i) < 0$ )
      Reassign  $w_i$  as negative orientation
      Add  $w_i$  to  $nl$ 
    end if
    if ( $w_i \in nl$  and  $WES(w_i) > 0$ )
      Reassign  $w_i$  as positive orientation
      Add  $w_i$  to  $pl$ 
    end if
  end for each
  Update  $TE = \sum_{p \in pl} WES(w_p) + \sum_{n \in nl} WES(w_n)$ 
end loop

```

Fig. 1. Procedure to minimize errors in the OWD

the sentence, the errors on the initial OWD can be reduced significantly. The complete procedure for minimizing errors in the OWD is shown in Figure 1.

4 Experiment

Although the property of our study is to construct an OWD, is difficult to directly evaluate how effective the dictionary would be. One might think of a possible way to manually annotate the dictionary derived after the first stage and then use it as a gold standard to evaluate the automatically refined OWD at the end. However, this method is not practical since it will take much time and human effort.

Once considering the areas of opinion mining to which OWD might effectively contribute, one can still verify the effectiveness of refined dictionary through a specific mining task. In this paper, we choose sentiment classification task to indirectly evaluate our work.

4.1 Experimental Setting

In order to utilize the OWD in sentiment classification task, we employed lexicon-based approach to classify the sentiment of a given sentence. We used sentence polarity dataset constructed by Pang and Lee⁴ for the experiments. This corpus contains 5,331 positive and 5,331 negative processed sentences in “Movie” domain. By filtering out few non-English sentences, we finally acquired 5,300 positive and 5,300 negative sentences. Additionally, we applied POS tagging⁵ for precisely matching opinion words in the OWD with the ones in the given sentence.

⁴ At <http://www.cs.cornell.edu/People/pabo/moviereview-data> as scale dataset v1.0.

⁵ Available at <http://www-tsujii.is.s.u-tokyo.ac.jp>

Since our work focuses on refining OWD by reassigning the polar orientation of words themselves, we do not pay much attentions to the strength of orientation even though the refining decision given in scores indicates how strong the orientation could be. For this reason, we employed a very simple model for classifying sentences into (*positive|negative*) class. This model was verified to be effective in [8]. The model simply sums up the orientation of every opinion word within the target sentence and then decides the sentence's sentiment as the sign of final orientation as follows:

$$S(s) = \text{sign} \left(\sum_i [\text{orientation}(w_i \in s)] \right) \quad (4)$$

where $S(s)$ is the sentiment of sentence s and $\text{orientation}(w_i)$ is the polar orientation of word i in sentence s . In this experiment, we also use negation rules [2,5,8] to reverse the orientation of those words follow negative expressions such as *never*, *hardly*, *not*, *without*. For example, in the sentence “*not a bad* journey at all”, negative word “**bad**” is reversed to positive due to tag “*not*”.

We first conducted the sentiment classification experiment with original “un-refined” OWD and chose it as a baseline. For verifying the effectiveness of EMA, we performed the classification task with the refined OWD. To see the effects of each resource, we repeated this testing scenario with increasing λ (λ is changed from 0 to 1 with 0.1 rate) and compared the final result with the baseline.

4.2 Result

In the first experiment using original OWD, the accuracy of baseline approach on the dataset is 62.75%. The second experiment using refined OWD yields highest result at 66.19% accuracy (as shown in Figure 2). This result indicates promising improvement even though the classification model is empirically simple. Figure 2. describes the effect of each resource to sentiment classification task. Using EMA shows to be effective to improve the classification performance. Also, it turns out that using only WordNet (accuracy = 65.01 with $\lambda = 1$) is more efficient than only using Web (accuracy = 63.52 with $\lambda = 0$). The highest accuracy achieved at $\lambda = 0.6$ (and remarkable yielding result with $\lambda = 0.5$) indicates that combining both local and global resources takes positive effects.

Table 4 shows some interesting examples that prove the effects of the refined OWD to sentiment classification. In these four examples, the correct classification is made after one opinion word was reassigned. We can see the obvious impact of refined OWD to those sentences which have balanced positive words and negative words. In such sentences, one positive word will take another negative out. The proposed model then makes a classifying decision based on the rest of opinion words. Therefore, errors can easily mislead the classification decision. Our refined OWD performed effectively in such cases.

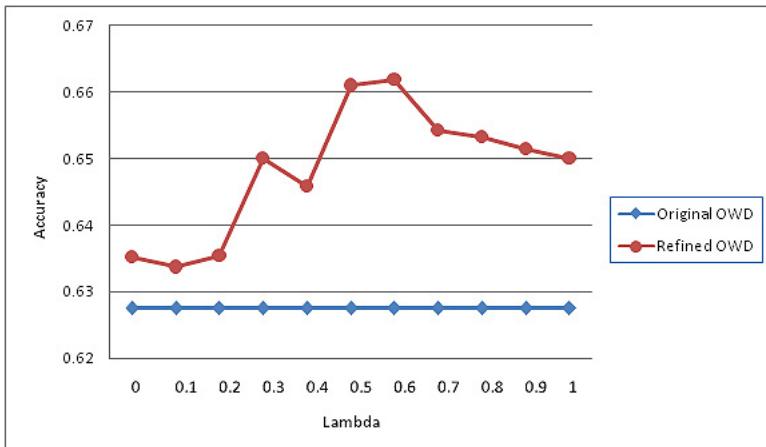


Fig. 2. Results of sentiment classification at sentence level

Table 4. Examples of reassigned sentences by using refined OWD

SENTENCE	grant gets to display his cadness to perfection , but also to show acting range that may surprise some who thought light-hearted comedy was his forte .
ANNOTATION	POSITIVE
CLASSIFICATION	grant[-1] / perfection[-1] / show[+1] / acting[-1] / surprise[+1] / light-hearted[+1] / forte[-1]
REASSIGNED WORD	perfection [+1]
ORIGINAL OWD	NEGATIVE
REFINED OWD	POSITIVE
SENTENCE	blade ii has a brilliant director and charismatic star , but it suffers from rampant vampire devaluation .
ANNOTATION	NEGATIVE
CLASSIFICATION	blade[-1] / brilliant[+1] / charismatic[+1] / star[-1] / suffers[+1] / rampant[+1] / devaluation[-1]
REASSIGNED WORD	suffers -1]
ORIGINAL OWD	POSITIVE
REFINED OWD	NEGATIVE

5 Discussion

If one looks at the performance of sentiment classification task, 66% achieved accuracy in our work seems to be not much impressive. We fairly believe that this result is still promising since we dealt with sentiment classification based on a very simple model. In addition, we experimented our approach on movie domain, which was claimed to be hard to classify in [12]. In that work, the author hypothesized the main reason is that good movie often contains unpleasant

Table 5. Statistical information of opinion words in refined OWD (#Positive, #Negative, and #Reassigned are the number of positive words, number of negative words and number of reassigned words respectively)

λ	#Positive	#Negative	#Reassigned
0	4,361	7,424	3,355
0.6	5,109	6,676	2,479
1	6,242	5,543	1,134

scenes. Positive reviews therefore are less positive orientated if it contains negative opinions of these unpleasant scenes. This common phenomenon was also mentioned as “*thwarted expectation*” in [9]. This problem should be our concern for latter improving classification performance.

In Table 4, there are several ambiguous matchings between opinion words and name entities. For examples, “**grant**” in the first example and “**blade**” in the second example are person’s and movie’s name respectively. Such words should not have orientation in the context of specific sentences. Obviously, we can deal with this ambiguity by applying a named-entity filter to every sentence. Regarding the remarkable amount of named entities in general domains (e.g, Movie), we believe removing named-entity would be a considerable attempt for improving performance of lexicon-based approach in our future work.

Table 5 includes statistical information of refined OWD. At $\lambda = 0$ (corresponding to using only Web for refining OWD), the total amount of reassigned words is surprisingly numerous (3,355 words were reassigned) but the classification performance was not worse than the baseline. In contrast, a smaller amount of words were reassigned when using only WordNet to refine OWD (with $\lambda = 1$, 1,134 words were reassigned) but it showed better performance than previous case. We justify this problem by a hypothesis that using only the Web or WordNet both effectively detects clear errors (those words have high SO or PE score itself clearly indicates their orientation). Those words are important detections that significantly contribute to improve classification performance. Once using the Web based on hits counting to estimate semantic orientation, it can introduce noise with low frequent words. However, noise does not really harm the classification performance because most of them do not occur in the target sentence. Eliminating such noise from refined OWD is one of our future revisions.

Because our EMA only reassigns the orientation of errors without changing the size of OWD, there should be another issue of “*neutral*” words. Examples in Table 4 indicates these *neutral* candidates, such as “**show**” or “**star**”. Again, we intend to investigate more on this problem at the next phase of our work.

6 Conclusion and Future Work

This paper proposed a two-stage approach for constructing dictionary of opinion words. The first stage iteratively expanded dictionary from a small annotated

seed words. The second stage detected errors caused in the first stage and minimized the amount of errors by using EMA. In this work, we focused on a reasonable combination of global and local resources for refining an OWD. The experiments showed that our proposed approach was effective. The refined dictionary brought remarkable effects on sentiment classification task.

For the future work, we first plan to deal with noise (e.g., named entities, low-frequent words and neutral words) in the OWD. Reducing such noise may promisingly enhance the OWD and consequently improve the performance of latter mining tasks. Secondly, we intend to experiment other methods to estimate the polar orientation of errors more precisely. Also, we plan to explore other possible combinations of semantic and statistical information. Finally, we aim to empirically evaluate the effectiveness of OWD by employing other lexicon-based models for classifying sentiment text.

References

1. Dave, K., Lawerence, S., Pennock, D.M.: Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In: WWW, pp. 519–528 (2003)
2. Ding, X., Liu, B., Yu, P.S.: A Holistic Lexicon-Based Approach to Opinion Mining. In: WSDM, pp. 231–240 (2008)
3. Gamon, M., Aue, A.: Automatic Identification of Sentiment Vocabulary: Exploiting Low Association with Known Sentiment Terms. In: EMNLP, pp. 57–64 (2005)
4. Hatzivassiloglou, V., McKeown, K.R.: Predicting the Semantic Orientation of Adjectives. In: ACL, pp. 174–181 (1997)
5. Hu, M., Liu, B.: Mining and Summarizing Customer Reviews. In: KDD, pp. 168–177 (2004)
6. Kanayama, H., Nasukawa, T.: Fully Automatic Lexicon Expansion for Domain-oriented Sentiment Analysis. In: EMNLP, pp. 355–363 (2006)
7. Kamps, J., Marx, M., Mokken, R.J., Rijke, M.D.: Using WordNet to Measure Semantic Orientation of Adjectives. In: LRE, pp. 1115–1118 (2004)
8. Kim, S.M., Hovy, E.: Determining the Sentiment of Opinions. In: COLING, pp. 1367–1373 (2004)
9. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: EMNLP, pp. 79–86 (2002)
10. Riloff, E., Wiebe, J.: Learning Extraction Patterns for Subjective Expressions. In: EMNLP, pp. 105–112 (2003)
11. Turney, P.D.: Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In: ECML, pp. 491–502 (2001)
12. Turney, P.D.: Thumbs up or Thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In: ACL, pp. 417–424 (2002)
13. Whitelaw, C., Garg, N., Argamon, S.: Using Appraisal Groups for Sentiment Analysis. In: CIKM, pp. 625–631 (2005)
14. Yu, H., Hatzivassiloglou, V.: Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences, pp. 129–136 (2003)

An Improvement of PAA for Dimensionality Reduction in Large Time Series Databases

Nguyen Quoc Viet Hung and Duong Tuan Anh

Faculty of Computer Science and Engineering,
HoChiMinh City University of Technology, Vietnam
`{nqvhung, dtanh}@cse.hcmut.edu.vn`

Abstract. Many dimensionality reduction techniques have been proposed for effective representation of time series data. Piecewise Aggregate Approximation (PAA) is one of the most popular methods for time series dimensionality reduction. While PAA approach allows a very good dimensionality reduction, PAA minimizes dimensionality by the mean values of equal sized frames. This mean value based representation may cause a high possibility to miss some important patterns in some time series datasets. In this work, we propose a new approach based on PAA, which we call Piecewise Linear Aggregate Approximation (PLAA). PLAA is the combination of a mean-based and a slope-based dimensionality reduction. We show that PLAA can improve representation precision through a better tightness of lower bound in comparison to PAA.

Keywords: Indexing, Similarity Search, Time Series, PAA, Dimensionality Reduction.

1 Introduction

Efficient and accurate similarity searching for a large amount of time series dataset is an important but non-trivial problem. Time series databases are often extremely large. Searching directly on these data will be very complex and inefficient. To overcome this problem, we should use some of transformation methods to reduce the magnitude of time series databases. These transformation methods, also called dimensionality reduction techniques.

Many dimensionality reduction techniques ([1], [2], [3], [5], [7], [12]) have been proposed for effective representation of time series data. Piecewise Aggregate Approximation (PAA), proposed by Keogh et al. ([5]), is one of the most popular methods for time series dimensionality reduction. While PAA approach allows a very good dimensionality reduction, PAA minimizes dimensionality by the mean values of equal sized frames. Since PAA is a mean value based representation, it may cause a high possibility to miss some important patterns in some kinds of time series datasets.

In this work, we propose a new approach based on PAA, which we call Piecewise Linear Aggregate Approximation (PLAA). For each equal sized segment, beside the mean value, PLAA uses one more value, the slope of the best straight line that fits the data points in the segment. We will show that PLAA can improve representation

precision through a better tightness of lower bound in comparison to PAA. We empirically compare the PLAA with the original PAA and demonstrate its quality improvement.

The rest of this paper is organized as follows. Section 2 briefly discusses the background and some existing works on time series dimensionality reduction. Section 3 introduces our proposed approach. Section 4 contains an experimental evaluation of the approach. Finally, section 5 offers some conclusions and suggestions for future work.

2 Background

A time series X of length n can be considered as a vector or point in an n -dimensional space. A dimensionality reduction method will condense the original time series X (in true space) into a lower dimensional time series X' (in feature space or reduced space). An important result given by Faloutsos et al. (1994) ([3]) is the proof that in order to guarantee completeness (no false dismissals), the distance function used in the feature space must underestimate the true distance measure. In other words, the distance $D(A, B)$ of two time series A, B in true space and the distance $DR(A', B')$ of A' and B' in reduced space (A', B' are the reduced forms of A, B respectively) must satisfy the following condition which is called the bounding inequality:

$$D(A, B) \geq DR(A', B')$$

There has been much work in dimensionality reduction for time series. Some well-known dimensional reduction methods include Discrete Fourier Transform (DFT) ([1], [3]), Discrete Wavelet Transform (DWT) ([2]), Single Value decomposition (SVD) ([5]), Adaptive Piecewise Constant Approximation (APCA) ([7]) and Piecewise Aggregate Approximation (PAA) ([5]). The PAA, proposed by Keogh et al., 2000 is a simple but widely used dimensionality reduction method for time series ([4], [11]). It has been proved theoretically and empirically that in many cases it outperforms some other methods such as DFT and DWT. To go with PAA dimensionality reduction technique, Lin et al., 2003 ([8]) proposed Symbolic Aggregate Approximation (SAX), a novel method to transform a condensed time series (through PAA) to a symbolic string. Lkhagva et al., 2006 ([9]) proposed Extended SAX, an improved variant of SAX for financial time series.

2.1 Dimensionality Reduction Via PAA

Using the PAA dimensionality reduction method ([5]), a time series T of length n , $T = t_1, \dots, t_n$, can be represented in a w -dimensional space by a $\bar{T} = \bar{t}_1, \dots, \bar{t}_w$. The \bar{t}_i is calculated by the following equation:

$$\bar{t}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} t_j$$

According to [5], “to reduce the time series from n dimensions to w dimensions, the data is divided into w equal sized segments. The mean value of the data within a segment is calculated and a vector of these values becomes the data-reduced representation”.

2.2 Limitations of PAA

Even though PAA is a popular method for dimensionality reduction, it still suffers two disadvantages. Firstly, since PAA is based on mean values approximation, it has high possibility to miss some important patterns in some time series datasets. The Fig.1 shows the cases that some frames have the same mean value but their Euclidean distances are quite large. Secondly, PAA representation has not yield a good tightness of lower bound.

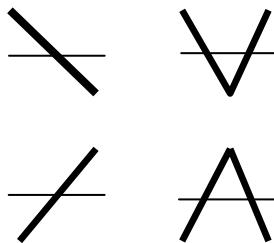


Fig. 1. A limitation of PAA: Some frames have the same mean value but their Euclidean distances are quite large

3 Piecewise Linear Approximation (PLAA)

To mitigate the two disadvantages of PAA, we propose an improved form of PAA, named Piecewise Linear Aggregate Approximation (PLAA). PLAA is the combination of a mean-based and a slope-based dimensionality reduction. PLAA can improve representation preciseness through a better tightness of lower bound in comparison to PAA.

3.1 Main Ideas

In PAA representation, we transform the original time series S of length n to S' , a time series of length w in reduced space. Corresponding to $k = n/w$ contiguous points $(m+1, x_{m+1}) \dots (m+k, x_{m+k})$ in the original space X , we maps them into one real value A' in the reduced space Y . A' is the mean value of the data points falling within the frame in the original space X .

$$A' = \frac{\sum_{i=m+1}^{m+k} x_i}{k}$$

However, in PLAA, k contiguous points in the original space X , are mapped not only to a value A' in the reduced space X but also a value A'' in the reduced space Z . A' is the mean value of the data points in the frame and A'' is the slope of the best straight line that fits the data points. The value of A'' is defined as:

$$A'' = \frac{k \left(\sum_{i=m+1}^{m+k} i^* x_i \right) - \left(\sum_{i=m+1}^{m+k} i \right) \left(\sum_{i=m+1}^{m+k} x_i \right)}{n \left(\sum_{i=m+1}^{m+k} i^2 \right) - \left(\sum_{i=m+1}^{m+k} i \right)^2} \quad (1)$$

Equation (1) is not dependent on the position of the point group. When the i -th point group is shifted to $(i+\delta)$ -th point group, the δ value in equation (1) will be removed after a simple mathematical manipulation.

3.2 The Distance Function

Given two time series $S = s_1, \dots, s_n$, and $Q = q_1, \dots, q_n$. S , Q are also viewed as two n -dimensional vectors. Through PLAA transformation, S , Q are mapped into 4 w -dimensional vectors $S' = s'_1 s'_2 \dots s'_{w'}, S'' = s''_1 s''_2 \dots s''_{w''}$, $Q' = q'_1 q'_2 \dots q'_{w'}$ and $Q'' = q''_1 q''_2 \dots q''_{w''}$ in which S' , Q' keeps the mean values and S'' , Q'' keep the slopes. Now, we consider some distance measures.

The distance between S and Q in the original space is defined as follows.

$$D(S, Q) = \sqrt{\sum_{i=1}^n (s_i - q_i)^2}$$

From [5], in PAA, the distance between S' and Q' is defined as.

$$D(S', Q') = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^w (s'_i - q'_i)^2}$$

PAA can be used as a dimensionality reduction method since it satisfies the bounding inequality ([5]). That means:

$$\sum_{i=1}^n (s_i - q_i)^2 \geq \frac{n}{w} \sum_{i=1}^w (s'_i - q'_i)^2$$

However, in this work, we can prove that PLAA brings out a better tightness of lower bound (see Appendix A), as follows:

$$\sum_{i=1}^n (s_i - q_i)^2 \geq \frac{n}{w} \sum_{i=1}^w (s'_i - q'_i)^2 + \frac{\left(\frac{n}{w}-1\right)\left(\frac{n}{w}+1\right)}{18} \sum_{i=1}^w (s''_i - q''_i)^2$$

Therefore, in PLAA, the lower bounding approximation of the distance between two points $\langle S', S'' \rangle$ and $\langle Q', Q'' \rangle$ in the reduced spaces $\langle Y, Z \rangle$ is given by:

$$DR(S', S'', Q', Q'') = \sqrt{\frac{n}{w} \sum_{i=1}^w (s'_i - q'_i)^2 + \frac{\left(\frac{n}{w}-1\right)\left(\frac{n}{w}+1\right)}{18} \sum_{i=1}^w (s''_i - q''_i)^2}$$

In PLAA, we obtain a lower bound for $D(S, Q)$, which is tighter than that of the PAA. Moreover, the factor $(n/w - 1)(n/w + 1)^2/18$ in $DR(S', S'', Q', Q'')$ is significant. When n/w is large, the factor will be higher. So, in the case of high dimensionality reduction, PLAA is better than PAA.

3.3 A Time Series Indexing Method Used with PLAA or PAA

The general time series query algorithm is as follows. Given a query subsequence Q , we try to find in the original sequence S any subsequence S_i such that its Euclidean distance to Q is smallest. To find such a S_i , we scan sequentially each subsequence. If the subsequence under consideration has the distance to Q smaller than the distance to the best match so far (*best_dis_sofar*), we will update the best match so far. Otherwise, we move to the next subsequence. So, the total number of times of checking is $|S| - |Q| + 1$ where $|S|, |Q|$ are the lengths of the sequences S and Q .

When using with PLAA or PAA, the above query algorithm can be modified as follows. At each subsequence under consideration S_i we calculate the distance dr between S'_i and Q' where S'_i and Q' are the transformed forms of S_i and Q . If dr is greater than the distance from the best match to Q' ($dr > best_dis_sofar$), then the actual distance d will be greater than *best_dis_sofar* and we do not need to check on the original sequence. Otherwise, we have to check S_i against Q in the actual space to avoid false alarms.

Notice that the time series, which is transformed through PLAA or PAA, can be indexed by a standard spatial access method such as R-tree and its many variants. The indexing tree represents the transformed sequences as points in w -dimensional space.

4 Experimental Results

The proposed approach was tested on Pentium IV 2.8GHz 512MB RAM PC with some stock time series data sets downloaded from Internet ([10]). The experiments focused on testing the proposed dimensionality reduction method, PLAA. The comparison between PAA and PLAA is based on the tightness of lower bounds, pruning power, and implementation systems. In our experiment, PAA or PLAA was used along with the indexing algorithm given in section 3.3.

4.1 Experiment Results: The Tightness of the Lower Bound

Although from the inequality (8), we can decide that the tightness of the lower bound in PLAA is better than that of PAA, we still wish to test this finding empirically. The tightness of lower bound of a time series indexing method is given by the formula.

$$\text{Tightness of Lower Bound} = \frac{D}{D(S, Q)}$$

where D is the distance measured on the transformed sequences using the selected method and $D(S, Q)$ is the precise distance measured on the original data.

To experiment on the tightness of lower bound, 10,000,000 pairs of segments of the length $n = 1024$ were tested. These segments were condensed into segments of w

points. The fig.2 shows the tightness of lower bounds that were gathered and estimated with regard to the ratio n/w .

The results show that for any value of n/w , the tightness of lower bound of PLAA is higher than that of PAA. Especially, at higher value of n/w , there is a considerable difference between them. This is due to the fact that the term $(n/w - 1)(n/w + 1)^2/18$ will become significant when n/w increasing.

Fig. 2. The empirically estimated tightness of lower bounds of the PAA and PLA

4.2 Experiment Results: Pruning Power

To compare the effectiveness of the two approaches PAA and PLAA, we need to compare their pruning powers. The *pruning power* is the fraction of the database that must be examined before we can guarantee that we have found the nearest match to a query. This ratio is based on the number of times we cannot perform similarity search on the transformed data and have to check directly on the original data to find a nearest match.

$$P = \frac{\text{Number of objects that must be examined directly}}{\text{Number of objects in database}}$$

Fig.3 shows the value of P over a range of reduction ratios. The experiment was conducted on 40 megabytes of a dataset with about 3,000,000 data points. From this dataset, we extract 1000 random queries with size $n = 1024$. The Fig.3 shows the results of pruning power that were compared in terms of the ratio n/w . When the value of P become smaller, approaching to 0, the querying approach is more effective. From Fig.3, we notice that when the reduction ratio n/w becomes larger, the tightness of lower bound decreases and the pruning power become closer to 1.

Fig. 3. The pruning power of the PAA and PLAA methods

4.3 Experiment Results: Implementation Systems

Although the two previous experiments are powerful predictors of the performance of time series indexing systems using two different dimensionality reduction schemes, we also conducted another experiment to compare implemented systems for completeness.

To evaluate the performance of the two competing methods, we measured the normalized CPU cost. The *normalized CPU cost* is the ratio of average CPU time to execute a query using the index to the average CPU cost required to perform a linear (sequential) scan. The normalized cost of a linear scan is 1.0.

The experiments were conducted over a range of database sizes and reduction ratios. Firstly, the experiment was conducted on a dataset with about 3,000,000 data points and with 1000 random queries. Fig.4.A shows the results of this experiment. With the ratio n/w at the values 16, 32, 34, 128, 256, there is relatively little difference between the two methods. Therefore, Fig.4.A gives a “zoom in” of the normalized CPU costs of the two methods.

(A) (B)

Fig 4. (A) The normalized cost of PAA and PLAA according to reduction ratios. **(B)** The normalized cost of PAA and PLAA according to database sizes.

According to the experimental results in Fig.4.A, the normalized CPU cost does not vary linearly to the reduction ratio n/w . From this experiment, the time series indexing method will be very effective when n/w falls in the range [32, 128].

Based on the Fig.3 and Fig.4.A, we can see that with n/w of value 64, the indexing system will be of the most effective. With $n/w = 64$, we conducted the experiment on the datasets with 100,000, 200,000, 400,000, 800,000, 1,600,000 data points. Fig.4.B shows the experimental results.

From Fig.4.B, we have the following observations. Firstly, with $n/w = 64$, the normalized CPU decreases almost 200 times. Secondly, PLAA is always faster than PAA on the large datasets.

5 Conclusion

The main contribution of this paper is to propose a new dimensionality reduction method, PLAA, which is an improvement of PAA. PLAA transformation offers two main advantages over the PAA. By adding one more important value in equal sized segments, PLAA yields a better tightness of lower bound and provides a more precise representation for many different datasets.

To compare PAA and PLAA, we have experimented the two approaches on three different tests: the tightness of lower bound, pruning power and implementation systems. Basing on mathematical analysis as well as empirical experiments, we can conclude that PLAA is better than PAA.

There can be several future research directions using this approach. One of the future work is to develop a method to symbolize the condensed time series in PLAA representation into a discrete string. Besides, the preliminary experimental results presented here mainly focus on similarity search. We think that PLAA also can be effectively used for other data mining tasks such as clustering, novelty detection, classification and other tasks.

References

1. Agrawal, R., Faloutsos, C., Swami, A.N.: Efficient similarity search in sequence databases. In: The 4th Foundations of Data Organization and Algorithms, pp. 69–84. Springer, Heidelberg (1993)
2. Chan, K., Fu, W.: Efficient time series matching by wavelets. In: The 15th IEEE International Conference on Data Engineering, pp. 126–133. IEEE Press, Los Alamitos (1999)
3. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast Subsequence Matching in Time Series Databases. In: The 1994 ACM SIGMOD Conference On Management of Data, pp. 419–429. ACM, New York (1994)
4. Han, W.S., Lee, J., Moon, Y.S., Jiang, H.: Ranked Subsequence Matching in Time-Series Databases. In: The 33rd International Conference on Very Large Data Bases, pp. 423–434. ACM, New York (2007)
5. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. In: Knowledge and Information Systems, vol. 3(3), pp. 263–286. Springer, Heidelberg (2001)

6. Keogh, E., Chu, S., Hart, D., Pazzani, M.: An Online Algorithm for Segmenting Time Series. In: IEEE International Conference on Data Mining, pp. 289–296. IEEE Press, Los Alamitos (2001)
7. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Locally adaptive dimensionality reduction for indexing large time series databases. In: ACM SIGMOD Conference on Management of Data, pp. 151–162. ACM, New York (2001)
8. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In: The 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discover, pp. 2–11. ACM, New York (2003)
9. Lkhagva, B., Suzuki, Y., Kawagoe, K.: New Time Series Data Representation ESAX for Financial Applications. In: 22nd International Conference on Data Engineering Workshops, p. 155. IEEE Press, Los Alamitos (2006)
10. Time Series Stock Data,
<http://www-cs.ucr.edu/~wli/FilteringData/stock.zip>
11. Toshniwal, D., Joshi, R.C.: Finding Similarity in Time Series Data by Method of Time Weighted Moments. In: The 16th Australasian Database Conference, pp. 155–164. IEEE Press, Los Alamitos (2005)
12. Yi, B.K., Faloutsos, C.: Fast Time Sequence Indexing for Arbitrary L_p Norms. In: The 26th International Conference on Very Large Data Bases, pp. 285–394. ACM, New York (2000)

Appendix A

Given two set of data points, A consisting of n data points: $(1, a_1), (2, a_2), \dots (n, a_n)$ and B consisting of $(1, b_1), (2, b_2), \dots (n, b_n)$. We need to prove that:

$$D \geq n(a' - b')^2 + \frac{(n-1)(n+1)^2}{18}(\alpha_a - \alpha_b)^2 \quad (\text{A1})$$

where a' , b' are the mean values of two set of data points A , B respectively. That means

$$a' = \frac{1}{n} \sum_{i=1}^n a_i, \quad b' = \frac{1}{n} \sum_{i=1}^n b_i$$

Let α_a , α_b be the slopes of the best fitting straight lines for the two set of data points A , B respectively. Thus, from (1) that means:

$$\begin{aligned} \alpha_a &= \frac{n \sum_{i=1}^n i * a_i - \frac{n(n+1)}{2} \sum_{i=1}^n a_i}{n \sum_{i=1}^n i^2 - (\sum_{i=1}^n i)^2} = \frac{\sum_{i=1}^n 2i * a_i - (n+1)\sum_{i=1}^n a_i}{\frac{n(n-1)(n+1)}{6}} \\ \alpha_b &= \frac{\sum_{i=1}^n 2i * b_i - (n+1)\sum_{i=1}^n b_i}{\frac{n(n-1)(n+1)}{6}} \end{aligned}$$

Let D be the Euclidean distance between two sets of data points A and B . That means:

$$D = (a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2$$

Now let denote

$$d_i = a_i - b_i$$

The inequality (A1) becomes:

$$\begin{aligned} \sum_{i=1}^n d_i^2 &\geq \frac{\left(\sum_{i=1}^n d_i\right)^2}{n} + \frac{(n-1)(n+1)^2}{18} \left(\frac{\sum_{i=1}^n (2i * d_i) - (n+1)\sum_{i=1}^n d_i}{\frac{n(n-1)(n+1)}{6}} \right)^2 \\ \sum_{i=1}^n d_i^2 &\geq \frac{\left(\sum_{i=1}^n d_i\right)^2}{n} + \frac{2}{(n-1)n^2} \left(\sum_{i=1}^n (2i * d_i) - (n+1)\sum_{i=1}^n d_i \right)^2 \\ \sum_{i=1}^n d_i^2 &\geq \left(\sum_{i=1}^n d_i \right)^2 + \frac{2}{(n-1)n} \left(\sum_{i=1}^n (2i * d_i) - (n+1)\sum_{i=1}^n d_i \right)^2 \end{aligned} \quad (\text{A2})$$

Notice that:

$$\begin{aligned} n \sum_{i=1}^n d_i^2 - \left(\sum_{i=1}^n d_i \right)^2 &= (d_1 - d_2)^2 + (d_1 - d_3)^2 + \dots + (d_1 - d_n)^2 + \\ &\quad + (d_2 - d_3)^2 + (d_2 - d_4)^2 + \dots + (d_2 - d_n)^2 \\ &\quad + \dots \\ &\quad + (d_{n-1} - d_n)^2 \end{aligned}$$

Therefore, applying Cauchy-Schwarz inequality¹:

$$\begin{aligned} n \sum_{i=1}^n d_i^2 - \left(\sum_{i=1}^n d_i \right)^2 &\geq \frac{1}{\frac{n(n-1)}{2}} ((n-1)d_1 + (n-3)d_2 + \dots + (-n+1)d_n)^2 \\ &\geq \frac{2}{n(n-1)} \left((n+1)\sum_{i=1}^n d_i - \sum_{i=1}^n (2i * d_i) \right)^2 \end{aligned} \quad (\text{A3})$$

From (A1), (A2), and (A3) we derive:

$$D \geq n(a - b)^2 + \frac{(n-1)(n+1)^2}{18} (\alpha_a - \alpha_b)^2$$

This completes the proof. When the inequality (A1) can be proved, the lower bound of PLAA is $DR(S', S'', Q', Q'')$.

¹ $\left(\left(\sum_{i=1}^n x_i^2 \right) \left(\sum_{i=1}^n y_i^2 \right) \right) \geq \left(\sum_{i=1}^n x_i y_i \right)^2 \quad \forall n \in \mathbb{N} \wedge x_i, y_i \in R.$

Stability Margin for Linear Systems with Fuzzy Parametric Uncertainty

Petr Hušek*

Department of Control Engineering,
Faculty of Electrical Engineering,
Czech Technical University in Prague,
Technická 2, 166 27 Prague,
Czech Republic
husek@fel.cvut.cz
<http://dce.felk.cvut.cz>

Abstract. The paper deals with the problem of determining stability margin of a linear continuous-time system with fuzzy parametric uncertainty. Non-symmetric multivariate ellipsoidal membership functions describing the uncertainty of coefficients of characteristic polynomial are considered. An elegant solution, graphical in nature, based on generalization of Tsyplkin-Polyak plot is presented.

Keywords: Fuzzy parametric uncertainty; Stability analysis; Continuous-time systems; Uncertain polynomials.

1 Introduction

When dealing with real systems it is not possible to obtain an accurate model of a system, some uncertainty has to be always considered. If the structure of a system is supposed to be given but the parameters are not known precisely we speak about parametric uncertainty. In engineering practice it is of fundamental importance that the systems preserve stable behaviour for a whole admissible parameter variations. In this view it could be also appropriate to know, if a system is stable for some nominal values of its parameters, within what boundary the stability remains preserved. Such a problem is called *stability margin* determination.

The problems mentioned above and solved by classical robust analysis approach ([2]) assume that the uncertainty remains the same independently on the working conditions. It means that the worst case has to be considered and conservative results are obtained. However, in many practical situations the uncertainty varies, e.g. depending on operation conditions. In such a case the uncertainty interval can be often parameterized by a confidence level. This parameter

* This work has been supported by the Research Program MSM6840770038 (sponsored by the Ministry of Education of the Czech Republic), the project INGO 1P2007LA297 and the project 1H-PK/22 (sponsored by Ministry of Industry and Trade of the Czech Republic).

is usually tough to measure but it can be estimated by a human operator. If each coefficient of a system is described in this way the system corresponds to a family of interval linear time-invariant systems parameterized by the confidence level.

To handle such type of uncertain systems a mathematical framework is desired. Such a framework was proposed by Bondia and Picó in [5]. They adopted the concept of *fuzzy numbers* and *fuzzy functions* [7]. The approach interprets a set of intervals parameterized by a confidence level as a fuzzy number with its membership degree given by this confidence level. It means that all the parameters c_i are characterized by means of fuzzy numbers with membership functions $\alpha_i = \mu_{\tilde{c}_i}(c_i)$. When a confidence level α_i is specified then the parameter interval is determined by the α_i -cut $[\tilde{c}_i]_{\alpha_i}$. If $\alpha_i = 1$ (the maximum confidence level – the system works in normal operating conditions) the parameter c_i can take any value (crisp or interval) within the cores of \tilde{c}_i 's ($c_i = \text{ker}(\tilde{c}_i)$). If $\alpha_i = 0$ (the minimum confidence level) the parameter c_i is the interval equal to the support of \tilde{c}_i ($c_i \in \text{supp}(\tilde{c}_i)$).

2 Interval Fuzzy Linear Systems

Let us consider a linear system with its parameters entering coefficients of characteristic polynomial independently. Such polynomial can be written as

$$\tilde{p}(s) = \tilde{a}_0 + \tilde{a}_1 s + \dots + \tilde{a}_n s^n \quad (1)$$

where the coefficients \tilde{a}_i , $i = 0, \dots, n$ are described by fuzzy sets with membership functions $\mu_{\tilde{a}_i}(a_i)$.

To be able to use techniques known from robust control theory it is convenient to represent the varying intervals expressed by the α -cuts by parameterization of varying endpoints of these intervals. If convex membership functions are used it is always possible to write $[\tilde{a}_i]_\alpha = [a_i^-(\alpha_i), a_i^+(\alpha_i)] \stackrel{\text{def}}{=} a_i(\alpha_i)$ where α_i is the confidence level in i -th parameter and $a_i^-(\cdot)$, $a_i^+(\cdot)$ are strictly increasing and strictly decreasing functions, respectively. If $a_i^0 = \text{ker}(\tilde{a}_i)$ and $[a_i^-, a_i^+] = \text{supp}(\tilde{a}_i)$ then

$$\begin{aligned} a_i^-(\alpha_i) &= \mu_{\tilde{a}_i}^{-1}(\alpha_i) \quad \text{for } a_i \leq a_i^0 \\ a_i^+(\alpha_i) &= \mu_{\tilde{a}_i}^{-1}(\alpha_i) \quad \text{for } a_i \geq a_i^0. \end{aligned}$$

The functions $a_i^-(\alpha_i)$ and $a_i^+(\alpha_i)$ satisfy $a_i^-(0) = a_i^-$, $a_i^+(0) = a_i^+$ and $a_i^-(1) = a_i^+(1) = a_i^0$.

For common confidence level $\alpha = \alpha_i$, the α -cut representation of polynomial (1) corresponds to an interval polynomial

$$[\tilde{p}(s)]_\alpha = p(s, \alpha) = a_0 + a_1 s + \dots + a_n s^n \quad (2)$$

where $a_i = [\tilde{a}_i]_\alpha$.

The main task of stability analysis of such polynomial is to determine its *stability margin*, i.e. minimum confidence level α preserving stability of (2). The problem has been solved using a binary search in [11] or using Argoun stability test [1] in [6] or with the help of Kharitonov theorem [8] and Tsyplkin-Polyak locus in [9].

Nevertheless, the parameters of a system or the coefficients of characteristic polynomials are very often identified using measured input-output data. In such case it is more realistic to characterize the set of parameters by a multidimensional membership function rather than employing fuzzy numbers. For example when utilizing well-known prediction error (PE) identification algorithm ([10], [12], [3], [4]) the coefficients $\mathbf{a} = [a_0, \dots, a_n]^T$ lie in an ellipsoidal set

$$(\mathbf{a} - \mathbf{a}^0)^T \Gamma_a (\mathbf{a} - \mathbf{a}^0) \leq 1 \quad (3)$$

where Γ_a is a positive definite matrix. Then it is reasonable to consider fuzzy set with membership function

$$\mu_{\tilde{\mathbf{a}}}(\mathbf{a}) = -(\mathbf{a} - \mathbf{a}^0)^T \Gamma (\mathbf{a} - \mathbf{a}^0) + 1 \quad (4)$$

where the confidence level $\alpha = \mu_{\tilde{\mathbf{a}}}(\mathbf{a})$ indicates the belief in the experiment the measured data were obtained by. For $\alpha = 0$ the coefficient vector \mathbf{a} can take any value inside hyperellipsoid $(\mathbf{a} - \mathbf{a}^0)^T \Gamma (\mathbf{a} - \mathbf{a}^0) \leq 1$, for $\alpha = 1$ equals to the nominal value, $\mathbf{a} = \mathbf{a}^0$. The natural question arises what a minimum confidence level α_{\min} is necessary so that the α -cut polynomial

$$[\tilde{p}(s)]_\alpha = p(s, \alpha) = a_0 + a_1 s + \dots + a_n s^n; \quad \mu_{\tilde{\mathbf{a}}}(\mathbf{a}) \geq \alpha_{\min} \quad (5)$$

remains stable.

3 Problem Formulation

In the sequel we will consider polynomial

$$\tilde{p}(s) = \tilde{a}_0 + \tilde{a}_1 s + \dots + \tilde{a}_n s^n \quad (6)$$

with the vector of coefficients $\tilde{\mathbf{a}} = [\tilde{a}_0, \dots, \tilde{a}_n]^T$ described by fuzzy set with non-symmetric membership function

$$\mu_{\tilde{\mathbf{a}}}(\mathbf{a}) = -(\mathbf{a} - \mathbf{a}^0)^T \Gamma (\mathbf{a} - \mathbf{a}^0) + 1 \quad (7)$$

where $\mathbf{a}^0 = [a_0^0, \dots, a_n^0]^T$ is a nominal point and Γ is $(n+1) \times (n+1)$ square diagonal matrix

$$\Gamma = \begin{bmatrix} \frac{1}{\gamma_0^2} & & & & 0 \\ & \frac{1}{\gamma_1^2} & & & \\ & & \ddots & & \\ 0 & & & & \frac{1}{\gamma_n^2} \end{bmatrix} \quad (8)$$

with

$$\begin{aligned}\gamma_k &= \gamma_k^+ > 0 \quad \text{for} \quad a_k \geq a_k^0 \\ \gamma_k &= \gamma_k^- > 0 \quad \text{for} \quad a_k < a_k^0, k = 0, \dots, n.\end{aligned}\quad (9)$$

The α -cut representation of polynomial (6) is defined as an uncertain polynomial

$$[\tilde{p}(s)]_\alpha = p(s, \alpha) = a_0 + a_1 s + \dots + a_n s^n \quad (10)$$

such that $\mu_{\bar{\mathbf{a}}}(\mathbf{a}) \geq \alpha$, $\mathbf{a} = [a_0, \dots, a_n]^T$.

Let us note that for any $0 \leq \alpha \leq 1$ the corresponding coefficient space of polynomial (10) is a non-symmetric hyperellipsoid.

Let us suppose that the nominal (1-cut) polynomial $p(s, 1) = \sum_{i=0}^n a_i^0 s^i$ is stable. The task is to find stability margin of the polynomial (6), i.e. minimum confidence level $\alpha_{\min} \in [0, 1]$ such that uncertain polynomial $p(s, \alpha)$ is stable for $\alpha > \alpha_{\min}$ and unstable for $\alpha \leq \alpha_{\min}$.

In order to solve the problem a generalization of the Tsyplkin-Polyak plot will be used [13].

4 Generalized Tsyplkin-Polyak Plot

Let us consider family of polynomials

$$p(s, A) = a_0 + a_1 s + \dots + a_n s^n, \mathbf{a} = [a_0, \dots, a_n]^T, \mathbf{a} \in A, A \subset \Re^{n+1} \quad (11)$$

centered at a nominal point $\mathbf{a}^0 = [a_0^0, a_1^0, \dots, a_n^0]^T$ with the coefficients lying in a non-symmetric hyperellipsoid of radius ρ , i.e.

$$A := \left\{ \mathbf{a} : \left[\sum_{k=0}^n \left| \frac{a_k - a_k^0}{\gamma_k} \right|^2 \right]^{\frac{1}{2}} \leq \rho \right\} \quad (12)$$

where $\gamma_k = \gamma_k^-$ for $a_k < a_k^0$ and $\gamma_k = \gamma_k^+$ for $a_k \geq a_k^0$.

In (12) $\gamma_k^- > 0$ and $\gamma_k^+ > 0$ are given lengths of the semiaxes of the ellipsoid for coefficients lying below and above their nominal values respectively. The family of polynomials (11) associated with the set (12) is loosely referred to as a non-symmetric hyperellipsoid of polynomials.

Let us note that for $\rho = 1 - \alpha$ the polynomial (10) is identical to the polynomial (11). It means that if we are able to determine the maximum $\rho = \bar{\rho}$ preserving robust stability of polynomial (11) with coefficient space (12) then the stability margin α_{\min} of polynomial (6) can be easily computed as

$$\alpha_{\min} = \begin{cases} 1 - \bar{\rho} & \text{for } \bar{\rho} \leq 1 \\ 0 & \text{for } \bar{\rho} > 1 \end{cases}. \quad (13)$$

The main result of the paper is based on well-known test of stability of a family of polynomials [2].

Theorem 1. (Zero exclusion principle). The family of polynomials $p(s, A)$ (11) is Hurwitz stable if and only if

- a) there exists a stable polynomial $p(s, \mathbf{a}^*)$, $\mathbf{a}^* \in A$,
- b) $0 \notin p(j\omega, A) \forall \omega \in \mathfrak{R}$,
- c) the coefficient a_n does not include 0.

Considering $p_1(\omega) = h(\omega)/S(\omega) + jg(\omega)/T(\omega)$ where $S(\omega)$ and $T(\omega)$ are positive functions of $\omega \geq 0$ where $\lim_{\omega \rightarrow \infty} h(\omega)/S(\omega)$ and $\lim_{\omega \rightarrow \infty} g(\omega)/T(\omega)$ are finite instead of $p(j\omega) = h(\omega) + j\omega g(\omega)$ where $h(s)$ and $sg(s)$ are the even and odd parts of the polynomial $p(s)$ respectively a variation of Zero exclusion principle can be stated.

Theorem 2. The family of polynomials $p(s, A)$ (11) is Hurwitz stable if and only if

- a) there exists a Hurwitz stable polynomial $p(s, \mathbf{a}^*)$, $\mathbf{a}^* \in A$,
- b) $0 \notin p_1(\omega, A) \forall \omega \geq 0$,
- c) for $\omega = \infty$ the value set $p_1(\omega, A)$ does not include points on the imaginary axis for n even or points on the real axis for n odd,
- d) for $\omega = 0$ the value set $p_1(\omega, A)$ does not include points on the imaginary axis.

Let us again decompose a member of family of polynomials (11) into its even and odd part. For $s = j\omega$ we can write

$$p(j\omega, \mathbf{a}) = h(\omega, \mathbf{a}) + j\omega g(\omega, \mathbf{a}), \mathbf{a} \in A. \quad (14)$$

The nominal polynomial $p_0(s)$ evaluated in $s = j\omega$ then can be written as

$$p_0(j\omega) = p(j\omega, \mathbf{a}^0) = h_0(\omega) + j\omega g_0(\omega) \quad (15)$$

where

$$\begin{aligned} h_0(\omega) &= a_0^0 - a_2^0 \omega^2 + a_4^0 \omega^4 - \dots, \\ g_0(\omega) &= a_1^0 - a_3^0 \omega^2 + a_5^0 \omega^4 - \dots. \end{aligned} \quad (16)$$

Denote

$$\begin{aligned} S(\omega) &= 0.5(S^-(\omega) + S^+(\omega)) + 0.5(S^-(\omega) - S^+(\omega)) \operatorname{sgn} h_0(\omega), \\ S^+(\omega) &= \left(\sum_{k/2 \text{even}} (\gamma_k^+ \omega^k)^2 + \sum_{k/2 \text{odd}} (\gamma_k^- \omega^k)^2 \right)^{\frac{1}{2}}, \\ S^-(\omega) &= \left(\sum_{k/2 \text{even}} (\gamma_k^- \omega^k)^2 + \sum_{k/2 \text{odd}} (\gamma_k^+ \omega^k)^2 \right)^{\frac{1}{2}}, \\ T(\omega) &= 0.5(T^-(\omega) + T^+(\omega)) + 0.5(T^-(\omega) - T^+(\omega)) \operatorname{sgn} g_0(\omega), \\ T^+(\omega) &= \left(\sum_{(k-1)/2 \text{even}} (\gamma_k^+ \omega^{(k-1)})^2 + \sum_{(k-1)/2 \text{odd}} (\gamma_k^- \omega^{(k-1)})^2 \right)^{\frac{1}{2}}, \\ T^-(\omega) &= \left(\sum_{(k-1)/2 \text{even}} (\gamma_k^- \omega^{(k-1)})^2 + \sum_{(k-1)/2 \text{odd}} (\gamma_k^+ \omega^{(k-1)})^2 \right)^{\frac{1}{2}}. \end{aligned} \quad (17)$$

Then the key theorem can be stated.

Theorem 3. The uncertain polynomial (11) is stable if and only if frequency plot of $h_0(\omega)/S_2(\omega) + jg_0(\omega)/T_2(\omega)$

- a) goes through n quadrants in counterclockwise direction,
- b) does not intersect circle centered at origin with radius ρ ,
- c) $a_n^0 > \rho\gamma_n^-$,
- d) $a_0^0 > \rho\gamma_0^-$.

Proof. Since both functions $S(\omega)$ and $T(\omega)$ are positive for $\omega \geq 0$ and $\lim_{\omega \rightarrow \infty} h(\omega, \mathbf{a})/S(\omega)$ and $\lim_{\omega \rightarrow \infty} g(\omega, \mathbf{a})/T(\omega)$ are finite for all $p(s, \mathbf{a}), \mathbf{a} \in A$ defined by (12) theorem 2 can be applied. The equivalence of the conditions a), c) and d) of the theorem 2 and 3 is evident. To show equivalence of the conditions b) denote by $\Delta a_k = a_k - a_k^0$ and $\mu_k = \Delta a_k/\gamma_k^+$ for $\Delta a_k \geq 0$, $\mu_k = \Delta a_k/\gamma_k^-$ for $\Delta a_k < 0$. The deviations of even and odd parts of a polynomial then can be expressed as

$$\begin{aligned}\Delta h(\omega) &= h(\omega, \mathbf{a}) - h_0(\omega) = \sum_{k \text{ even}} (-1)^{k/2} \Delta a_k \omega^k, \\ \Delta g(\omega) &= g(\omega, \mathbf{a}) - g_0(\omega) = \sum_{k \text{ odd}} (-1)^{(k-1)/2} \Delta a_k \omega^{k-1}\end{aligned}\quad (18)$$

respectively.

Let us discuss four different cases according to the signs of $\Delta h(\omega)$ and $\Delta g(\omega)$.

1. $\Delta h(\omega) \geq 0, \Delta g(\omega) \geq 0$:

For $\Delta h(\omega) \geq 0$ we can write

$$\Delta h(\omega) \leq \sum_{k/2 \text{ even}} \mu_k \gamma_k^+ \omega^k - \sum_{k/2 \text{ odd}} \mu_k \gamma_k^- \omega^k. \quad (19)$$

For its absolute value we have

$$|\Delta h(\omega)| \leq \sum_{k/2 \text{ even}} |\mu_k \gamma_k^+ \omega^k| + \sum_{k/2 \text{ odd}} |\mu_k \gamma_k^- \omega^k|. \quad (20)$$

Applying Hölder's inequality one obtains

$$|\Delta h(\omega)| \leq \left(\sum_{k \text{ even}} |\mu_k|^2 \right)^{\frac{1}{2}} \left(\sum_{k/2 \text{ even}} (\gamma_k^+ \omega^k)^2 + \sum_{k/2 \text{ odd}} (\gamma_k^- \omega^k)^2 \right)^{\frac{1}{2}}. \quad (21)$$

Analogically, for $\Delta g(\omega) \geq 0$ we have

$$\Delta g(\omega) \leq \sum_{(k-1)/2 \text{ even}} \mu_k \gamma_k^+ \omega^{(k-1)} - \sum_{(k-1)/2 \text{ odd}} \mu_k \gamma_k^- \omega^{(k-1)} \quad (22)$$

and

$$|\Delta g(\omega)| \leq \left(\sum_{k \text{ odd}} |\mu_k|^2 \right)^{\frac{1}{2}} \left(\sum_{(k-1)/2 \text{ even}} (\gamma_k^+ \omega^{(k-1)})^2 + \right.$$

$$+ \sum_{(k-1)/2_{\text{odd}}} \left(\gamma_k^- \omega^{(k-1)} \right)^2 \right)^{\frac{1}{2}}. \quad (23)$$

Substituting (17) into (21), (23) and (12) one obtains

$$\left(\frac{|\Delta h(\omega)|}{S^+(\omega)} \right)^2 + \left(\frac{|\Delta g(\omega)|}{T^+(\omega)} \right)^2 \leq \sum_{k \text{ even}} |\mu_k|^2 + \sum_{k \text{ odd}} |\mu_k|^2 = \sum_{k=0}^n |\mu_k|^2 \leq \rho^2 \quad (24)$$

or equivalently

$$\left[\left(\frac{|\Delta h(\omega)|}{S^+(\omega)} \right)^2 + \left(\frac{|\Delta g(\omega)|}{T^+(\omega)} \right)^2 \right]^{\frac{1}{2}} \leq \rho. \quad (25)$$

It means that for $h_0(\omega) \leq 0$ and $g_0(\omega) \leq 0$ the origin is excluded from the value set of the polynomial (11) specified by (12) if and only if

$$\left[\left(\frac{h_0(\omega)}{S^+(\omega)} \right)^2 + \left(\frac{g_0(\omega)}{T^+(\omega)} \right)^2 \right]^{\frac{1}{2}} \geq \rho \quad (26)$$

or by other words if and only if frequency plot of $h_0(\omega)/S^+(\omega) + jg_0(\omega)/T^+(\omega)$ does not intersect left lower quarter of circle with radius ρ centered at origin $\mathcal{D}_3(\rho)$,

$$\mathcal{D}_3(\rho) := \left\{ (x, y) : x \leq 0, y \leq 0; [|x|^2 + |y|^2]^{\frac{1}{2}} \leq \rho \right\}. \quad (27)$$

This statement is equivalent to the condition 3b) for $h_0(\omega) \leq 0$ and $g_0(\omega) \leq 0$.

2. $\Delta h(\omega) \leq 0, \Delta g(\omega) \leq 0$:

For $\Delta h(\omega) \leq 0$ we have

$$\Delta h(\omega) \geq \sum_{k/2_{\text{even}}} \mu_k \gamma_k^- \omega^k - \sum_{k/2_{\text{odd}}} \mu_k \gamma_k^+ \omega^k \quad (28)$$

or equivalently for its absolute value

$$|\Delta h(\omega)| \leq \sum_{k/2_{\text{even}}} |\mu_k \gamma_k^- \omega^k| + \sum_{k/2_{\text{odd}}} |\mu_k \gamma_k^+ \omega^k|. \quad (29)$$

Using Hölder's inequality gives

$$|\Delta h(\omega)| \leq \left(\sum_{k \text{ even}} |\mu_k|^2 \right)^{\frac{1}{2}} \left(\sum_{k/2_{\text{even}}} (\gamma_k^- \omega^k)^2 + \sum_{k/2_{\text{odd}}} (\gamma_k^+ \omega^k)^2 \right)^{\frac{1}{2}}. \quad (30)$$

Analogically, for $\Delta g(\omega)$ we have

$$\Delta g(\omega) \geq \sum_{(k-1)/2_{\text{even}}} \mu_k \gamma_k^- \omega^{(k-1)} - \sum_{(k-1)/2_{\text{odd}}} \mu_k \gamma_k^+ \omega^{(k-1)} \quad (31)$$

and for the absolute value

$$|\Delta g(\omega)| \leq \left(\sum_{k \text{ odd}} |\mu_k|^2 \right)^{\frac{1}{2}} \left(\sum_{(k-1)/2 \text{ even}} \left(\gamma_k^- \omega^{(k-1)} \right)^2 + \sum_{(k-1)/2 \text{ odd}} \left(\gamma_k^+ \omega^{(k-1)} \right)^2 \right)^{\frac{1}{2}}. \quad (32)$$

Substitution of (17) into (30), (32) and (12) gives

$$\left(\frac{|\Delta h(\omega)|}{S^-(\omega)} \right)^2 + \left(\frac{|\Delta g(\omega)|}{T^-(\omega)} \right)^2 \leq \sum_{k \text{ even}} |\mu_k|^2 + \sum_{k \text{ odd}} |\mu_k|^2 = \sum_{k=0}^n |\mu_k|^2 \leq \rho^2 \quad (33)$$

or equivalently

$$\left[\left(\frac{|\Delta h(\omega)|}{S^-(\omega)} \right)^2 + \left(\frac{|\Delta g(\omega)|}{T^-(\omega)} \right)^2 \right]^{\frac{1}{2}} \leq \rho. \quad (34)$$

It means that for $h_0(\omega) \geq 0$ and $g_0(\omega) \geq 0$ the origin is excluded from the value set of the polynomial (11) specified by (12) if and only if

$$\left[\left(\frac{h_0(\omega)}{S^-(\omega)} \right)^2 + \left(\frac{g_0(\omega)}{T^-(\omega)} \right)^2 \right]^{\frac{1}{2}} \geq \rho \quad (35)$$

or by other words if and only if frequency plot of $h_0(\omega)/S^-(\omega) + jg_0(\omega)/T^-(\omega)$ does not intersect right upper quarter of circle with radius ρ centered at origin $\mathcal{D}_1(\rho)$,

$$\mathcal{D}_1(\rho) := \left\{ (x, y) : x \geq 0, y \geq 0; [|x|^2 + |y|^2]^{\frac{1}{2}} \leq \rho \right\}. \quad (36)$$

This statement is equivalent to the condition 3b) for $h_0(\omega) \geq 0$ and $g_0(\omega) \geq 0$.

Using similar reasoning one can state that for $h_0(\omega) \leq 0$ and $g_0(\omega) \geq 0$ the origin is excluded from the value set of the polynomial (11) if and only if frequency plot of $h_0(\omega)/S^+(\omega) + jg_0(\omega)/T^-(\omega)$ does not intersect left upper quarter of circle with radius ρ centered at origin $\mathcal{D}_2(\rho)$,

$$\mathcal{D}_2(\rho) := \left\{ (x, y) : x \leq 0, y \geq 0; [|x|^2 + |y|^2]^{\frac{1}{2}} \leq \rho \right\}, \quad (37)$$

and for $h_0(\omega) \geq 0$ and $g_0(\omega) \leq 0$ if and only if frequency plot of $h_0(\omega)/S^-(\omega) + jg_0(\omega)/T^+(\omega)$ does not intersect right lower quarter of circle with radius ρ centered at origin $\mathcal{D}_4(\rho)$,

$$\mathcal{D}_4(\rho) := \left\{ (x, y) : x \geq 0, y \leq 0; [|x|^2 + |y|^2]^{\frac{1}{2}} \leq \rho \right\}. \quad (38)$$

These statements complete the equivalence of the conditions 2b) and 3b).

The maximum ρ preserving stability of polynomial (11), $\bar{\rho}$, can be determined as maximum ρ satisfying all the conditions of theorem 3.

5 Example

Let the coefficients of a 6-th order polynomial $\tilde{p}(s) = \sum_{k=0}^6 \tilde{a}_k s^k$ be characterized by membership function (7) with the following parameters:

$$\begin{aligned}\mathbf{a}^0 &= [a_0^0, a_1^0, a_2^0, a_3^0, a_4^0, a_5^0, a_6^0]^T \\ &= [433.5, 667.25, 502.25, 251.25, 80.25, 14, 1]^T \\ \gamma^+ &= [\gamma_0^+, \gamma_1^+, \gamma_2^+, \gamma_3^+, \gamma_4^+, \gamma_5^+, \gamma_6^+]^T \\ &= [196.980, 143.480, 117.424, 45.948, 12.980, 8.560, 0.600]^T \\ \gamma^- &= [\gamma_0^-, \gamma_1^-, \gamma_2^-, \gamma_3^-, \gamma_4^-, \gamma_5^-, \gamma_6^-]^T \\ &= [173.400, 133.440, 100.548, 60.300, 22.470, 5.600, 0.400]^T.\end{aligned}$$

The nominal (1-cut) polynomial $p(s, 1) = 433.5 + 667.25s + 502.25s^2 + 251.25s^3 + 80.25s^4 + 14s^5 + s^6$ is stable. The task is to determine maximum confidence level α_{\min} preserving stability of $\tilde{p}(s)$. The frequency plot of $h_0(\omega)/S(\omega) + jg_0(\omega)/T(\omega)$ is depicted in Fig. 1. Maximum ρ satisfying all conditions of theorem 4, $\bar{\rho} = 0.6616$ and the minimum confidence level preserving stability of $\tilde{p}(s)$, $\alpha_{\min} = 1 - \bar{\rho} = 0.3384$.

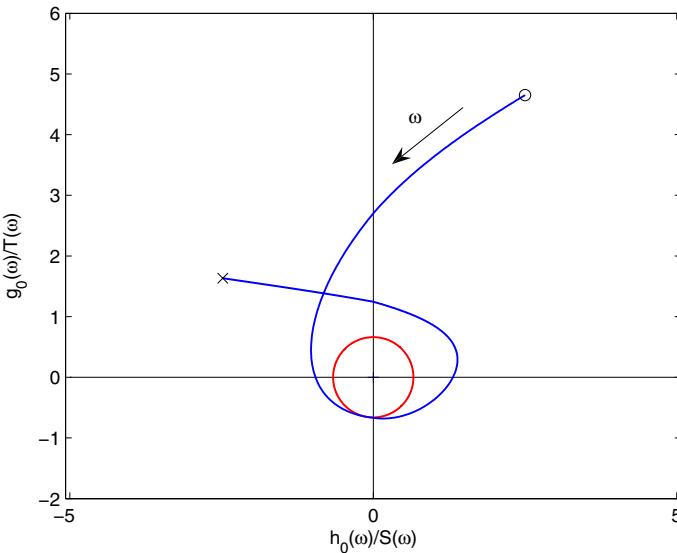


Fig. 1. Frequency plot of $h_0(\omega)/S(\omega) + jg_0(\omega)/T(\omega)$

6 Conclusion

In this paper an algorithm for determining minimum confidence level preserving stability of continuous-time linear systems with fuzzy parametric uncertainty

is presented. The coefficients of characteristic polynomial are supposed to be characterized by multivariate non-symmetric ellipsoidal membership functions. The algorithm is graphical in nature and is based on generalization of Tsyplkin-Polyak plot.

References

1. Argoun, M.B.: Frequency Domain Conditions for the Stability of Perturbed Polynomials. *IEEE Trans. Automat. Control* 32, 913–916 (1987)
2. Bhattacharyya, S.P., Chapellat, H., Keel, L.H.: Robust Control: The Parametric Approach. Prentice-Hall, Inc., Englewood Cliffs (1995)
3. Bombois, X.: Connecting Prediction Error Identification and Robust Control Analysis: a new framework. Ph.D thesis, Université Catholique de Louvain, Belgium (2000)
4. Bombois, X., Gevers, M., Scorletti, G., Anderson, B.D.O.: Robustness analysis tools for an uncertainty set obtained by prediction error identification. *Automatica* 37, 1629–1636 (2001)
5. Bondia, J., Picó, J.: Analysis of Systems with Variable Parametric Uncertainty Using Fuzzy Functions. In: Proc. of European Control Conference ECC 1999 (1999)
6. Bondia, J., Picó, J.: Analysis of Linear Systems with Fuzzy Parametric Uncertainty. *Fuzzy Sets and Systems* 135, 81–121 (2003)
7. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, Inc., London (1980)
8. Kharitonov, V.L.: Ob Asymptoticeskoj Ustojcivosti Polozenija Rovnovesija Semejstva Sistem Linejnych Differentialnych Uravnenij. *Differentialnye Uravnenija* 14, 2086–2088 (1978)
9. Lan, L.H.: Robust stability of fuzzy-parameter systems. *Automation and Remote Control* 66, 596–605 (2005)
10. Ljung, L.: System identification—theory for the user, 2nd edn. Prentice-Hall, Upper Saddle River (1999)
11. Nguyen, H.T., Kreinovich, V.: How stable is a fuzzy linear system. In: Proc. 3rd IEEE Conf. on Fuzzy Systems, pp. 1023–1027 (1994)
12. Söderström, T., Stoica, P.: System Identification. Prentice Hall, Inc., London (1989)
13. Tsyplkin, Y.Z., Polyak, B.T.: Frequency domain criteria for l_p -robust stability of continuous linear systems. *IEEE Trans. Automat. Control* 36, 1464–1469 (1991)

An Imperative Account of Actions

Victor Jauregui¹ and Son Bao Pham²

¹ National ICT Australia Ltd (NICTA), and
The School of Computer Science and Engineering
The University of New South Wales, Sydney, Australia
vicj@cse.unsw.edu.au
² College of Technology
Vietnam National University, Hanoi
sonpb@vnu.edu.vn

Abstract. This article reports on an investigation into an alternative semantics for actions which is based on modal logic, with an underlying computational theme, where actions are interpreted as computations in an abstract machine model of the world. The frame problem is addressed by reformulating and generalising minimal change principles to the principle of ‘Occam’s razor’—the intended interpretation of an action is given by the simplest computations which realise its direct effects.

1 Introduction

Broadly speaking, reasoning about action is concerned with describing how the world is affected when an ‘action’ is performed. Central to this account is the notion of a ‘world state’. For example, when an agent is confronted with a closed door, the action of ‘opening the door’ will realise a state in which ‘the door is open’, provided ‘the door is unlocked’.

As this example suggests, a *state* is a description of the world which is determined by its properties, or *fluents*, such as ‘the door is open’, ‘the television is on’, and so on. We will treat these properties abstractly, labelling them p_1, p_2, \dots, p_n (for some finite $n \geq 1$). Moreover, the state of the world changes when *actions* are performed. Actions induce state transitions—changes in the state of the world—by modifying the truth values of some of these properties.

Independence, Inertia, and the Frame Problem. Typically, in the worlds we model, the majority of properties are independent; i.e., changing one property does not affect another. We adopt the default assumption that, in the absence of information indicating otherwise, each property p_i is independent of each other distinct property p_j ($i \neq j$)—if no dependency is entailed then none will be assumed. For example, p_i might be the property ‘the door is open’, and p_j ‘the television is on’. The intuition is that, for instance, the action of ‘opening the door’—an action which has as a *direct effect* making the ‘the door is open’ property true—does not have direct bearing on whether the television is on or not. Shanahan [1] refers to this principle as the ‘commonsense law of inertia’.

As discussed by McCarthy and Hayes [2], we wish to describe an action only through its direct effects. In our earlier example, we might give the following description of the action of ‘opening the door’: “if ‘the agent is in front of the door’ and ‘the door is unlocked’, then after performing the action of ‘opening the door’, it will be the case that ‘the door will be open’. However, when we do this the *frame problem* emerges because by specifying only the direct effects we have supplied only a partial description of the resulting state. As a consequence, among the state transitions which realise the direct effect may be ones which are unintended; e.g., in the example above we have not explicitly stipulated, even though we intend that ‘the television remains turned off’.

Imperative Semantics. Our approach is based on the operation of modern programmable computers. Each property of our domain is represented by a variable which is typically assigned a distinct (independent) location in the computer’s memory. The values assumed by the entirety of these variables comprises the state of the machine which corresponds to the state of the world being modelled. Changes in the machine state are induced by executing instructions in some programming language. Consequently, actions are modelled/represented by programs. So for example, modern imperative programming languages have instructions for setting the values of variables. A typical assignment statement might be: $x \leftarrow 5$, which is an instruction intended to give the value 5 to the variable x . Moreover, it is implicit in the operation of the machine that all variables other than x remain unaffected by the execution of this instruction. That is, the language’s imperatives act locally on independent variables. If we want to effect a change in another variable y , say in order to give y the value 3, then we have to add a further instruction $y \leftarrow 3$. It is this feature which makes the *imperative programming* account a good one for reasoning about action, by forcing us to explicitly specify any additional intended effects.

Another contribution of this paper is to furnish a semantics of action based on the possible worlds semantics of modal logic. The possible worlds semantics of modal logics, we feel, gives a natural account in which to reason about action. This effort is a continuation of the theme adopted in an increasing number of approaches [3], [4], [5], [6], *et al.*

Paper Outline. The presentation of material in this paper is as follows. In section 2 we describe the operation of the underlying machine model which governs our action semantics. In section 3 we furnish a logic for reasoning about action. This logic incorporates both a binary conditional connective and a family of unary modal connectives, as in Dynamic logic [7], to reason about action. In section 4 we furnish a semantics for this logic, showing, in section 5, how it can be employed to represent various scenarios in the literature. There we conduct an analysis of our approach, comparing it in section 6 with related work in the literature. Finally, in section 7 we conclude with some further discussion.

2 Machine Principles

As outlined in the introduction, the basic idea which will be developed in this paper is to interpret actions as the computations of an abstract machine \mathbf{U} . The state of the machine \mathbf{U} is determined by a finite number of binary variables and the possible worlds adopted in the semantics correspond to these machine states: themselves mappings which assign a truth value to each variable. Under this interpretation actions naturally correspond to programs. Moreover, actions are identified with their ‘direct effects’, making them incompletely specified. In the presence of this partial description, to obtain the ‘intended interpretation’ of the action, those programs which, given an initial state, produce states which satisfy the action’s direct effect, but which are otherwise as simple—of shortest length—as possible, are considered.

As was mentioned, actions will be interpreted as programs governing the operation of an abstract machine \mathbf{U} with a finite number of binary ‘memory’ variables $P = \{p_0, p_1, \dots, p_n\}$. To make the correspondence clear these variables will coincide with those of the logical language described later. A *state*, or *configuration*, of the machine \mathbf{U} (also called a \mathbf{U} -state, or \mathbf{U} -configuration) is an assignment, $s : P \rightarrow \{0, 1\}$, to each of the variables, such that for $p \in P$, $s(p)$ denotes the value of p in state s . The set of all possible machine states of \mathbf{U} is denoted $\Sigma^{\mathbf{U}} = \{0, 1\}^P$.

The Programming Language \mathcal{P} . The language \mathcal{P} defined below describes the *programs* which govern the operation of \mathbf{U} . The intuition is that an action, based on a description of its direct effects, will correspond to certain programs in \mathcal{P} , comprising the action’s *imperative interpretation*.

The language \mathcal{P} is inductively defined as the smallest language such that:

- $\lambda \in \mathcal{P}$;
- the pair $(p, b) \in \mathcal{P}$, for every $p \in P$, $b \in \{0, 1\}$;
- if $\pi, \tau \in \mathcal{P}$ then $\pi; \tau \in \mathcal{P}$.

The *instructions*, or *primitive operations*, or sometimes just *primitives*, of \mathcal{P} are λ and the pairs (p, b) , for each $p \in P$ and $b \in \{0, 1\}$. λ denotes the *null-op* (the operation which does nothing), while (p, b) denotes the operation of setting variable p with the value b ; these will be defined formally later. The final clause provides a way, through the connective “;”, to compose programs together to form longer programs. Intuitively, $\pi; \tau$ is the program which executes π followed by τ . It follows from this definition that a program π , comprises a finite sequence of primitives: $\pi_0; \pi_1; \dots; \pi_n$. The *complexity* of program π is denoted $|\pi|$, and is defined inductively by: $|\lambda| = 0$, $|(p, b)| = 1$, and $|\pi; \tau| = |\pi| + |\tau|$.

The operation of the machine \mathbf{U} under a program $\pi \in \mathcal{P}$ is defined inductively as follows: execution of a program $\pi_0; \pi_1; \dots; \pi_n$, begins at step $i = 0$ executing instruction π_0 in the initial state s_0 . If instruction π_i is executed at step i then instruction π_{i+1} is executed at step $i + 1$. The program terminates after step n .

Transitions. The semantics of an instruction i is determined by how it transforms a \mathbf{U} -state s to produce the state s' , denoted: $s \xrightarrow{i} s'$. Using this notation, the operation of the instructions λ and (p, b) is given by:

$$s \xrightarrow{\lambda} s \quad \text{where } s'(q) = \begin{cases} b & \text{if } q = p; \\ s(q) & \text{otherwise.} \end{cases}$$

In words, λ has no effect (i.e., it performs the identity transformation), while the instruction (p, b) writes the value b to variable p (overwriting the previous value) and leaves all other variables unaffected.

3 A Logic of Actions

Our logic of actions is modal in nature, incorporating unary modalities $([\alpha])$ as employed in Dynamic Logic (see e.g., Goldblatt [7]), and a binary modal conditional (\rightsquigarrow) in additional to the familiar material conditional (\rightarrow) .

The unary modalities $[\alpha]$ employ ‘action terms’ α as seen in Dynamic Logic. A simple language of action terms \mathcal{A} , based on a set of primitive action symbols A , is defined inductively as the smallest language including A , such that if $\alpha, \beta \in \mathcal{L}$, then $\alpha; \beta \in \mathcal{A}$.

The complete language \mathcal{L} , which we employ to reason about actions, is defined inductively as the smallest language containing $P = \{p_1, p_2, \dots, p_n\}$ —a set of symbols denoting primitive propositions—and \perp , and closed under:

- if $\varphi \in \mathcal{L}$, and $\alpha \in \mathcal{P}$, then $[\alpha]\varphi \in \mathcal{L}$
- if $\varphi, \psi \in \mathcal{L}$, then $\varphi \rightarrow \psi, \varphi \rightsquigarrow \psi \in \mathcal{L}$

The expression $[\alpha]\varphi$ is to be read as: “after performing action α , φ holds”. In this way we can talk about the effects of an action, such as “after performing the ‘open the door’ action, it will be the case that the property ‘the door is open’ holds”. The expression $\varphi \rightsquigarrow \psi$ reads: “in the ‘selected transitions’ which bring about φ, ψ holds”. Identifying what are good choices of the ‘selected transitions’—selecting the ‘intended transitions’ from among the possible ones—is the key to our semantics. Note that as actions induce state transitions, the conditional \rightsquigarrow , under this semantics, may impact on the effects actions may have. We will return to this point later.

4 Imperative Semantics

Our semantic structures consist of a set of possible worlds Δ . For our purposes, each world represents a state of the machine U ; i.e., $\Delta \subseteq \Sigma^U$. Further, for each action term α we supply its *imperative interpretation*, which is achieved through a mapping $\rho : \mathcal{A} \times \Delta \rightarrow \mathcal{P}$, such that $\rho(\alpha, s)$ is the program which corresponds to action term α in state $s \in \Delta$.

Action Structures. An *action structure* is a structure $\mathcal{M} = (\Delta, f, \rho)$ such that, $\Delta \subseteq \Sigma^U$, and $f : 2^\Delta \times \Delta \rightarrow 2^\Delta$, with $f(A, s) \subseteq A$, for every $A \subseteq \Delta$, and $s \in \Delta$, and $\rho : \mathcal{A} \times \Delta \rightarrow \mathcal{P}$. The map f is a selection function, as is found in the standard semantics of conditional logics (see Chellas [8]), which maps a

proposition—in this case representing the direct effects of an action—at a world, to another proposition—the possible successors of the action. More about the role of f will be said subsequently. The map ρ takes an action term α and, in a state/context $s \in \Delta$, maps it to a program $\rho(\alpha, s) \in \mathcal{P}$, as discussed previously.

The truth of a formula φ at state $s \in \Delta$ in a μ -model is defined inductively:

$$\begin{aligned} s \models_{\mathcal{M}} p &\text{ iff } s(p) = 1 & (p \in P) \\ s \not\models_{\mathcal{M}} \perp & \\ s \models_{\mathcal{M}} \varphi \rightarrow \psi &\text{ iff } s \not\models_{\mathcal{M}} \varphi \quad \text{or} \quad s \models_{\mathcal{M}} \psi \\ s \models_{\mathcal{M}} \varphi \rightsquigarrow \psi &\text{ iff } f([\varphi], s) \subseteq [\psi] \\ s \models_{\mathcal{M}} [\alpha]\varphi &\text{ iff } \rho(\alpha, s)(s) \models_{\mathcal{M}} \varphi. \end{aligned}$$

where, for $\varphi \in \mathcal{L}$, $[\varphi] = \{s \in \Delta \mid s \models_{\mathcal{M}} \varphi\}$. The second-to-last last case says: $\varphi \rightsquigarrow \psi$ holds at s iff among the selected φ -states $f([\varphi], s)$, ψ holds. As we intimated earlier, of particular interest are the states produced by the transitions of simplest transformational complexity into the selection $f([\varphi], s)$.

The last case affirms that $[\alpha]\varphi$ holds at a state $s \in \Delta$ in \mathcal{M} iff the program that \mathcal{M} (through ρ) associates with α at s ($\pi = \rho(\alpha, s)$) takes s and transforms it into a state $t = \pi(s)$ which satisfies φ .

A further restriction on action structures is that the transition corresponding to action α must be one among those in the f -selection. Stated otherwise, if from $s \in \Delta$, action α brings about φ it can only do so in one of the f -sanctioned ways. So if $\pi = \rho(\alpha, s)$ is the program representing $\alpha \in \mathcal{A}$ at $s \in \Delta$, then, for $A \subseteq \Delta$, if $\pi(s) \in A$, then $\pi(s) \in f(A, s)$. In particular, if $\pi(s) \in [\varphi]$, then $\pi(s) \in f([\varphi], s)$. This means, for $\psi \in \mathcal{L}$, if $s \models_{\mathcal{M}} [\alpha]\varphi$ and $s \models_{\mathcal{M}} \varphi \rightsquigarrow \psi$, then $s \models_{\mathcal{M}} [\alpha]\psi$. So any action structure \mathcal{M} must satisfy:

$$\models_{\mathcal{M}} (\varphi \rightsquigarrow \psi) \rightarrow ([\alpha]\varphi \rightarrow [\alpha]\psi). \quad (1)$$

This will be of importance later, as by imposing constraints on allowed transitions, the effects of actions may be similarly constrained.

Moreover, the connective ‘;’ must be well behaved in the sense that the corresponding *standard models* criterion of dynamic logic must be adhered to:¹

$$\rho(\alpha; \beta, s) = \rho(\alpha, s); \rho(\beta, \rho(\alpha, s)(s)). \quad (2)$$

This simply states that, if α is mapped to program π in state s and β is mapped to program τ in state $\pi(s)$, then $\alpha; \beta$ must be mapped to $\pi; \tau$ in s . In this way, when the mapping $\rho(a, s)$ is determined for all atomic action identifiers $a \in A$, then $\rho(\alpha, s)$ is uniquely determined for all $\alpha \in \mathcal{A}$.

Imperative Models. An *imperative action model* of $\varphi \in \mathcal{L}$ is an action structure such that $|\rho(\alpha, s)|$ (the complexity of the program associated with α at s) is minimal over all action models of φ , for each $\alpha \in \mathcal{A}$, and $s \in \Delta$. If \mathcal{M} is an imperative action model of φ it is written $\models_{\mathcal{M}}^{\mathcal{I}} \varphi$. For $\Gamma \subseteq \mathcal{L}$, \mathcal{M} is an imperative model of Γ , denoted $\models_{\mathcal{M}}^{\mathcal{I}} \Gamma$, if it is an imperative model of each

¹ See e.g., Goldblatt [7] for details on standard models in dynamic logic.

formula in Γ . If φ is true in all imperative action models of every member of $\Gamma \subseteq \mathcal{L}$, then it is written $\Gamma \models^{\mathcal{I}} \varphi$. In this case φ is said to be an *imperative consequence* of Γ .

5 Assessment

In this section we give an assessment of our approach by testing it against representative examples which characterise important classes of problems/scenarios.

The Yale Shooting Problem. The Yale Shooting scenario consists of a turkey which we intend to kill by firing a loaded gun at it. Consider an initial state in which the turkey is alive but the gun is not loaded. The intention is that, after loading the gun and then shooting, the turkey should die (represented by it being not-alive). Along with the expected actions: ‘load the gun’, and ‘shoot’; a trivial action ‘wait’ is included during which nothing is expected to happen. Let this scenario be axiomatised by:

$$\Gamma = \{ [Lo]l, \quad l \rightarrow [Sh](\neg a \wedge \neg l) \}$$

describing the ‘load’ action (Lo) and the ‘shoot’ action (Sh), respectively. The ‘wait’ action (Wa) requires no axiomatisation, as it does nothing.

In the approach outlined here, after executing the action sequence ‘load’, ‘wait’, and then ‘shoot’, it would be expected that the turkey should die; i.e., from Γ it should be possible to infer $[Lo; Wa; Sh]\neg a$. The intuition is that the gun becomes loaded, and remains loaded after the ‘wait’ action. Finally, shooting the loaded gun kills the turkey.

However, ‘global’ minimisation of change will lead to counter-intuitive models. For example, the ‘intended model’ described above has the gun being loaded, remaining loaded after the ‘wait’ action, and then the turkey dying after the ‘shoot’ action (along with the gun becoming unloaded). This entails three fluent changes. However, if the gun mysteriously unloads itself during the ‘wait’ action, then the turkey no longer dies, nor does the gun discharge, after attempting the ‘shoot’ action. This ‘anomalous’ model entails only two fluent changes; making it equally, if not more preferred under the principle of minimal change; this was the observation made by Hanks and McDermott [9]. As will be shown below, the compositionality of imperative models (as implied by the ‘standard model’ property) eliminates this ambiguity, ruling out the anomalous model described above. More formally, it is shown that in the imperative action models of Γ , the formula $[Lo; Wa; Sh]\neg a$ holds.

In the imperative models of Γ we obtain, for ρ , the following:

$$\begin{aligned} \rho(Lo, s) &= \begin{cases} (l, 1) & \text{if } s(l) = 0; \\ \lambda & \text{otherwise,} \end{cases} & \rho(Sh, s) &= \begin{cases} (a, 0); (l, 0) & \text{if } s = al; \\ (l, 0) & \text{if } s = \bar{a}l; \\ \lambda & \text{otherwise.} \end{cases} \\ \rho(Wa, s) &= \lambda \quad \forall s \in \Delta; \end{aligned}$$

By the standard model property (2), the following obtains:

$$\rho(Lo; Wa; Sh, s) = \begin{cases} (l, 1); \lambda; (l, 0) & \text{if } s = \bar{al}; \\ (l, 1); \lambda; (a, 0); (l, 0) & \text{if } s = al; \\ \lambda; \lambda; (l, 0) & \text{if } s = \bar{al}; \\ \lambda; \lambda; (a, 0); (l, 0) & \text{if } s = al. \end{cases}$$

For each of these interpretations it can readily be observed, by inspection, that, in the given state s , the corresponding program π (e.g., $\pi = (l, 1); \lambda; (a, 0); (l, 0)$ in $s = \bar{al}$) yields $\pi(s) \models_M \neg a$. Consequently, for every state $s \in \Delta$, $s \models_M [Lo; Wa; Sh]\neg a$, and hence $\Gamma \models^T [Lo; Wa; Sh]\neg a$.

One important thing to note is that the interpretation of the wait action ‘ Wa ’ at each $s \in \Delta$ is the null program. That is, when it is consistent to do so, an action will be interpreted by the identity transition. In this respect, the approach here resembles that of Baker [10] in the situation calculus, in which change is minimised by allowing the *Result* function to vary under the application of circumscription. Because the identity map between worlds/situations always produces no fluent changes, Baker’s approach, like the one here, will produce the identity transition whenever possible.

Circuits Example. Consider an electrical circuit containing two switches (sw_1 and sw_2) connected in series to a power source (possibly a battery) and a light (l): $P = \{sw_1, sw_2, l\}$. The only action considered is “closing switch sw_1 ”, denoted C_1 : $A = \{C_1\}$. A simple axiomatisation (albeit a problematic one) is the following:

$$\Gamma = \{ [C_1]sw_1, \quad sw_1 \wedge sw_2 \leftrightarrow l \}.$$

It would be desirable to be able to infer that if the second switch is closed, then closing the first leads to the light coming on; i.e., to obtain $\Gamma \models^T sw_2 \rightarrow [C_1]l$: for example, in the state $s = \bar{sw}_1, sw_2, \bar{l}$, in which the first switch is off, the second is on, and, therefore, the light is off, if the first switch is closed (switched on) then it should be inferred that the light comes on.

Note that in an imperative model of Γ , with $\Delta = [sw_1 \wedge sw_2 \leftrightarrow l]$, for $s \in \Delta$, then, as Γ imposes no restriction on f (i.e., $f(A, s) = A$, for each $A \subseteq \Delta$), it follows that $f(\llbracket sw_1 \rrbracket, s) = \llbracket sw_1 \rrbracket = \{sw_1 \bar{sw}_2 \bar{l}, sw_1 sw_2 l\}$. The following specification of ρ determines the imperative model which is the ‘intended’ model, denoted \mathcal{M} , of Γ :

$$\rho(C_1, s) = \begin{cases} \lambda & \text{if } s(sw_1) = 1; \\ (sw_1, 1) & \text{if } s(sw_2) = 0; \\ (sw_1, 1); (l, 1) & \text{otherwise.} \end{cases}$$

This model satisfies $\models_M^T sw_2 \rightarrow [C_1]l$.

However, it can be shown that there is an anomalous/unintended imperative model \mathcal{M}' of Γ with:

$$\rho'(C_1, s) = \begin{cases} \lambda & \text{if } s(sw_1) = 1; \\ (sw_1, 1) & \text{if } s(sw_2) = 0; \\ (sw_1, 1); (sw_2, 0) & \text{otherwise.} \end{cases}$$

The latter model entails that sw_2 mysteriously opens when sw_1 is closed, and, hence, that $\not\models_{\mathcal{M}'}^{\mathcal{T}} sw_2 \rightarrow [C_1]l$; making it the case that $\Gamma \not\models^{\mathcal{T}} sw_2 \rightarrow [C_1]l$.

The problem with the latter model—the reason it conflicts with intuition—is that, given background knowledge of the domain, while it is natural to associate a dependency between sw_1 and l , the switches sw_1 and sw_2 are presumed to be independent of each other; making it counterintuitive for sw_2 to change as a result of affecting sw_1 . This problem can be resolved if this extra dependency is encoded in the domain description Γ ; the connective \rightsquigarrow supplies a way to achieve this.

Causal Dependency. The expression $sw_2 \rightarrow (sw_1 \rightsquigarrow l)$ says that, when the second switch is closed, transitions which affect the first switch, will affect the light—implicitly doing so in preference to affecting any other switch/fluent. More specifically, if the second switch is on when the first is closed, then the light will come on. In effect this expression encodes the ‘influence’ relation which appears in Thielscher [11] or the ‘dependency’ relation of Jauregui [12]. Similar information, appears in the form of *fluent frames* in Lifschitz [13], and Giordano and Schwind [6], *et al.*

Proceeding with the former, adding it to Γ gives:

$$\Gamma' = \{ [C_1]sw_1, \quad sw_1 \wedge sw_2 \leftrightarrow l, \quad sw_2 \rightarrow (sw_1 \rightsquigarrow l) \}.$$

Let \mathcal{N} be an imperative model of Γ' . From (1), $\models_{\mathcal{N}} (sw_1 \rightsquigarrow l) \rightarrow ([C_1]sw_1 \rightarrow [C_1]l)$. But $\models_{\mathcal{N}} sw_2 \rightarrow (sw_1 \rightsquigarrow l) \in \Gamma'$. Hence $\models_{\mathcal{N}} sw_2 \rightarrow ([C_1]sw_1 \rightarrow [C_1]l)$. As $\models_{\mathcal{N}} [C_1]sw_1 \in \Gamma'$, then $\models_{\mathcal{N}} sw_2 \rightarrow [C_1]l$.

It is not hard to see that the ‘intended’ model \mathcal{M} described earlier is an imperative model of Γ' . Moreover, the anomalous model \mathcal{M}' above is not a model of Γ' at $s = \overline{sw_1}sw_2\overline{l}$ —as it yields program π' such that $\pi'(s) \in \llbracket -l \rrbracket$. But the domain axioms $[C_1]sw_1$, and $sw_2 \rightarrow (sw_1 \rightsquigarrow l)$ would require $\pi'(s) \in f(\llbracket sw_1 \rrbracket, s) \subseteq \llbracket l \rrbracket$, in contradiction. So via $sw_2 \rightarrow (sw_1 \rightsquigarrow l)$, the new axiomatisation Γ' eliminates the anomalous model.

Conditionals and Commonsense Inference. The example above shows that the connective \rightsquigarrow can, at least in certain respects, fulfil the role of a ‘causal conditional’. The anomalous model has been regarded ‘anomalous’ on account that it goes against the intended ‘causal direction’. Claims for the need to incorporate causal information about a domain have been made by e.g., Lin [14], Thielscher [11], McCain and Turner [15], *et al.*. The conditional \rightsquigarrow allows such ‘causal directions’ to be encoded in the domain description. A causal conditional was missing in the framework of Jauregui *et al.* [12]. As demonstrated in the example above, the introduction of a conditional here extends that approach by allowing the meta-logical ‘dependency relation’ which featured in Jauregui [12] to be encoded into the language.

Indeed a large part of the motivation for the framework above was the intuition that the notions which were formulated by Solomonoff could help define ‘commonsense inference’ in the sense that they are the simplest inferences consistent with the given information. This makes more precise the designs of this

thesis: generalising the principle of minimal change to that of Occam’s razor. The U-models presented here represent an attempt to formulate the latter so that it coincides with the principle of minimal change. Other formulations remain as future work.

6 Related Work

The approach in this paper shares some similarities with the work of Zhang and Foo [5] who extend PDL for reasoning about actions. In view of the connection between conditional logic and ‘propositionally indexed modalities’ (Chellas [8]) the approach of Zhang and Foo can be seen to furnish a ‘causal conditional’ which plays a role similar to the one we have employed here.

Giordano and Schwind [6] also propose a causal conditional for reasoning about action. Their semantics differs with the one here in that it is based on the classification of ‘frame fluents’, in the tradition of Lifschitz [13]. Furthermore, they argue against certain principles which hold of ours, such as *ID*: $\varphi \rightsquigarrow \varphi$, arguing that causation is not reflexive. The choice of whether or not to include such properties as *ID* is a matter of semantics; so while it is not appropriate in theirs, it is valid given the approach presented here.

Without being able to go into great detail, a number of other approaches in reasoning about action deserve mention. McCain and Turner [15] give a conditional account of causal reasoning. Actions do not appear explicitly in their approach so things like sequences of actions (e.g., the Hanks and McDermott problem) cannot be modelled in their approach. Thielscher [11] adopts an operational approach to the problem of causally determining indirect effects. The imperative semantics presented here accords well with this perspective. The work of Lin [14] similarly employs a causal perspective to address the ramification problem in the situation calculus. Lin’s approach to the frame problem is to add a *Caused* predicate and to circumscribe it, so that any fluent that is not explicitly caused to assume a value is implied to retain its former value.

7 Discussion and Conclusions

In this paper we presented a system for reasoning about actions with the intention that a natural account for incomplete action descriptions is provided by the semantics of imperatives; programs like those of typical imperative programming languages. One of the central themes that this paper addresses is the view that the most plausible commonsense inferences an agent makes are often the *simplest* feasible conclusions. Although this paper only takes a small step in this direction, this involves a departure from minimal change principles to instead embrace the principle of Occam’s razor.

This paper extends the work of Jauregui [12] by giving an account of indirect effects through the conditional \rightsquigarrow which was absent in [12]. This allows us to address the ramification problem within the language as we saw in section 5.

Some immediate avenues for future research are to extend the action language \mathcal{A} to incorporate other connectives like choice and iteration. This would make the departure into the principle of Occam's razor much more pronounced, and would conceivably give some interesting perspectives on how we might reason about regular change, and other ways in which the world might evolve, besides inertially. Further work is required to construct an adequate proof system to go with the semantics just presented.

References

1. Shanahan, M.: Solving the frame problem. MIT Press, Cambridge (1997)
2. McCarthy, J., Hayes, P.: Some philosophical problems from the standpoint of artificial intelligence. *Machine intelligence* 4, 463–502 (1969)
3. Prendinger, H., Schurz, G.: Reasoning about action and change: A dynamic logic approach. *Journal of Logic, Language, and Information* 5(2), 209–245 (1996)
4. Castilho, M., Gasquet, O., Herzig, A.: Formalizing action and change in modal logic i: the frame problem. *Journal of Logic and Computation* 5(9), 701–735 (1999)
5. Zhang, D., Foo, N.: EPDL: a logic for causal reasoning. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence, IJCAI 2001, pp. 131–138 (2001)
6. Giordano, L., Schwind, C.: Conditional logic of actions and causation. *Artificial intelligence* 157, 239–279 (2004)
7. Goldblatt, R.: Logics of time and computation, 2nd edn. CSLI (1992)
8. Chellas, B.F.: Basic conditional logic. *Journal of philosophical logic* 4, 133–153 (1975)
9. Hanks, S., McDermott, D.: Nonmonotonic logic and temporal projection. *Artificial intelligence* 33, 379–412 (1987)
10. Baker, A.: Nonmonotonic reasoning in the framework of the situation calculus. *Artificial intelligence* 49, 5–23 (1991)
11. Thielscher, M.: Ramification and causality. *Artificial intelligence* 87, 317–364 (1997)
12. Jauregui, V.: Semantical considerations for a logic of actions: an imperative manifesto. In: Proceedings of the 10th international conference on principles of knowledge representation and reasoning, KR 2006 (2006)
13. Lifschitz, V.: Frames in the space of situations. *Artificial intelligence* 46, 365–376 (1990)
14. Lin, F.: Embracing causality in specifying the indirect effects of actions. In: Proceedings of IJCAI 1995, pp. 1995–1991 (1995)
15. McCain, N., Turner, H.: A causal theory of ramifications and qualifications. In: Proceedings of IJCAI 1995, pp. 1978–1984 (1995)

Natural Language Interface Construction Using Semantic Grammars

Anh Kim Nguyen and Huong Thanh Le

Faculty of Information Technology, Hanoi University of Technology, Vietnam
{anhnk-fit, huonglt-fit}@mail.hut.edu.vn

Abstract. This paper is a study on constructing a natural language interface to relational databases, which accepts natural language questions as inputs and generates textual responses. The question is translated into a SQL query using a semantic grammar and then, a database management system is left to find the result table with its own specialized optimization and planning techniques. The textual responses are generated from the result table based on another semantic grammar and the query type. Experimental results show that this approach can analyze a wide range of questions with high accuracy and produce reasonable textual responses.

Keywords: natural language interface, database, semantic grammar.

1 Introduction

Database management systems (DBMSs) have been widely used thanks to their efficiency in storing and retrieving data. However, databases are often hard to use since their interface is quite rigid in cooperating with users. To get information from a database, the user has to fill some search criteria into a predefined form and receive results as a table or a fixed report. Such an interacting method is inconvenient for users who do not know the structure of the database being used. Using natural language to interact with a database is a better choice for users, especially non-expert ones.

The research on natural language interface to databases (NLI2DBs) has recently received attention from the research communities ([1], [4], [6]). The purpose of natural language interfaces is to allow users to compose questions in natural language and to receive responses under the form of tables or short answers. Since natural language always contains ambiguities, most NLI2DBs are implemented in a specific domain and can only understand a subset of a natural language.

This paper presents our approach to the NLI2DB. Our implemented system includes two main modules: (i) a Query Translator (QTRAN) to translate a natural language question to an SQL query; and (ii) a Text Generator (TGEN) to generate textual responses from a query result table. QTRAN uses a restricted, domain independent semantic grammar and a CYK parsing algorithm to translate user questions into SQL queries. TGEN produces answers from query result tables based on query types and a grammar that combines a template-based approach with a phrase-based one. To test the feasibility of the system, a specific DBMS - a student management database in Vietnamese – is used. The proposed system architecture guarantees its portability across domains.

The remaining sections of this paper are organized as follows. Section 2 introduces our technique to translate natural language questions to SQL queries. Section 3 describes our method to generate textual responses from a query result table. Our experimental results are discussed in Section 4. Finally, Section 5 concludes the paper and proposes possible future work on this approach.

2 Query Translator

Most existing NLI2DBs are quite rigid in translating natural language queries. They just look for keywords in the sentence [9] or using some templates in analyzing the user's input [15]. Such approaches cannot deal with questions in unpredicted formats. Some NLI2DBs such as AVENTINUS [3] has some efforts in carrying out real semantic analysis. However, no significant success is achieved yet in this direction because of the polysemantic and ambiguity in natural language. Most NLI2DBs are domain-dependent, as they require predefined knowledge of the working domain in constructing templates or semantic rules ([2], [13]). In this paper, we represent our query translator named QTRAN - a module that uses a restricted, domain independent semantic grammar to translate Vietnamese questions to SQL queries. In this system, users can type questions without following any predefined template. This semantic grammar is introduced next.

2.1 The Semantic Grammar

In general, each natural language question uses a particular grammatical structure. Each position in the question is used for a specific purpose, such as storing an object asked by the user, an object attribute, a value, or a linkage between objects. By analyzing a large set of user questions, we found that although user questions are expressed in various ways, they always follow a specific organizing principle. For that reason, we defined a semantic grammar that contains a set of rules reflecting syntactic and semantic structures of user questions. This grammar allows us to analyze restricted, domain independent questions. The questions are in the following formats:

- (i) Questions start with an interrogative pronoun (e.g., “Ai_{Who}/Cái_{What}/.....?”)
- (ii) Questions start with an instructive verb (e.g., “Đưa ra_{show}/tìm_{find}/liệt kê_{list}/.....”)

Especially, our rule set is also capable of handling negative questions or questions with stress words (e.g., “ít nhất_{at least}”). These questions are hardly representable by a query language supported by a DBMS. This grammar is strong enough to cover a large number of questions that are often posted by users. It consists of 76 rules, using 46 non terminal symbols and 34 terminal ones. Some examples of our semantic rules are shown below:

1. <conditions> → <selection condition><conjunction><conditions>
2. <conditions> → <joint condition><conditions>
3. <joint condition> → <source> <negative> <SR>
4. <joint condition> → <SR><source>
5. <negative> → ‘chưa_{not yet}/không_{no}’
6. <source> → <quantity><entity><conditions>
7. <source> → <values>
8. <quantity> → <stress word><number>

To translate a user's question to an SQL query, three processes are carried out: (i) parsing the user's question using the semantic grammar; (ii) interpreting the syntactic-semantic tree of the user's question; and (iii) translating the standard tree to an SQL query. These processes are described in sections 2.2, 2.3 and 2.4. below.

2.2 Parsing a User's Question

First, the parser detects words and their semantic categories in the user's question using a word-category dictionary. Then, the parser derives syntactic-semantic trees from the input question using the semantic grammar introduced in Section 2.1. Since the proposed grammar is context free and is not in the Chomsky normal form, an improvement of the Cocke-Younger-Kasami (CYK) algorithm is used to parse the input question.

As our grammar is context free and domain independent, the number of generated trees may be large. Meaningless trees or trees those do not represent the user's intention should be removed from the output. To solve this problem, two actions are carried out by QTRAN. First, QTRAN determines all possible constraints from the input question. These constraints are the correspondence between attributes and entities, between attributes and values (e.g., a date should accompanies with the 'date' attribute), etc. Then, if the input question contains semantic ambiguities, QTRAN generates a multi-choice question asking the user to pick the closest one to his/her intention¹. The syntactic-semantic trees that fit with the user's intention are selected as the output for the parser. An example of the output tree generated by the parser is shown in Fig. 1.

2.3 Interpreting the Syntactic-Semantic Tree of the User's Question

The syntactic-semantic trees generated by the previous process satisfy all possible constraints in the database. However, the semantic relations are not explicitly and completely presented in the trees. This process deals with this problem. It analyzes semantic relations in these trees and transforms them into another format that is closer to the SQL syntax. The trees after being transformed are called standard trees. This process uses a data mapping dictionary that provides information about the database structure including entities, attributes, and semantic roles. For example, an attribute is described in this dictionary by the following information: (i) attribute's name in Vietnamese; (ii) attribute's name in the database; (iii) name of the entity containing this attribute. Information about a semantic role includes (i) role's name; (ii) entity1's name; (iii) entity2's name; and (iv) SQL query corresponding to this relation.

The interpreter clarifies semantic roles of nodes in the syntactic-semantic tree and analyzes relations among them using the entity relationship, the data mapping dictionary and attribute values in the database. If the question lacks of information, the interpreter will modify and complete the tree. These operations are fully described in [12].

After being modified and completed, the trees are standardized by pushing all selection conditions forward all joint conditions for each entity (*Source* or *Destination*). The standard tree of the syntactic-semantic tree in Fig. 1 is shown in Fig. 2. Finally,

¹ For a detailed description of this technique, see [12].

Fig. 1. The syntactic-semantic tree for the question “Tim các sinh viên học ít nhất 2 môn do giáo viên A dạy” (“Find all students who study at least 2 subjects taught by lecturer A”)

QTRAN rephrases the input question based on the standard tree and displays it to the user². If the rephrased question expresses exactly the user’s intention, (s)he will click a button forcing QTRAN to generate the SQL-query from the standard tree.

2.4 Translating to SQL

In this process, each *Source* node is considered as a subtree, in which *Entity* nodes and *Conditions* nodes are translated as sub-SQL queries. The translating process traverses an input tree in the bottom up direction. After the root node ‘*Query*’ has been traversed, a complete SQL is generated. It is clear that if translation rules for all structures of *Source* and *Query* nodes is provided, it is possible to translate any standard tree to an SQL query. Such a set of translation rules is proposed and used in QTRAN. Some of our translation rules are shown below:

1. $SL(S \text{ 'phù định}_{\text{negative}}' <SR> R) = \text{select } K_S \text{ from } S \text{ where } K_S \text{ not in (select } K_R \text{ from } R \text{ where } K_R \text{ in (select } K_R \text{ from } R \text{))}$

² For example, the rephrased question of the question in Fig. 1 is “Tim các sinh viên học ít nhất 2 môn được dạy bởi giáo viên có tên là A” (“Find all students who study at least 2 subjects taught by the lecturer whose name is A”).

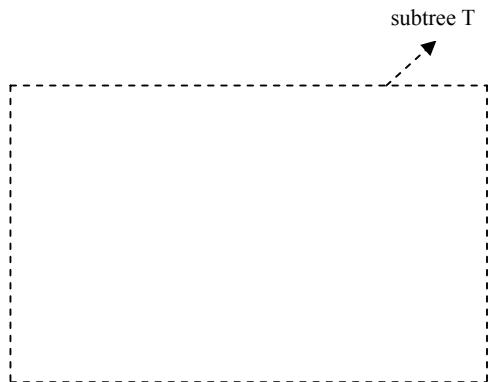


Fig. 2. The standard tree of the tree in Fig.1

2. $SL(S \text{ 'nhiều nhất}_{\text{at most}} n' \langle SR \rangle R) = \text{select } K_S \text{ from } S \text{ where } K_S \text{ not in (select } K_S \text{ from } \langle SR \rangle, R \text{ where } \langle SR \rangle.K_R = R.K_R \text{ group by } K_S \text{ having count}(K_R) > n)$
3. $SL(E \langle \text{selecting conditions} \rangle \langle \text{joint conditions} \rangle) = SL(SL(E \langle \text{selecting conditions} \rangle) \langle \text{joint condition 1} \rangle \text{ intersect } SL(SL(E \langle \text{selecting conditions} \rangle) \langle \text{joint condition 2} \rangle) \text{ intersect} \text{ intersect } SL(SL(E \langle \text{selecting conditions} \rangle) \langle \text{joint condition n} \rangle))$
 $\langle \text{joint conditions} \rangle = \langle \text{joint condition 1} \rangle \text{ 'and' } \dots \text{ 'and' } \langle \text{joint condition n} \rangle$, the joint conditions are translated by this rule set based on its structure in the tree).
4. $SL(E \langle \text{selecting conditions} \rangle) = \text{select } K_E \text{ from } E \text{ where } \langle \text{selecting condition 1} \rangle \text{ and } \langle \text{selecting condition 2} \rangle \text{ and and } \langle \text{selecting condition m} \rangle$
 $\langle \text{selecting conditions} \rangle = \langle \text{selecting condition 1} \rangle \text{ 'and' } \langle \text{selecting condition 2} \rangle \text{ 'and' } \dots \text{ 'and' } \langle \text{selecting condition m} \rangle$.

In the above rules:

- $SL(X)$ represents for a SeLect translation rule;
- $\langle SR \rangle$ is the entity2's name corresponding to a semantic role $\langle SR \rangle$ in the data mapping dictionary.
- K_S, K_R, K_E are keys of entities S, R , and E , respectively. (E can be the query's target).

Clause 1 [11]: The above rules do not modify the semantic meaning of structures in trees whose root node is 'Source' or 'Query'.

The following SQL query is generated by applying Rule 4 for the subtree T in Fig. 2:

```
SL(Lecturer lectname='A') =
SELECT lecturerID FROM Lecturer WHERE lectname='A'
```

The obtained SQL query after translating the standard tree in Fig. 2 is:

```
SELECT studname
FROM Student
WHERE studentID IN
    SELECT studentID
    FROM Study, (SELECT subjectID
                  FROM Teach
                  WHERE lecturerID IN
                      SELECT lecturerID FROM Lecturer
                      WHERE lectname='A') AS TP
    WHERE Study.subjectID = TP.subjectID
    GROUP BY studentID
    HAVING COUNT(subjectID) > 1
```

The “TP” in the above query is an abbreviation of the word “*temporary*”, which represents for a temporary result table.

3 Text Generator

When we ask a question, we expect to receive a meaningful and informative answer in reply. The traditional feedback of a DBMS, which is a result table or a report in a fixed format, is quite hard for naïve-users to understand, as they require some understanding of the database structure being used. In a NLI2DB, the result table is translated into a textual response in order to make the system easier to be used. The textual response should answer correctly the user’s question. In addition, it should be understandable, brief and fluency.

Approaches to text generation can be divided into four directions. The simplest approach, canned text systems, is too rigid as it simply prints available phrases storing in the system. A more sophisticated approach is the template-based one ([8], [16]), which is used mainly for multisentence generation. This approach uses predefined form that is filled by the information specified by either the user or the application at run-time. The phrase-based systems ([10], [14]) define a set of generalized templates that represents various kinds of phrases in a natural language (e.g., a verb phrase). Such systems start with a word or topic and create phrases based on the structure of a sentence. This approach is mainly used for single-sentence generation. The most sophisticated generators are feature-based systems ([5], [7]). However, only one sentence is outputted by most of these systems. In feature-based systems, each sentence is specified by a unique set of features. Generation proceeds by the incremental collection of features appropriate for each portion of the input until the sentence is fully determined.

Our system aims at generating text that can have more than one sentence. We propose an approach that combines the template-based approach with the phrase-based approach. In order to generate understandable, brief and fluent responses, beside a set of semantic rules, TGEN also needs knowledge sources to understand data meanings and data relations in result tables. The set of generation rules of TGEN is described next.

3.1 The Grammar Used in TGEN

Since people have preconceived ideas about the ways in which information can be integrated to form a text, we defined an organizing principle for text and used it to structure the information that will be included in the answer. We identified all question types which are translatable to SQL by QTRAN and the corresponding answer structures. The questions have been categorized into three types:

1. *Single_value* questions (e.g., *Who is the leader of the class BK20 in the academic year 2004-2005?*³).
2. *List* questions (e.g., *Which subjects did the class BK20 study in 2007-2008?*)
3. *Description* questions (e.g., *Give us information about the student Pham Thanh of the class BK20.*)

Each question type associates with one or several answer structures, which are formalized as frames in our approach. These frames can produce flexible text by using generation rules in a context free grammar. The generation rules describe semantic relations among values in the result table. A subset of the generation rules that is used to generate textual responses for a *Description* question is shown below:

- (1) [frame_Description_student] → [studname] ([sex]) [vp_studentID]. [studname] [vp_DOB], [vp_cityborn]. [studname] [vp_classID].
- (2) [frame_Description_student] → [studname] ([sex]) [vp_studentID]. [studname] [vp_classID]. [studname] [vp_DOB], [vp_cityborn].
- (3) [vp_classID] → is a student of the class [classID]
- (4) [vp_classID] → studies in the class [classID]
- ...

The strings in the square brackets (*[]*) are considered as non-terminal symbols; whereas the string that are not in the square brackets are terminal ones. Non-terminal symbols will be expanded by other generation rules or will be replaced by a value in the query result. During the generation process, if some slots in the frame do not have values to fill in, these slots will be removed from the frame.

The rules whose left hand side (LHS) starts with *[frame_* define the structure of a frame. The non-terminal symbols starting with *[vp_*, *[np_* or *[s_* represent for a verb phrase, a noun phrase or a sentence, respectively. The symbols in the square bracket and do not start with *[frame_*, *[vp_*, *[np_* and *[s_* are pre-terminal symbols. They will be replaced by values in the result table.

The above *Description* frame reflexes relations among attributes of the *Student* entity. In order to keep the generality of the frame set, we design *Description* frames for each entity of the database. Each frame consists of all attributes of an entity. If an answer describes relations among attribute values of several entities, the system will automatically generate the answer by connecting the *Description* frames of these entities through the entity's key. The system selects only frame's parts that relate to attributes of the result table. To make the textual outputs understandable, the entities' keys are often replaced by their corresponding name attributes during the generating process (e.g, *[studentID]* is replaced by *[studname]*).

³ For the convenient, examples from Section 3 were translated to English.

3.2 Generating Textual Responses

In order to generate text, four main steps are carried out in TGEN: (1) selecting candidate frames; (2) filling in frame slots; (3) correcting grammatical errors; and (4) generating answers. These four steps are carried out by our four modules: a Frame Selector, a Slot Filler, a Syntactic Refiner, and an Answer Generator. These steps will be described in detailed below.

3.2.1 The Frame Selector

To select appropriate frames, three factors are being considered: keywords in the user's question (e.g., *List*), the SQL query generated by QTRAN, and the shape of the result table. If a result table has multiple rows and columns, this table will be displayed to users since such tables are more condensed and informative than text. In other cases, QTRAN will generate a short textual response as casual users prefer a natural conversation than just looking at tables. The shape of the result table decides the frame type by the following policy:

- If the result table has only one value, the *Single_value* frame is selected.
- If the result table has several columns and one row, the *Description* frame is chosen. If the result table is a join among several entities, a join of the corresponding *Description* frames will be established.
- If the result table has one column and several rows, the *List* frame is chosen.

3.2.2 The Slot Filler

After frames have been selected, they are deployed by a top-down generating algorithm using the rule set described in Section 3.1. This process uses the data mapping dictionary mentioned in Section 2.3 to map values in the result table with pre-terminal symbols in the rules. For example, the table column *Student name* is mapped with the attribute *studname* in the data mapping dictionary. As a result, values in this column are filled in slots [*studname*] of the considering frames.

The slots that do not have values to fill in will be removed from the frame. This action may cause sentential fragments (e.g., a sentence without a verb phrase) in the output texts. For that reason, TGEN needs a Syntactic Refiner to solve this problem. The Syntactic Refiner is introduced next.

3.2.3 The Syntactic Refiner

The purpose of the Syntactic Refiner is to produce grammatical sentences from the outputs of the Slot Filler. It first parses the outputs of the Slot Filler to detect ungrammatical sentences. In order to do that, it locates positions of NPs and VPs in the Slot Filler's outputs by tracing the applied generation rules. Then, it checks the syntax of sentences given their NPs and VPs. If a sentence lacks a major part such as an NP or a VP, the Syntactic Refiner will combine it with its adjacent sentences.

3.2.4 The Answer Generator

Although the output texts of the Syntactic Refiner are grammatically correct, it may not be fluent. There are some reasons for this problem: the sentences can be too short or too long; some words are repeated several times; etc. The Answer Generator refines the texts so that they can be as natural as possible. It replaces repeated words by

their synonyms or reference words. A domain independent thesaurus, which stores semantic relations among words, is used in this process.

Given the user's question “*Give us the name, the class and the faculty of the student whose student number is 20050245*” and the query result in Table 1, one of the textual responses generated by TGEN is:

“*The name of the student is Pham Thanh. Pham Thanh is a student of the class BK20. This class belongs to the Information Technology faculty.*”

This text was produced by applying an automatically generated frame that connects three available frames [*frame_Student*], [*frame_Class*], and [*frame_Faculty*] through entities' keys.

Table 1. A Query Result Table

Student name	Class	Faculty
Pham Thanh	BK20	Information Technology

4 Experimental Results

To evaluate the system performance, we carried out experiments with a student management database. The evaluation focuses on two aspects: (i) the accuracy of rephrased questions from user questions; and (ii) the quality of textual responses. These aspects were evaluated independently by us.

20 users were asked to participate in our experiments. Their tasks were to formulate questions on this database without knowing the database structure and to evaluate system responses. Since these participants do not know the SQL syntax, they cannot evaluate directly SQL queries. Instead, they were asked to evaluate the rephrased questions derived in the second process of QTRAN (see Section 2.3). As mentioned in Clause 1 (Section 2.4), the SQL query does not modify the semantic meaning of structures in trees whose root node is ‘*Source*’ or ‘*Query*’. Therefore, we can guarantee that if the rephrased question is correct, the SQL query generated by QTRAN is also correct.

The experiments were conducted by testing questions with various length and with different characteristics: perfect or imperfect, positive or negative, quantified or unquantified. Each question was stated in several ways in order to test the robustness of the system. We got user feedbacks by asking them to answer the following questions:

- Do you think that the rephrased question is fluent (Y/N)?
- Do you think that the system rephrased correctly your intention (Y/N)?

The experimental results with QTRAN are shown in Table 2. 309 questions in total were used in the experiments. 91.91% rephrased questions are correct, among which 80.26% rephrased questions are fluent. It indicates that QTRAN is quite accurate and robust in translating user questions.

The SQL queries generated by QTRAN were used to query the database. Their result tables were used by TGEN to generate textual responses. In evaluating the output of TGEN, we categorized the user questions into three types: *Single_value*, *List* and *Description*. The *Description* questions were tested with two possibilities: (i) values

Table 2. Experimental Results with QTRAN

Question's characteristics			Rephrased questions			Total questions	
			Correct		Incorrect		
			Fluent	Unfluent			
Perfect	Positive	Quantified	29	2	1	32	
		Unquantified	66	1	1	68	
	Negative	Quantified	12	3	2	17	
		Unquantified	42	7	4	53	
Imperfect	Positive	Quantified	18	4	4	26	
		Unquantified	43	8	6	57	
	Negative	Quantified	9	3	3	15	
		Unquantified	29	8	4	41	
Total questions			248	36	25	309	
%			80.26	11.65	8.09		
%			91.91		8.09		

of the result table are retrieved from one entity; and (ii) values of the result table are retrieved from several entities. The textual responses were evaluated based on user satisfaction with the textual responses. Since QTRAN generates SQL queries corresponding to user's questions and TGEN generates textual responses from query result tables, the accuracy of information returned by the system is decided by QTRAN. Therefore, TGEN's performance was assessed by the following criteria:

- Do you think that the answer is syntactically correct (Y/N)?
- Do you think that the answer is fluent and clear (Y/N)?
- Do you think that the system produces flexible⁴ text (Y/N)?

Table 3 presents our experimental results with TGEN, using result tables returning from the SQL queries generated in the experiments with QTRAN. Since QTRAN only translates correct rephrased questions into SQL queries, 284 result tables were used as inputs in TGEN's experiments.

Table 3. Experimental Results with TGEN

Textual responses	% user satisfaction with question type			
	Single_value	List	Description (one entity)	Description (several entity)
syntactically correct	94.23	96.82	86.96	84.51
fluent and clear	91.35	93.65	69.57	74.65
flexible	92.3	88.89	84.78	85.92

Table 3 shows that our system is good at generating *Single_value* answers and *List* answers. The user satisfaction with *Description* answers is lower since generating text

⁴ "Flexible" means one result table can be expressed in different ways.

for such kind of answers is more complicated than the two other types. A *Description* answer often contains more than one sentence and covers a variety of attributes.

The user satisfaction on the flexibility of responses is rather high as several rules can be applied to generate text from a result table. The user satisfaction on the “*fluent and clear*” criterion is lowest among three evaluation criteria. To solve this problem, we need to investigate a method to improve the answer generating process of TGEN.

5 Conclusions and Future Work

In this paper, we introduced our approach to natural language interfaces. The system consists of two main modules: a Query Translator and a Text Generator. QTRAN uses a restricted, domain independent semantic grammar to translate user questions to SQL queries. TGEN uses generation rules to produce textual responses from query result tables. The current prototype of our system has achieved the following results:

- A friendly interface has been provided to users: users can type natural language questions that do not need to follow any predefined form and receive feedbacks under the form of short answers. It is suitable for people that have little or no knowledge of the database structure being used.
- The system accepts quantified questions and negative questions, which are very difficult to express in SQL syntax by naïve users.
- The system can assist users to rephrase questions correctly to his/her intention.
- Some questions (such as incomplete queries) can be automatically corrected without asking users to pick a choice.
- The system can produce flexible and grammatical textual responses. It proves that a hybrid approach to text generation is feasible for this kind of applications while a deep NLG approach is still far to be reached.
- The system is portable to other domains. When applying to other domains, we only need to modify the generation rule set and the dictionaries.

To improve the system performance, future work includes:

- Enriching the knowledge sources of the system to increase the system efficiency;
- Researching methods to improve the coherence and the fluency of output texts.

Acknowledgements. The author gratefully acknowledges the receipt of a grant from the Flemish Interuniversity Council for University Development Cooperation (VLIR UOS) which enabled the research team to carry out this work.

References

1. Androutsopoulos, I., Ritchie, G.D., Thanisch, P.: Natural language interfaces to databases - an introduction. *Natural Language Engineering* 2(1), 29–81 (1995)
2. Androutsopoulos, I.: Interfacing a Natural Language Front-End to Relational Database, Tech. Paper no.11, Dept.of AI, Univ. of Edinburg (1993)
3. AVENTINUS - Advanced Information System for Multinational Drug Enforcement, <http://www.dcs.shef.ac.uk/nlp/funded/aventinus.html>

4. ELF Software Co.: Access ELF: the amazing software that lets you communicate with Microsoft Access in plain English, <http://www.elfsoft.com/ns/prodserv.htm>
5. Elhadad, M.: Using Argumentation to Control Lexical Choice: A Functional Unification-Based Approach. Ph.D thesis, Computer Science Department, Columbia University (1992)
6. Hallett, C., Power, R., Scott, D.: Intuitive Querying of e-Health Data Repositories. In: 4th UK e-Science All Hands Meeting, Nottingham, UK (2005)
7. Mann, W.C.: An overview of the Penman text generation system. In: 3rd National Conference on Artificial Intelligence (AAAI 1983), Washington, DC, pp. 261–265 (1983)
8. McKeown, K.R.: Text generation: using discourse strategies and focus constraints to generate natural language text. Cambridge University Press, Cambridge (1985)
9. Meng, F., Chu, W.W.: Database Query Formation from Natural Language Using Semantic Modeling and Statistical Keyword Meaning Disambiguation. Technical Report CSD-TR 990003, University of California, Los Angeles (1999)
10. Meteer, M.W., McDonald, D.D., Anderson, S.D., Forster, D., Gay, L.S., Huettner, A.K., Sinun, P.: Mumble-86: Design and implementation. Technical report COINS 87-87. Computer and Information Science, University of Massachusetts at Amherst (1987)
11. Nguyen, K.A.: Translating natural language queries into SQL queries using semantic grammar (in Vietnamese). Journal of Computer Science and Cybernetics, Vietnam (2008)
12. Nguyen, K.A., Pham, T.T.H.: Imperfect natural language queries to relational databases (in Vietnamese). In: 20th Scientific Conference Hanoi University of Technology, Vietnam, pp. 117–122 (2006)
13. Owei, V.: Enriching the conceptual basis for query formulation through relationship semantics in databases. Inf. Syst. 26(6), 445–475 (2001)
14. Smith, G.: Computer and Human Language. Oxford University Press, Oxford (1991)
15. Stratica, N., Kosseim, L., Desai, B.C.: NLIDB Templates for Semantic Parsing. In: Applications of Natural Language to Databases (NLDB 2003), Germany, pp. 235–241 (2003)
16. Van Deemter, K., Theune, M., Krahmer, E.: Real vs. Template-Based Natural Language Generation: A false Opposition? Computational Linguistics 31(1) (2005)

Exploiting the Role of Named Entities in Query-Oriented Document Summarization

Wenjie Li¹, Furu Wei^{1,2}, Ouyang You¹, Qin Lu¹, and Yanxiang He²

¹ Department of Computing

The Hong Kong Polytechnic University, Hong Kong

{cswjli, csyouyang, csluqin}@comp.polyu.edu.hk

² Department of Computer Science and Technology

Wuhan University, China

{frwei, yxhe}@whu.edu.cn

Abstract. In this paper, we exploit the role of named entities in measuring document/query sentence relevance in query-oriented extractive summarization. Named entity driven associations are defined as the informative, semantic-sensitive text bi-grams consisting of at least one named entity or the semantic class of a named entity. They are extracted automatically according to seven pre-defined templates. Question types are also taken into consideration if they are available when dealing with query questions. To alleviate problems with low coverage, named entity based association and uni-gram models are integrated together to compensate each other in similarity calculation. Automatic ROUGE evaluations indicate that the proposed idea can produce a very good system that among the best-performing system at the DUC 2005.

Keywords: Query-Oriented Summarization, Named Entity based Association.

1 Introduction

In recent years, the focus has been noticeably shifted from generic summarization to query-oriented summarization, which aims to produce a summary from a set of relevant documents with respect to a given query, i.e. a short description of the user's information need containing one or more narrative and/or question sentences. As anticipated, the machine generated summaries should concisely describe information contained in the documents and also should facilitate the user in understanding documents according to his/her interests. The advantages of query-oriented summarization in information retrieval have been widely acknowledged. Brief summaries allow people to judge the relevance of the returned results without having to look through the whole documents.

Currently, most query-oriented summarization approaches are to extract the salient sentences from the documents which are supposed to be relevant to the given query. The fundamental issue with these approaches is how to measure the relevance of document sentences to the query sentences. In earlier studies, sentences are represented as bags of words. There are at least two drawbacks with this representation. First, the single word (i.e. word Uni-gram) is not informative enough to represent

underlying information in the sentences. For example, the meaning of the residence of US president would be lost when “White House” was represented by “White” and “House” separately. As a result, named entities, like other words, should be treated as meaningful text units when measuring relevance. Second, the ordering information, especially the semantic underlying information and the sentence structure can not be captured by Uni-gram models. N-gram, such as Bi-gram, model provides a mean to take into account of the shallow structural information by combining two text units. But meanwhile, any N-gram model will more or less suffer from the bottleneck of low coverage. That is why Uni-gram and Bi-gram models are normally combined in use, or constraints on Bi-gram models are relaxed.

In this study, we tend to highlight the role of named entities (NE) in variety of NE driven models. Named entities are regarded as text uni-grams and NE centered associations are defined as the informative and semantic-sensitive text bi-grams involving at least one named entity in representing sentences. Associations combine named entities, their semantic classes, as well as other representative words (adjacent to the named entities in certain models). Question types, which indicate what kind of information the questions are looking for, if applicable, are also concerned in associations when dealing with the questions in query. Because of this, NE-driven models can help effectively locate the sentences that contain the most relevant information to the questions. Consequently better summaries could be expected. Automatic ROUGE evaluations show that the summaries produced by the combinatorial models of NE/word uni-grams and NE-driven bi-grams are comparatively good with the summaries produced by the best systems competing at the DUC 2005.

2 Related Work

Query-oriented summarization has been boosted by DUC evaluations since 2005. Many previous approaches rank the sentences according to their relevance to the query and then select the most relevant ones into the summary. Regardless of the approaches taken, query-oriented summarization involves three basic aspects: text content representation; query formulation; and relevance judgment. Among them, how to estimate the relevance between query and sentences is the most fundamental issue, which has been extensively studied in the past.

The simplest yet effective way is to calculate the cosine similarity of the two sentences represented by the vectors of the words [7, 13, 14]. Some related work also utilizes WordNet as the external resource to solve the word mismatch problem by calculating the semantic similarity between the words. An extension to vector space models is dimension reduction performed with latent semantic analysis [5]. In addition to various kinds of word occurrence, frequency and semantic matching techniques, similarity can be also measured by the matching of the other text contents, such as named entities [8, 14], basic elements [6], and grammatical relations [3]. Normally, the relevance is judged based on a set of features, which are linearly combined to decide how a sentence is likely to be included in the summary. An alternative way is to construct a single but complicated feature, such as dependency tree [12] or document graph [4, 11]. It is however limited by the complexity of feature construction and relevance judgment.

Question answering (QA) is closely related to query-oriented summarization in terms of needs for question interpretation. Although question type identification [2, 8], question reformulation [3, 12] and question expansion [1] have been applied in the context of query-oriented summarization, special handling of query questions is not well concerned in many related work.

3 Measuring Relevance with Named Entity Driven Association Models

3.1 NE Driven Bi-gram Association (NeBiA) Model

In the NeBiA model, content associations are defined as the bi-grams involving at least one named entity or the semantic class of a named entity. They are the combinations of the named entities and the content representative words (i.e. non-stopwords) immediately adjacent to the named entities. All the associations fall into 4 categories and appear in one of the following forms:

Table 1. Templates for the Extraction of NE Driven Bi-grams

Category	Form
NE-NE	(NE_1, NE_2)
NE-WORD	$(NE, word), (word, NE)$
NE-TAG	$(NE_1, NE_2 tag), (NE_1 tag, NE_2)$
TAG-WORD	$(NE tag, word), (word, NE tag)$

Table 1 provides seven templates to guide the automatic extraction of NE centered bi-grams from both document sentences and query sentences so that the similarity can be calculated according to the bi-grams they match and the matching extent. Notice that NE-NE represents two successive named entities in a sentence. But they are not necessarily adjoining to each other and might be separated by a couple of words in-between. In fact, the NeBiA bi-grams defined in Table 1 are the selected subset of the text bi-grams, where the role of named entities is highlighted.

It is common for the same entity to be expressed in the different ways when it is mentioned in the text. For example, “US”, “U.S.”, “the US” and “the United States” all refer to the Unite States. Consequently, most of time, named entities fail to find their matches simply because of this. Coreference resolution can definitely provide a solution to this problem, but itself is also a problem being worked out in natural language processing. Our solution is to relax the matching restriction to allow for both named entities and their semantic classes being included in the bi-gram associations. The semantic classes considered in this paper include `<Person>`, `<Organization>`, `<Location>`, `<Date>`, and `<Number>`, which are called NE tags. Another advantage from the use of the NE tags is being able to integrate QA techniques into query-oriented summarization. This will be detailed in Section 3.3.

The NeBiA model can be extended to the NeBiA-II model to include all the words within the window of the sentence instead of the adjacent ones, i.e. extended from rigid to soft NE(TAG)-WORD bi-gram combinations.

3.2 NE Driven Event Bi-gram Association (NEvBiA) Model

As we know, named entities always play an important role in characterizing the events which can be defined as “[Who] did [What] to [Whom] [When] and [Where]”. The design of the NEvBiA model is based on the assumptions that if the words in the NeBiA model could be restricted to those related to the events, the bi-grams might be able to reflect the underling intra-event associations. In this paper, we choose verbs (such as “elect”) and action nouns (such as “supervision”) as event words that can characterize or partially characterize the actions or the incident occurrences in the world. They roughly relate to the “did [What]” mentioned above. Meanwhile, the named entities <Person>, <Organization>, <Location> and <Date> convey the information of [Who], [Whom], [Where] and [When], while [Number] complements other event descriptions, such as the extent. Clearly, the NEvBiA bi-grams are the selected subset of the NeBiA bi-grams, where the words are limited to the event words.

Similarly, the NEvBiA model can be extended to the NEvBiA-II model, corresponding to the NeBiA and NeBiA-II models.

3.3 Handling Query Questions

We strongly support the idea of incorporating QA techniques into query-oriented summarization. Thus, the models introduced in Section 3.1 and 3.2 are also designed to facilitate the formulation of both narrative and question sentences in query. For a query question, its question type is concerned and handled in the same way as the tags of the named entities presented in the sentence. Question type indicates what kind of information the question is looking for. It can help locate the sentences containing the information related to a particular question, and select the appropriate sentences in the summary. For example, if a sentence contains the named entity tagged as <Person> or <Organization>, it should be more likely to provide the answer to the question “Who has criticized the World Bank?”.

Figure 1 in the next page illustrates four categories of five NE driven bi-grams extracted from this question. Notice that the ordering information is reserved in them, i.e. $(NE, word) \neq (word, NE)$. This can avoid the mistakes in including the sentences containing the phrase “World Bank criticized <Person>” in the summary responding to the previous question. These sentences are obviously not expected.

Question types are determined by a set of heuristic rules. For the questions beginning with the interrogatives like “who”, “where”, and “when”, a straightforward mapping between these interrogatives and the classes of the named entity to be questioned is established. “who” \leftrightarrow <Person>, “where” \leftrightarrow <Location>, and “when” \leftrightarrow <Date>. If the sentence begins with “which”, “what” or the word “name”, the classes are deduced based on the semantics of the nouns in the patterns of “which + noun”, “what + noun”, “what be + noun”, and “name + noun”. WordNet supplies the semantic information needed. See (Li, 2005) for more details.

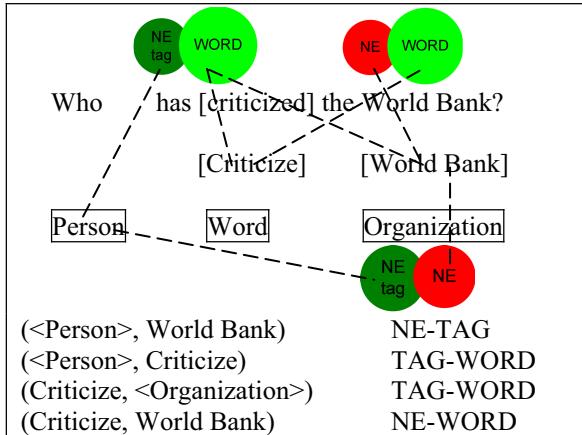


Fig. 1. Example of the 3 Categories of Bi-gram association

3.4 Matching-Based Relevance Measure

Sentence and query relevance are measured based on the words and the associations they match. In this study, we make an attempt on three matching strategies: (1) exact matching (EM); (2) semantic matching (SM); and degreeed matching (DM).

EM and SM are binary decisions. While EM returns binary 0 or 1 depending on whether a matching succeeds or fails, SM considers the hyponyms of the words and returns 1 when the two words (or the two words in the two associations under comparison) belong to the same synset in WordNet. This is motivated by the observation that some words of the same or quite similar meanings are in different surface forms. These words are commonly synonyms or hyponyms, such as “diminish” and “reduction”. The third strategy, i.e. DM, backs off EM with SM. It performs EM first. Only when EM fails, it gets back to SM, and returns a value smaller than 1 (e.g. 0.7) if SM succeeds. The relevance is then measured by calculating the similarity of the sentences and the query according to the frequencies of the matches. The matching strategies are applied not only to bi-gram association matching but also to uni-gram matching.

For the extracted bi-gram associations, once the matching is done, they naturally form a collection of n association groups, denoted by A . An association group contains either a set of associations matched or a single association if no match is found. The similarity of a sentence s^D in a document set D and a query $T = \{s_1^T, s_2^T, \dots, s_m^T\}$ is then calculated by cosine similarity based on frequencies of a_i

$$Sim_{bi}(s^D, T) = \frac{\sum_{i=1}^n tf(a_i, s^D) * tf(a_i, T)}{\sqrt{\sum_{i=1}^n (tf(a_i, s^D))^2} * \sqrt{\sum_{i=1}^n (tf(a_i, T))^2}} \quad tf(a_i, T) = \sum_{j=1}^m (tf(a_i, s_j^T))$$

where $a_i \in A$, $tf(a_i, *)$ is the frequency of a_i in s^D or T .

Associations provide important means for relational content matching. But it often suffers from low coverage. If the similarity is calculated solely based on association matching, an actual relevant sentence might be mistakenly judged as non-relevance. To remedy this shortage, the overall similarity is actually calculated by linearly combining the association model and the uni-gram model.

$$\text{Sim}(s^D, T) = \lambda_1 \text{Sim}_{uni}(s^D, T) + \lambda_2 \text{Sim}_{bi}(s^D, T)$$

where

$$\text{Sim}_{uni}(s^D, T) = \frac{\sum_{i=1}^n \text{tf}(u_i, s^D) * \text{tf}(u_i, T)}{\sqrt{\sum_{i=1}^n (\text{tf}(u_i, s^D))^2} * \sqrt{\sum_{i=1}^n (\text{tf}(u_i, T))^2}}. \text{ tf}(u_i, *) \text{ denotes the frequency of } u_i \text{ in } s^D \text{ and } T.$$

λ_1 and λ_2 are the weights for uni-gram and association based similarity respectively. Similarly,

$$\text{tf}(u_i, T) = \sum_{j=1}^m (\text{tf}(u_i, s_j^T))$$

4 Experimental Studies

4.1 Experiment Set-Up

The experiments are conducted on the DUC 2005 50 document sets. Each set of documents is given a query which simulates the user's information need. All documents and queries are pre-processed by TextPrepEngine, a text pre-processing engine developed upon GATE¹ and Porter Stemmer². Sentences can then be represented by a group of words which are stemmed, part of speech (POS) tagged, and the stop-word removed³. Moreover, named entities are tagged for each sentence. According to the task definitions, system generated summaries are strictly limited to the 250 English words in length. Based on the calculated similarities, we pick up the highest scored sentences from the original documents into the summary until the word limitation is reached. Duplicate sentences are prohibited.

Automatic evaluation methods and criteria are still a research topic in summarization community. Many literatures have addressed different methods for automatic evaluation other than human judges. Among them, the ROUGE toolkit⁴ [10], though being argued by quite a few researchers, is supposed to produce the most reliable and stable scores comparing with human evaluation. Moreover, the DUC 2005 officially adopts ROUGE as the automatic evaluation method. Therefore, we also take it as the evaluation means in this work. Specifically, the machine-generated summaries are evaluated in terms of average recalls of ROUGE-1, ROUGE-2, and ROUGE-SU4.

¹ <http://www.gate.ac.uk>

² <http://www.tartarus.org/~martin/PorterStemmer>

³ A list of 199 words is used to the filter stop words.

⁴ ROUGE 1.5.5 is used, and the ROUGE parameters are “-n 2 -x -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0”, according to the DUC task definition.

4.2 Evaluation of Uni-gram Models

The word-based uni-gram model is implemented as the baseline model. When the named entities in the text are recognized and manipulated as the integrated text units, we call it NE-based models. Table 2 below compares the NE-based uni-gram model with the word-based uni-gram model. However, the advantage is visible but not markedly. In later experiments, we will further evaluate the combinations of uni-gram and various bi-gram models.

Table 2. Evaluations of Uni-gram Models

	ROUGE-1	ROUGE-2	ROUGE-SU4
Word-based	0.35952	0.06932	0.12602
NE-based	0.36400	0.06988	0.12743

4.3 Evaluation of Uni-gram and Bi-gram Combinatorial Models

The aims of the following experiments are to examine the performance of various combinational models integrated with NE-based uni-gram and the bi-gram models and more important to discovery the most informative and representative text units. In the rigid approaches, the NeBiA and NEvBiA models have been described in Section 3. Bi-gram in the NeBi model is constrained to two adjoining words (non stop-words) according to their appearance in the sentence. This is the normal use of the bi-gram model. Differently, in the soft approaches, the two words within a given window size will be combined as a bi-gram in NeBi⁵-II, while in NeBiA-II and NEvBiA-II models, the NE-NE and NE-TAG bi-gram associations are constrained to two successive named entities (or tags), and a named entity (or tag) together with a word within the given window size⁶ will be combined to the NE-WORD or TAG-WORD associations. The numbers behind the soft models in the following table denote the best window sizes used in our experiments (tuned experimentally). In this set of experiments, the SM strategy is adopted and $\lambda_1 : \lambda_2 = 2 : 1$ is set experimentally.

Table 3. Results of Combinatorial Models

Rigid	ROUGE-1	ROUGE-2	ROUGE-SU4
+ NeBi	0.36169	0.07010	0.12620
+ NeBiA	0.36563	0.07345	0.12974
+ NEvBiA	0.36588	0.07336	0.12986
Soft	ROUGE-1	ROUGE-2	ROUGE-SU4
+ NeBi-II (7)	0.36201	0.07068	0.12648
+ NeBiA-II (6)	0.36663	0.07354	0.12987
+ NEvBiA-II (8)	0.36670	0.07357	0.13014

⁵ NeBi here denotes the original NE based bi-gram model.

⁶ Notice that the word within the given widow size to a named entity (or tag) can not cross another named entity (or tag).

Table 3 above presents the ROUGE results of the combinational models. We can see that the NeBi model improves the performance of the original word-based unigram slightly. And when text representative units are narrowed down gradually in NeBiA and NEvBiA models, the improvement becomes visible. Furthermore, more significant performance can be achieved when the soft models are taken into account,. The best performance is obtained by the NEvBiA-II model. These results strongly support the ideas of using NE-driven bi-gram associations in query-oriented summarization.

4.4 Coverage Problems with Single-Handed Bi-gram Models

As mentioned previously, the single-handed bi-gram based approaches, i.e. NeBi, NeBiA and NEvBiA models suffer from the coverage problem. For some set of documents in our experiments, we can even hardly find enough sentences, which can be considered relevant to the query, in order to produce a summary with the length close to the given 250 words limitations by solely using the bi-gram based measuring methods. The proportion of “x/50” in the each row of table 4 denotes that “x” out of the total 50 document sets are capable of producing the 250 word length summary.

Table 4. Results of Word based Models

	ROUGE-1	ROUGE-2	ROUGE-SU4
NeBi (34/50)	0.35272	0.06430	0.12061
NeBiA (7/50)	0.37521	0.07947	0.13268
NEvBiA (7/50)	0.37711	0.07973	0.13163

Obviously, the results in table 4 indicates the NeBiA and NEvBiA models can achieve quite encouraging and significant performance, but they are both limited by the low coverage. That's why normally bi-gram and uni-gram models are combined in use. It can be also observed that the performance of the bi-gram model NeBi is even worse than its corresponding uni-gram model. The sparse nature is the possible reason. This motivates us to restrict the bi-gram combinations in the proposed model.

4.5 Evaluations on Impacts of Matching Techniques

The following set of experiments aims to examine the three matching strategies, i.e. exact matching (EM), semantic matching and degreeed matching (DM). WordNet 2.0⁷ and JWNL⁸ are used to determine whether the two words are semantically matched according to whether they are in the same synset. In our implementation, DM will return 0.7 when the matching of two associations fails in EM but successes in SM. Table 5 shows the comparison results of the best-performing models, i.e. NEvBiA-II with 8 as the window size, in our former experiments.

⁷ <http://wordnet.princeton.edu/>

⁸ <http://sourceforge.net/projects/jwordnet>

Table 5. Comparison of EM/SM/DM strategies

	ROUGE-1	ROUGE-2	ROUGE-SU4
EM	0.36604	0.07340	0.13009
SM	0.36670	0.07357	0.13014
DM	0.36704	0.07360	0.13029

As seen, there exists improvement when semantic relation between two words is considered. However, the improvement is not quite obvious. This may due to the fact that the number of the named entity centered bi-gram associations involving with one word is still small in our current system, so that the contribution of the semantic relation is limited.

4.6 Comparison with DUC 2005 Top 3 Systems

The following table shows the comparison of our models with the DUC 2005 participating systems, where S15, S17 and S10 are the top three performing systems. As seen, both the NEvBiA-II and NeBiA-II models can achieve very competing performance. Although no further post-processing is carried out, the results of the NEvBiA-II model outperform the top system in the DUC 2005 in the ROUGE-2 evaluation, rank the second in ROUGE-SU4 evaluation and among the top three systems in the ROUGE-1 evaluation.

Table 6. Comparison with DUC 2005 top-3 systems

	ROUGE-1	ROUGE-2	ROUGE-SU4
NEvBiA-II	0.36704	0.07360	0.13029
NeBiA-II	0.36663	0.07354	0.12987
S15	0.37383	0.07251	0.13163
S17	0.36901	0.07174	0.12972
S10	0.36640	0.07089	0.12649
NIST Baseline	0.30217	0.04947	0.09788

6 Conclusion

In this paper, the role of named entity has been emphasized in query-oriented summarization. The effects of named entities in uni-gram and bi-gram models are investigated. ROUGE evaluation based on the DUC 05 data set shows that the proposed models can achieve very competitive and significant performance. The NE based uni-gram and NE driven bi-gram combinatorial model can even outperform the best system in the DUC 2005.

However, we also note that the use of name entities centered bi-gram associations is limited by the coverage problem, which can be improved by a more appropriate and wide-coverage named entity recognizer. Furthermore, since named entity co-reference is very useful in our investigation, co-reference resolution achievement in the natural language processing community will be further studied in the future work.

Acknowledgments

The work described in this paper was supported by the grants from the RGC of HK, (Project No. PolyU5211/05E and PolyU5217/07E), the grant from the NSF of China (Project No. 60703008), and the internal grant from the Hong Kong Polytechnic University (Project No. A-PA6L).

References

1. Barzilay, R., Lapata, M.: Modeling Local Coherence: An Entity-based Approach. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, pp. 141–148 (2005)
2. Conroy, J.M., Schlesinger, J.D.: CLASSY Query-Based Multi-Document Summarization. In: Proceedings of Document Understanding Conferences 2005 (2005)
3. Doran, W., Newman, E., Stokes, N., Dunnion, J., Carthy, J.: IIRG-UCD at DUC 2005. In: Proceedings of Document Understanding Conferences 2005 (2005)
4. Erakn, G.: Using Biased Random Walks for Focused Summarization. In: Proceedings of Document Understanding Conferences 2006 (2006)
5. Hachey, B., Murray, G., Reitter, D.: The Embra System at DUC 2005: Query-oriented Multi-document Summarization with a Very Large Latent Semantic Space. In: Proceedings of Document Understanding Conferences 2005 (2005)
6. Hovy, E., Lin, C.Y., Zhou, L.: A BE-based Multi-document Summarizer with Query Interpretation. In: Proceedings of Document Understanding Conferences 2005 (2005)
7. Jagarlamudi, J., Pingali, P., Varma, V.: Query Independent Sentence Scoring approach to DUC 2006. In: Proceedings of Document Understanding Conferences 2006 (2006)
8. Li, W., Li, W., Li, B., Chen, Q., Wu, M.: The Hong Kong Polytechnic University at DUC2005. In: Proceedings of Document Understanding Conferences 2005 (2005)
9. Li, W., Li, B., Wu, M.: Query Focus Guided Sentence Selection Strategy for DUC 2006. In: Proceedings of Document Understanding Conferences 2006 (2006)
10. Lin, C.Y., Hovy, E.: Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In: Proceedings of HLT-NAACL, pp. 71–78 (2003)
11. Mohamed, A.A., Rajasekaran, S.: Query-Based Summarization Based on Document Graphs. In: Proceedings of Document Understanding Conferences 2006 (2006)
12. Schilder, F., McCulloh, A., McInnes, B.T., Zhou, A.: TLR at DUC: Tree similarity. In: Proceedings of Document Understanding Conferences 2005 (2005)
13. Seki, Y., Eguchi, K., Kando, N., Aono, M.: Multi-Document Summarization with Subjectivity Analysis at DUC 2005. In: Proceedings of Document Understanding Conferences 2005 (2005)
14. Zhao, L., Huang, X., Wu, L.: Fudan University at DUC 2005. In: Proceedings of Document Understanding Conference 2005 (2005)

A Probabilistic Model for Understanding Composite Spoken Descriptions

Enes Makalic, Ingrid Zukerman, Michael Niemann, and Daniel Schmidt

Faculty of Information Technology, Monash University

Clayton, Victoria 3800, Australia

{enes, ingrid, niemann, dschmidt}@csse.monash.edu.au

Abstract. We describe a probabilistic reference disambiguation mechanism developed for a spoken dialogue system mounted on an autonomous robotic agent. Our mechanism receives as input referring expressions containing intrinsic features of individual concepts (lexical item, size and colour) and features involving more than one concept (ownership and location). It then performs probabilistic comparisons between the given features and features of objects in the domain, yielding a ranked list of candidate referents. Our evaluation shows high reference resolution accuracy across a range of spoken referring expressions.

1 Introduction

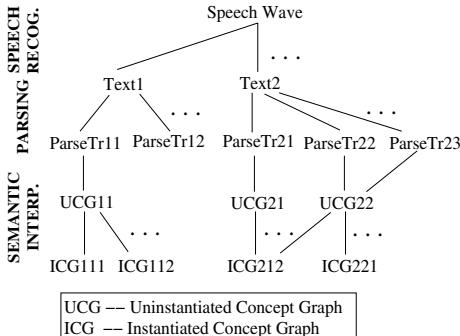
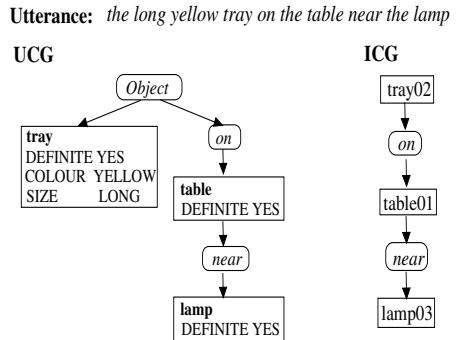
In this paper, we describe the reference disambiguation mechanism of *Scusi?* — the spoken language interpretation module of a robot-mounted dialogue agent. Our mechanism interprets referring expressions such as “the blue mug on the table near the lamp” by performing probabilistic comparisons between the requirements stated in a referring expression and the features of candidate objects (e.g., those in the room).

The contributions of our mechanism are (1) probabilistic procedures that perform feature comparisons; and (2) a function that combines the results of these comparisons. These contributions endow our mechanism with the ability to handle imprecise or ambiguous referring expressions. For instance, the expression “the bag near the green table” is ambiguous if there is a bag *on* a green table, and there is a bag next to a table that isn’t green. Such candidate objects are ranked according to how well they match the specifications in an utterance. Our system handles the following feature types: lexical item, colour, size, ownership and location. Our evaluation shows that our mechanism exhibits high resolution accuracy for different types of referring expressions.

This paper is organized as follows. Section 2 outlines the interpretation process and the estimation of the probability of an interpretation. Section 3 describes the probabilistic feature comparison. The results of our evaluation appear in Section 4. Related research and concluding remarks are given in Sections 5 and 6 respectively.

2 Interpretation Process

Scusi? processes spoken input in three stages: speech recognition, parsing and semantic interpretation (Figure 1). First, it runs Automatic Speech Recognition (ASR) software

**Fig. 1.** Stages of the interpretation process**Fig. 2.** UCG and ICG for a sample utterance

(Microsoft Speech SDK 5.1) to generate candidate texts from a speech signal. Each text is assigned a score that reflects the probability of the words given the speech wave. Next, *Scusi?* applies Charniak's probabilistic parser (<ftp://ftp.cs.brown.edu/pub/nlp/parser/>) to generate parse trees from the texts. The parser produces up to N ($= 50$) parse trees for each text, associating each parse tree with a probability.

During semantic interpretation, parse trees are successively mapped into two representations based on Conceptual Graphs [1]: first *Uninstantiated Concept Graphs* (UCGs), and then *Instantiated Concept Graphs* (ICGs) (Figure 2). UCGs are obtained from parse trees deterministically — one parse tree generates one UCG. A UCG represents syntactic information, where the concepts correspond to the words in the parent parse tree, and the relations are derived from syntactic information in the parse tree and prepositions. Each UCG can generate many ICGs. This is done by nominating different instantiated concepts and relations from the system's knowledge base as potential realizations for each concept and relation in a UCG.

Our interpretation process applies a selection-expansion cycle to build a search graph, where each level of the graph corresponds to one of the stages of the interpretation process (Figure 1). In each selection-expansion cycle, our algorithm selects an option for consideration (speech wave, textual ASR output, parse tree or UCG). At any point after an expansion, *Scusi?* can return a list of ranked interpretations (ICGs) with their parent sub-interpretations (text, parse tree(s) and UCG(s)).

Figure 2 illustrates a UCG and an ICG for an utterance containing the composite referring expression ("the long yellow tray on the table near the lamp"). The *intrinsic* features of an object (e.g., colour and size of the tray) are stored in the UCG node for this object. In contrast, *structural* features, which involve at least two objects (e.g., "the *table* near the *lamp*"), are represented as sub-graphs of the UCG (and then the ICG). This distinction is made because intrinsic features can be compared directly to features of objects in the knowledge base, while features that depend on the relationship between several objects require the identification of these objects and the verification of this relationship. In our example, all the tables and all the lamps in the room need to be considered, and the table/lamp combination that best matches the given specification is eventually selected. The procedures for selecting objects that match intrinsic and structural features are described in Section 3.

2.1 Estimating the Probability of an ICG

Scusi? ranks candidate ICGs according to their probability of being the intended meaning of a spoken utterance. Given a speech signal W and a context \mathcal{C} , the probability of an ICG I is represented as follows.

$$\Pr(I|W, \mathcal{C}) \propto \sum_{\Lambda} \Pr(I|U, \mathcal{C}) \cdot \Pr(U|P) \cdot \Pr(P|T) \cdot \Pr(T|W) \quad (1)$$

where U , P and T denote a UCG, parse tree and text respectively.

The summation is taken over all possible paths $\Lambda = \{P, U\}$ from a parse tree to the ICG, because a UCG and an ICG can have more than one parent. As mentioned above, the ASR and the parser return an estimate of $\Pr(T|W)$ and $\Pr(P|T)$ respectively; and $\Pr(U|P) = 1$, since the process of generating a UCG from a parse tree is deterministic. The estimation of $\Pr(I|U, \mathcal{C})$ is described in detail in [2]. Here we present the final equation obtained for $\Pr(I|U, \mathcal{C})$, and outline the ideas involved in its calculation.

$$\Pr(I|U, \mathcal{C}) \approx \prod_{k \in I} \Pr(u|k) \Pr(k|k_p, k_{gp}) \Pr(k|\mathcal{C}) \quad (2)$$

where k is an instantiated node in ICG I , u is the corresponding node in UCG U , k_p is the parent node of k in ICG I , and k_{gp} the grandparent node. For example, *near* is the parent of `lamp03`, and `table01` the grandparent of `lamp03` in the ICG in Figure 2.

- $\Pr(u|k)$ is the “match probability” between the specifications for node u in UCG U and the features of the corresponding node k in ICG I , e.g., how similar an object in the room is to the “long yellow tray” (Section 3.1).
- $\Pr(k|k_p, k_{gp})$ represents the structural probability of ICG I , simplified to node trigrams, e.g., whether `table01` is *near* `lamp03` (Section 3.2).
- $\Pr(k|\mathcal{C})$ is the probability of a concept in light of the context, which at present includes only domain knowledge.

3 Probabilistic Feature Comparison

Scusi? handles three intrinsic features, viz lexical item, colour and size; and two structural features, viz ownership and several types of locative references. The procedure for generating ICGs for a referring expression and calculating their probability is described in Algorithm 1. First, the intrinsic features of the objects in our world are used to calculate the probability of a match with each UCG concept (first factor in Equation 2, Step 2 in Algorithm 1). These probabilities are used to build a list of candidate objects that are a reasonable match for each UCG concept (Step 3). The objects in each list are iteratively combined into candidate ICGs, where each candidate represents an interpretation of the referring expression (Step 5). *Scusi?* then considers the structural features of each ICG to calculate its structural probability (Step 7, second factor in Equation 2), and combines the intrinsic and structural probabilities to calculate the probability of the ICG (Step 8). Finally, the ICGs are ranked according to their probability (Step 10).

For example, consider a request for “the blue mug on the table”, assuming that the knowledge base contains several mugs, some of which are blue. First, for all the objects

Algorithm 1. Generate candidate ICGs for a referring expression

Require: UCG U comprising concepts and relations u , knowledge base \mathcal{K} of objects

- 1: **for all** objects $u \in \text{UCG } U$ **do**
- 2: Estimate $\Pr(u|k)$, the probability of the match between the features of u and those of each object $k \in \mathcal{K}$.
- 3: Rank the candidate objects $k \in \mathcal{K}$ in descending order of probability.
- 4: **end for**
- 5: Construct candidate ICGs by iteratively going down the list of objects generated for each concept u in UCG U — each candidate ICG contains one object from each list.
- 6: **for all** ICGs I **do**
- 7: Estimate the probabilities $\Pr(k|k_p, k_{gp})$ for each *object-relation-object* trigram in I .
- 8: Combine these estimates with the probabilities from Step 2 to obtain the probability of I .
- 9: **end for**
- 10: Rank the candidate ICGs in descending order of probability.

in the knowledge base, we estimate the probability that they could be called ‘mug’ (e.g., mugs, cups), and the probability that their colour could be considered ‘blue’; similarly, we calculate the probability that an object could be called ‘table’ (Section 3.1). The candidates for ‘blue mug’ and ‘table’ are then ranked in descending order of probability. Candidate ICGs are built by iteratively combining each candidate blue mug with each candidate table. The structural probability of each ICG is then calculated on the basis of the location coordinates of the mug and table instances in the ICG (Section 3.2).

At present, we make the following simplifying assumptions: (1) the robot is co-present with the user and the possible referents of an utterance; and (2) the robot has an unobstructed view of the objects in the room and up-to-date information about these objects. This information could be obtained through a scene analysis system [3] activated upon entering a room. These assumptions obviate the need for planning physical actions, such as moving to get a better view of certain objects, or leaving the room to seek objects that better match the given specifications.

3.1 Estimating the Probabilities of Intrinsic Features

The probability of the match between a node u specified in UCG U and a candidate instantiated concept $k \in \mathcal{K}$ (Step 2 of Algorithm 1) is estimated as follows.

$$\Pr(u|k) = \Pr(\mathbf{u}_{f_1}, \dots, \mathbf{u}_{f_p} | \mathbf{k}_{f_1}, \dots, \mathbf{k}_{f_p}) \quad (3)$$

where $(f_1, \dots, f_p) \in \mathcal{F}$ are the features specified with respect to node u , \mathcal{F} is the set of features allowed in the system, \mathbf{u}_{f_i} is the value of the i -th feature of UCG node u , and \mathbf{k}_{f_i} is the value of this feature for the instantiated concept k .

Assuming that the features of a node are independent, the probability that an instantiated concept k matches the specifications in a UCG node u can be rewritten as

$$\Pr(u|k) = \prod_{i=1}^p \Pr(\mathbf{u}_{f_i} | \mathbf{k}_{f_i}) \quad (4)$$

In the absence of other information, it is reasonable to use a linear distance function $h: \mathbb{R}^+ \rightarrow [0, 1]$ to map the outcome of a feature match to the probability space. That is,

the higher the similarity between requested and instantiated feature values (the shorter the distance between them), the higher the probability of a feature match. Specifically,

$$\Pr(\mathbf{u}_f | \mathbf{k}_f) = h_f(\mathbf{u}_f, \mathbf{k}_f) \quad (5)$$

Below we present the calculation of Equation 5 for the intrinsic features supported by our system (lexical item, colour and size). In agreement with [4,5], lexical item and colour are considered *absolute* features, and size a *relative* feature (its value depends on the size of other candidates).

Lexical Item. We employ the Leacock and Chodorow [6] similarity measure, denoted LC , to compute the similarity between the lexical feature of u and k . This measure is applied to the words in a database constructed with the aid of WordNet (the LC measure yielded the best results among those in [7]). The LC similarity score, denoted s_{LC} , is converted to a probability by applying the following h_{lex} function.

$$\Pr(\mathbf{u}_{\text{lex}} | \mathbf{k}_{\text{lex}}) = h_{\text{lex}}(s_{LC}(\mathbf{u}_{\text{lex}}, \mathbf{k}_{\text{lex}})) = \frac{s_{LC}(\mathbf{u}_{\text{lex}}, \mathbf{k}_{\text{lex}})}{s_{\max}}$$

where s_{\max} is the highest possible LC score.

Colour. The colour model chosen for *Scusi?* is the CIE 1976 (L, a, b) colour space, which has been experimentally shown to be approximately perceptually uniform [8]. The L coordinate represents brightness ($L = 0$ denotes black, and $L = 100$ white), a represents position between green ($a < 0$) and red ($a > 0$), and b position between blue ($b < 0$) and yellow ($b > 0$). The range of L is $[0, 100]$, while for practical purposes, the range of a and b is $[-200, 200]$. Thus, the probability of a colour match between a UCG concept u and an instantiated concept k is

$$\Pr(\mathbf{u}_{\text{colr}} | \mathbf{k}_{\text{colr}}) = h_{\text{colr}}(\mathbf{u}_{\text{colr}}, \mathbf{k}_{\text{colr}}) = 1 - \frac{ED(\mathbf{u}_{\text{colr}}, \mathbf{k}_{\text{colr}})}{d_{\max}}$$

where ED is the Euclidean distance between the (L, a, b) coordinates of the colour specified for u and the (L, a, b) coordinates of the colour of k , and d_{\max} is the maximum Euclidean distance between two colours (=574.5).

Size. Unlike lexical item and colour, size is considered a relative feature, i.e., the probability of a size match between an object $k \in \mathcal{K}$ and a UCG concept u depends on the sizes of all suitable candidate objects in \mathcal{K} (those that have a reasonable match for lexical and colour comparisons). The highest probability for a size match is then assigned to the object that best matches the required size, while the lowest probability is assigned to the object which has the worst match with this size.

This requirement is achieved by the following h_{size} function, which like Kelleher *et al.*'s pixel-based mapping [9], performs a linear mapping between \mathbf{u}_{size} and \mathbf{k}_{size} .

$$\Pr(\mathbf{u}_{\text{size}} | \mathbf{k}_{\text{size}}) = h_{\text{size}}(\mathbf{u}_{\text{size}}, \mathbf{k}_{\text{size}}) = \begin{cases} \frac{\alpha \mathbf{k}_{\text{size}}}{\max_i \{\mathbf{k}_{\text{size}}^i\}} & \text{if } \mathbf{u}_{\text{size}} \in \{ \text{'large'}/\text{'big'}/\dots \} \\ \frac{\alpha \min_i \{\mathbf{k}_{\text{size}}^i\}}{\mathbf{k}_{\text{size}}} & \text{if } \mathbf{u}_{\text{size}} \in \{ \text{'small'}/\text{'little'}/\dots \} \end{cases}$$

where α is a normalizing constant, and $\mathbf{k}_{\text{size}}^i$ is the size of candidate object k^i (this formula is adapted for individual dimensions, e.g., length).

Combining Feature Scores. To determine how intrinsic features are used in our domain, we conducted a survey where people were asked to refer to household objects laid out in a space [10]. The results of our survey agree with Dale and Reiter’s findings [4], whereby people often present features that are not strictly necessary to identify an item, and use features in the following order of frequency: *type* \succ *absolute adjectives* \succ *relative adjectives*, where colour is an absolute feature and size is a relative feature.

These findings prompted us to incorporate a weighting scheme into Equation 4, whereby features are weighted according to their usage in referring expressions. That is, higher ranking or more frequently used features are assigned a higher weight than lower ranking or less frequently used features. Specifically, given a match probability $\Pr(\mathbf{u}_{f_i} | \mathbf{k}_{f_i})$ and a weight w_{f_i} for feature f_i ($0 < w_{f_i} \leq 1$), the adjusted match probability for this feature is

$$\Pr'(\mathbf{u}_{f_i} | \mathbf{k}_{f_i}) = \Pr(\mathbf{u}_{f_i} | \mathbf{k}_{f_i}) \times w_{f_i} + \frac{1}{2}(1 - w_{f_i})$$

The effect of this mapping is that features with high weights have a wide range of probabilities (and hence a substantial influence on the match probability of an object), while features with low weights have a narrow range (and a reduced influence on match probability).

3.2 Estimating the Probabilities of Structural Features

As shown in Equation 2, the overall probability of an ICG structure can be decomposed into a product of the probabilities of the trigrams that make up the ICG. A trigram consists of a relationship k_p (e.g., ownership or location) and two instantiated concepts k and k_{gp} , e.g., `table01`–`near`–`lamp03` in Figure 2. A probability is assigned to this trigram based on the physical coordinates of `table01` and `lamp03`.

Below we present the probability calculation for the two structural features supported by our system (ownership and location). This calculation involves validating the structural feature against the information in our world, and, as for intrinsic features, performing a linear mapping from the result of this validation to a probability.

Ownership. In our world, an object is either owned by one or more people or no owner has been recorded for this object. This leads to a simple probabilistic mapping.

$$\Pr(\mathbf{k} | \text{own}, \mathbf{k}_{gp}) = \begin{cases} 0 & \text{if } \mathbf{k} \notin \text{owner-of}(\mathbf{k}_{gp}) \\ \beta & \text{if } \text{owner-of}(\mathbf{k}) \text{ is unknown} \\ 1 & \text{if } \mathbf{k} \in \text{owner-of}(\mathbf{k}_{gp}) \end{cases}$$

where β is currently set to 0.5.

Location. At present, we assume that all the objects in our world are rigid, and hence can be represented by a circumscribing box, e.g., a lamp is represented by the smallest box that contains the lamp. As a result, each object \mathbf{k} has three dimensions and one position coordinate. The dimensions are $(\mathbf{k}_l, \mathbf{k}_w, \mathbf{k}_h)$, corresponding to the object’s length, width and height respectively. The position coordinate is $(\mathbf{k}_x, \mathbf{k}_y, \mathbf{k}_z)$, measured between a starting coordinate $(0, 0, 0)$ and the closest corner of the box. The system handles the following locative prepositions: *on*, *under*, *above*, *in (inside)* and *near (by)*.

- ***on, under, above*** – these prepositions have the following directional semantics.

- ***on*** means that $\mathbf{k}_{gp_z} = \mathbf{k}_z + \mathbf{k}_h$, where $\mathbf{k}_z + \mathbf{k}_h$ represents the height of the top surface of the bounding box for object \mathbf{k} ;
- ***under*** means that $\mathbf{k}_{gp_z} + \mathbf{k}_{gp_h} \leq \mathbf{k}_z$; and
- ***above*** means that $\mathbf{k}_{gp_z} > \mathbf{k}_z + \mathbf{k}_h$.

If the objects \mathbf{k} and \mathbf{k}_{gp} in an ICG satisfy the directional requirement of their location preposition ($loc \in \{on, under, above\}$), we say that $Pr(\mathbf{k}|loc, \mathbf{k}_{gp})$ is proportional to the area shared by the horizontal surfaces of (the bounding boxes of) the two objects. Otherwise, $Pr(\mathbf{k}|loc, \mathbf{k}_{gp})$ is set to a low probability (ϵ). Specifically, let $A(\mathbf{k})$ denote the area of the top face of object \mathbf{k} , and let $A(\mathbf{k}, \mathbf{k}_{gp})$ denote the overlapping area between the top faces of objects \mathbf{k} and \mathbf{k}_{gp} in the xy plane. The probability of a trigram involving location relations *on*, *under* or *above* is

$$Pr(\mathbf{k}|loc, \mathbf{k}_{gp}) = \begin{cases} \frac{A(\mathbf{k}, \mathbf{k}_{gp})}{\min\{A(\mathbf{k}), A(\mathbf{k}_{gp})\}} & \text{if directional requirement is satisfied} \\ \epsilon \ll 0.1 & \text{otherwise} \end{cases}$$

For example, consider the utterance “the book on the table”, for which one of the candidate ICGs is `book01→on→table02`. The directional semantics for *on* stipulate that the z coordinate of (the bottom of) the book (\mathbf{k}_{gp}) must be equal to the z coordinate of the table (\mathbf{k}) plus the height of the table. If this condition is satisfied, then the degree of overlap between the surface of the book and that of the table is calculated. That is, a book that is entirely on a table top satisfies the *on* relationship with a higher probability than a book overhanging the table.

- ***in (inside)*** – the probability of an object being inside another is proportional to the volume shared by their bounding boxes (one object could be partially inside another). Formally, let $V(\mathbf{k})$ denote the volume of (the bounding box of) object \mathbf{k} , and let $V(\mathbf{k}, \mathbf{k}_{gp})$ denote the shared volume between (the bounding boxes of) objects \mathbf{k} and \mathbf{k}_{gp} . The probability of an *in*-trigram is

$$Pr(\mathbf{k}|in, \mathbf{k}_{gp}) = \frac{V(\mathbf{k}, \mathbf{k}_{gp})}{\min\{V(\mathbf{k}), V(\mathbf{k}_{gp})\}}$$

For example, if we are asked for “the mug inside the box”, a mug that is wholly contained within a box would yield a higher probability than a mug whose top exceeds the top of a box.

- ***near*** – following [9], we employ a formulation inspired by the gravitational model to calculate the probability of two objects being near each other. However, since the density of objects is not specified in our world, we approximate the mass of an object by its volume. Formally, let $d(\mathbf{k}, \mathbf{k}_{gp})$ represent the shortest distance between the bounding boxes of \mathbf{k} and \mathbf{k}_{gp} . The probability of a *near*-trigram is

$$Pr(\mathbf{k}|near, \mathbf{k}_{gp}) = \frac{V(\mathbf{k})V(\mathbf{k}_{gp})}{d^2(\mathbf{k}, \mathbf{k}_{gp}) G_{max}}$$

where G_{max} , the maximum gravitational pull in our world, is obtained when the two biggest objects in our world abut (i.e., d is arbitrarily small).

This model enables the size of the objects to influence the nearness probability. For example, if one asks for “the ball next to the table”, and there is a tennis ball a few centimeters from the table, and a beach ball farther from the table, this model will identify the ambiguity, and support the generation of a clarification question.

4 Evaluation

To evaluate our system, we constructed a simulated world that represents an open-plan house (in keeping with our co-presence assumption, Section 3). The world contains 54 objects distributed among four areas in the house, and five people (Figure 3 illustrates one of the areas of the house, with various objects labeled). The objects were chosen so that they had similar features, i.e., several objects could be referred to by the same lexical item (there were 2-4 instances of each type of object), had similar colours and sizes, and were placed in adjacent locations. The ownership of most objects was distributed among the five people in our world, and some objects had no known owner.

In total, 90 referring expressions of varying complexity were used for the system evaluation. Each referring expression consisted of a noun phrase comprising between one and three concepts (sample expressions are shown in Figure 4). Mean utterance length was 4.27 words, with a maximum length of 8 words. The expressions were constructed to test *Scusi?*'s ability to identify target objects (the intended book, mug, table, etc) in different situations. Specifically, objects were referred to by near synonyms (e.g., "mug" and "cup"), by colours and sizes that were shared by several objects, and by their proximity to a reference object that was adjacent to several objects. For example, the utterance "the book near Paul's mug" tests the system's ability to identify an object by its ownership and location in a world that contains several books and mugs.

Scusi? was set to generate at most 300 sub-interpretations in total (including texts, parse trees, UCGs and ICGs) for each referring expression. An ICG proposed by *Scusi?* was deemed correct if it matched the speaker's intention, which was represented by one or more Gold ICGs. These ICGs were manually constructed by one of the authors for each referring expression on the basis of the information in *Scusi?*'s knowledge base. Multiple Gold ICGs were allowed if several objects in the domain matched a specified object, e.g., "a bowl". A baseline measure of performance was obtained by executing a beam search. That is, only the top-ranked ASR result was parsed, and only the top-ranked parse tree yielded a UCG, which in turn produced only one top-ranked ICG.

Table 1 summarizes our results. Column 1 shows the procedure (*Scusi?*'s or baseline). Columns 2 and 3 show how many of the descriptions had Gold ICG referents whose probability was the highest (top 1) or among the three highest (top 3), e.g., *Scusi?* yielded 82 Gold ICGs with the top probability, and all the 90 referring expressions had

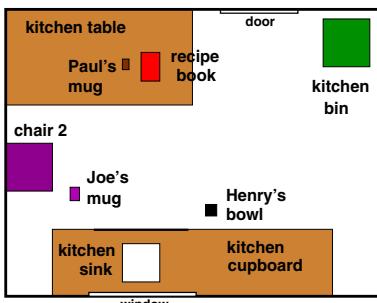


Fig. 3. Sample area from our world

- 1 A desk
- 2 The purple bowl
- 3 Paul's book
- 4 The green mug in the lounge
- 5 Sarah's bowl in the lounge
- 6 The long pants in the bathroom
- 7 The wardrobe under the fan
- 8 A bin near the small plant
- 9 The mug near the book on the table
- 10 The shirt in the bag near the plant

Fig. 4. Sample referring expressions

Table 1. *Scusi?*’s interpretation performance

	# Gold ICGs with prob in top 1	# Gold ICGs with prob in top 3	Average adj rank(rank)	Not found	Avg # to Gold ICGs (iters)
BASELINE	44	44	0 (0)	46	0 (4)
<i>Scusi?</i>	82	90	0.96 (0.11)	0	2.45 (25)

Gold referents within the top 3 probabilities. The average *adjusted rank* and *rank* of the Gold referent appear in Column 4. The rank of a referent r is its position in a list sorted in descending order of probability (starting from position 0), such that all equiprobable referents are deemed to have the same position. The adjusted rank of a referent r is the mean of the positions of all referents that have the same probability as r . For example, if we have 3 top-ranked equiprobable referents, each has a rank of 0, but an adjusted rank of $\frac{0+2}{2}$. Column 5 indicates the number of referring expressions for which a Gold ICG was not found, and Column 6 shows the average number of referents created and iterations performed until the Gold referent was found (from a total of 300 iterations).

Our results show that maintaining multiple hypotheses at each stage of the interpretation process yields a substantial improvement in interpretation accuracy in comparison to the baseline approach. *Scusi?* found the Gold interpretation for all 90 utterances tested, in contrast to the baseline approach, which found only 44 Gold ICGs. The average rank of the correct text in the output returned by the ASR was 1.5 (where the top rank is 0), and the correct text was top ranked by the ASR in 70% of the cases. This level of accuracy is higher than the accuracy of the baseline approach with respect to Gold ICGs ($44/90 = 49\%$), which indicates that even when presented with the correct text, the baseline approach may not find the intended interpretation. Furthermore, ASR accuracy is lower than *Scusi?*’s accuracy for top-1 Gold ICGs ($82/90 = 91\%$), demonstrating the robustness of the probabilistic multi-stage interpretation process in the face of ASR inaccuracy.

5 Related Research

Reference disambiguation is an essential aspect of discourse understanding to which a large research effort has been devoted. Much of the research on reference resolution has focused on the generation of referring expressions, which involves constructing expressions that single out a target object from a set of distractors, e.g., [4,5]. Methods for understanding referring expressions in dialogue systems are examined in [9,11] among others. Kelleher *et al.* [9] propose a reference resolution algorithm that accounts for four attributes: lexical type, colour, size and location, where the score of an object is estimated by a weighted combination of the visual and linguistic salience scores of each attribute. Like in *Scusi?*, the values of the weights are pre-defined and based on empirical observations. However, Kelleher *et al.* limit the probabilistic comparison of features to size and location, and use binary comparisons for lexical item and colour. Pfleger *et al.* [11] use modality fusion to combine hypotheses from different analyzers (linguistic, visual and gesture), choosing as the referent the first object satisfying a ‘differentiation criterion’. As a result, their system does not handle situations where more than one object satisfies this criterion.

6 Conclusion

We have offered a probabilistic reference disambiguation mechanism which considers intrinsic and structural features. Our mechanism performs probabilistic comparisons between features specified in referring expressions (specifically lexical item, colour, size, ownership and location) and features of objects in the domain. Our mechanism was empirically evaluated for these features, exhibiting very good interpretation performance for a range of referring expressions.

In the future, we propose to extend the weighting mechanism devised for intrinsic features to cater for structural features and their combination with intrinsic features. We also propose to integrate our mechanism with a vision system, which will affect the type of information we can obtain from our knowledge base. Finally, we intend to remove the co-presence and unobstructed-view assumptions (Section 3), which will demand the integration of our feature comparison mechanism with planning procedures.

References

1. Sowa, J.: *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading (1984)
2. Zukerman, I., Makalic, E., Niemann, M., George, S.: A probabilistic approach to the interpretation of spoken utterances. In: Ho, T.-B., Zhou, Z.-H. (eds.) PRICAI 2008. LNCS (LNAI), vol. 5351, pp. 581–592. Springer, Heidelberg (2008)
3. Makihara, Y., Takizawa, M., Shirai, I., Miura, J., Shimada, N.: Object recognition supported by user interaction for service robots. In: Proceedings of the 16th International Conference on Pattern Recognition, Quebec, Canada, vol. 3, pp. 561–564 (2002)
4. Dale, R., Reiter, E.: Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science* 18(2), 233–263 (1995)
5. Wyatt, J.: Planning clarification questions to resolve ambiguous references to objects. In: Proceedings of the 4th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Edinburgh, Scotland, pp. 16–23 (2005)
6. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. In: Fellbaum, C. (ed.) *WordNet: An Electronic Lexical Database*, pp. 265–285. MIT Press, Cambridge (1998)
7. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet:Similarity – measuring the relatedness of concepts. In: AAAI 2004 – Proceedings of the 19th National Conference on Artificial Intelligence, San Jose, California, pp. 25–29 (2004)
8. Puzicha, J., Buhmann, J., Rubner, Y., Tomasi, C.: Empirical evaluation of dissimilarity measures for color and texture. In: Proceedings of the 7th IEEE International Conference on Computer Vision, Kerkyra, Greece, vol. 2, pp. 1165–1172 (1999)
9. Kelleher, J., Kruijff, G., Costello, F.: Proximity in context: an empirically grounded computational model of proximity for processing topological spatial expressions. In: COLING-ACL 2006 Proceedings, Sydney, Australia, pp. 745–752 (2006)
10. Zukerman, I., Makalic, E., Niemann, M.: Using probabilistic feature matching to understand spoken descriptions. In: AI 2008 Proceedings – the 21st Australasian Joint Conference on Artificial Intelligence, Auckland, New Zealand (2008)
11. Pfleger, N., Alexandersson, J., Becker, T.: A robust and generic discourse model for multimodal dialogue. In: Proceedings of the 3rd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Acapulco, Mexico (2003)

Fuzzy Communication Reaching Consensus under Acyclic Condition

Takashi Matsuhisa

Department of Natural Sciences, Ibaraki National College of Technology
Nakane 866, Hitachinaka-shi, Ibaraki 312-8508, Japan
mathisa@ge.ibaraki-ct.ac.jp

Dedicated to Professor Tatsuo Kimura in the occasion of his 60th birthday

Abstract. We present a fuzzy communication model in multi-agent system. In the model agents have the fuzzy structure associated with a partition, and each agent obtains the membership value of an event under his/her private information. Each agent can consider the event as a fuzzy set, and he/she sends not exact information on the membership value but fuzzy information on it to the another according to a communication graph. We show that consensus on the fuzzy event can still be guaranteed among all agents in the communication according to acyclic communication graph; i.e., all the membership values are equal after long running communication.

Keywords: Fuzzy communication, consensus, messages protocol, agreeing to disagree.

AMS 2000 Mathematics Subject Classification: Primary 91B50, 91B60; Secondary 03B45.

1 Introduction

This article considers the relationship between communication and agreement in multi-agent system. How we capture the fact that the agents agree on a statement or they get consensus on it? We treat the problem from Fuzzy set theoretical flavour. The purpose is to introduce a knowledge revision system through communication on multi-agent system, and by which we show that all agents can agree on a truth degree of a sentence.

Let us consider there is a set N of n agents more than two and the agents have the *fuzzy structure* given by the dual structure of the Kripke semantics for the multi-modal logic $\mathbf{S5}_n$. Assume that all agents have a common probability measure μ on Ω . Let X be an *event* of Ω . By the *membership value* of X under agent i 's private information $\Pi_i(\omega)$ at ω we mean the conditional probability value $d_i(X; \omega) = \mu(X|\Pi_i(\omega))$. We say that *consensus* on the fuzzy set X can be guaranteed among all agents (or they *agree on* it) if the fuzzy sets (X, d_i) are equal for all $i \in N$; i.e.; $d_i(X; \omega) = d_j(X; \omega)$ for any agent $i, j \in N$ and in all $\omega \in \Omega$.

R.J. Aumann [1] considered the situation that the agents has common-knowledge of an event; that is, simultaneously everyone knows the event, everyone knows that everyone knows the event, and so on. He showed the famous agreement theorem:

Theorem 1 (R. J. Aumann [1], R. Fagin et al [4]). *The agents can agree on an event if all membership values of the event under private information are common-knowledge.*

That is, if they has common-knowledge of the event $\{\xi \in \Omega \mid d_i(X; \xi) = d_i(X; \omega)\}$ at ω , then $d_i(X; \omega) = d_j(X; \omega)$ for any $i, j \in N$.

This article considers the situation that the agents communicate each other under fuzzy information on the membership values. Let us start the agents interact in pairs with private announcement: Each agent has the membership value of an event under his/her private information, and he/she privately announces it to the another agent through fuzzy messages; i.e., by the fuzzy message we mean the possibility set of the membership values of the event. Hence there is a possibility losing a bit information on when the agent receives the message from the other agent. The recipient revises his/her information structure and recalculates the membership values under the approximate information on the membership value of the event. The agent sends the revised membership value of the event to another agent according to a communication graph. The recipient revises his/her membership value of X and send it to another, and so on. We shall show that.

Theorem 2. *Suppose that all agents have a common prior distribution. Consensus on the limiting membership values of an event under his/her private information among all agents can still be guaranteed in the communication even when each agent sends not exact information but fuzzy information on it through messages.*

This paper organises as follows. In Section 2 we present the fuzzy structure associated with a partition information structure, and introduces the fuzzy communication system. Section 3 gives the formal statement of Theorem 2 with a sketch of the proof. In Section 4 we conclude with remarks. The illustrated example will be shown in the conference talk.

2 The Model

Let N be a set of finitely many *agents* and i denote an agent. A *state-space* is a non-empty set, whose members are called *states*. An *event* is a subset of the state-space. If Ω is a state-space, we denote by 2^Ω the field of all subsets of it. An event X is said to *occur* in a state ω if $\omega \in X$.

2.1 Information and Fuzzy Operator

A *partition information structure* $\langle \Omega, (\Pi_i)_{i \in N} \rangle$ consists of a state space Ω and a class of *information functions* Π_i of Ω into 2^Ω satisfying the postulates

- (i) $\{\Pi_i(\omega) \mid \omega \in \Omega\}$ is a partition of Ω ; i.e., Ω is decomposed into the disjoint union of components $\Pi_i(\omega)$ for $\omega \in \Omega$; and
- (ii) $\omega \in \Pi_i(\omega)$ for every $\omega \in \Omega$.

The set $\Pi_i(\omega)$ will be interpreted as the set of all the states of nature that i knows to be possible at ω , or as the set of the states that i cannot distinguish from ω . We will therefore call $\Pi_i(\omega)$ i 's *information set* at ω , and we will give the interpretation that i can recognise an event X as the fuzzy set of all the states ω with $X \cap \Pi_i(\omega) \neq \emptyset$. Formally we introduce the fuzzy structure:

Definition 1. The *fuzzy structure* $\langle \Omega, (\Pi_i)_{i \in N}, (F_i)_{i \in N} \rangle$ consists of a partition information structure $\langle \Omega, (\Pi_i)_{i \in N} \rangle$ and a class of i 's *fuzzy operator* F_i on 2^Ω such that

$$F_i X = \{\omega \in \Omega \mid X \cap \Pi_i(\omega) \neq \emptyset\}.$$

The event $F_i X$ will be interpreted as the set of states of nature for which X to be possible, and so we shall call $F_i(X)$ i 's *possibility set* of X .

We record the properties of i 's fuzzy operator: For every X, Y of 2^Ω ,

- | | | | |
|-----------|--|-----------|--------------------------------------|
| FN | $F_i \Omega = \Omega$ and $F_i \emptyset = \emptyset$; | FK | $F_i(X \cup Y) = F_i X \cup F_i Y$; |
| FT | $X \subseteq F_i X$; | F4 | $F_i F_i X \subseteq F_i X$; |
| F5 | $F_i(\Omega \setminus F_i(X)) \subseteq \Omega \setminus F_i(X)$. | | |

Given another interpretation, an agent i for whom $\Pi_i(\omega) \subseteq X$ knows, in the state ω , that some state in the event X has occurred. In this case we say that in the state ω the agent i knows X .

Definition 2. The *knowledge structure* $\langle \Omega, (\Pi_i)_{i \in N}, (K_i)_{i \in N} \rangle$ consists of a partition information structure $\langle \Omega, (\Pi_i)_{i \in N} \rangle$ and a class of i 's *knowledge operator* K_i on 2^Ω such that $K_i E$ is the set of states of Ω in which i knows that E has occurred; that is,

$$K_i X = \{\omega \in \Omega \mid \Pi_i(\omega) \subseteq X\}.$$

The event $K_i X$ will be interpreted as the set of states of nature for which i knows X to be possible.

We record the properties of i 's knowledge operator¹: For every X, Y of 2^Ω ,

- | | | | |
|----------|--|----------|--------------------------------------|
| N | $K_i \Omega = \Omega$ and $K_i \emptyset = \emptyset$; | K | $K_i(X \cap Y) = K_i X \cap K_i Y$; |
| T | $K_i X \subseteq X$; | 4 | $K_i X \subseteq K_i K_i X$; |
| 5 | $\Omega \setminus K_i(X) \subseteq K_i(\Omega \setminus K_i(X))$. | | |

Remark 1. i 's knowledge operator K_i is the dual of the fuzzy operator F_i , and so each operator is uniquely determined by i 's information function Π_i .

¹ According to these properties we can say the structure $\langle \Omega, (K_i)_{i \in N} \rangle$ is a model for the multi-modal logic $S5_n$.

2.2 Decision Function and Membership Value

Let X be an event of Ω . By a *decision function* of X we mean a mapping $f(X|\cdot)$ of 2^Ω into the unit interval $[0, 1]$. Assume here that the function f satisfies the following properties:

Union consistency: For every pair of disjoint events S and T , if $f(S) = f(T) = d$ then $f(S \cup T) = d$;

Preserving under difference: For all events S and T such that $S \subseteq T$, if $f(S) = f(T) = d$ then we have $f(T \setminus S) = d$.

By the *membership function* of X under agent i 's private information at ω we mean the function $d_i(X; \cdot)$ from Ω into $[0, 1]$ defined by $d_i(X; \omega) = f(X|\Pi_i(\omega))$, and we call $d_i(X; \omega)$ the *membership value* of X under agent i 's private information at ω . The pair (X, d_i) can be considered as a fuzzy set X associated with agent i 's membership function d_i . We say that *consensus* on X can be guaranteed among all agents (or they *agree on* it) if $d_i(X; \omega) = d_j(X; \omega)$ for any agent $i, j \in N$ and in all $\omega \in \Omega$. This is interpreted as the fuzzy sets (X, d_i) and (X, d_j) are equal for any i, j .

If f is intended to be a posterior probability, we assume given a probability measure μ which is common for all agents and some event X . Then the *membership value* of X is the conditional probability value $d_i(X; \omega) = \mu(X|\Pi_i(\omega))$.

2.3 Protocol²

We assume that the agents communicate by sending *messages*. Let T be the time horizontal line $\{0, 1, 2, \dots, t, \dots\}$. A *protocol* is a mapping $\Pr : T \rightarrow N \times N, t \mapsto (s(t), r(t))$ such that $s(t) \neq r(t)$. Here t stands for *time* and $s(t)$ and $r(t)$ are, respectively, the *sender* and the *recipient* of the communication which takes place at time t . We consider the protocol as the directed graph whose vertices are the set of all agents N and such that there is an edge (or an arc) from i to j if and only if there are infinitely many t such that $s(t) = i$ and $r(t) = j$.

A protocol is said to be *fair* if the graph is strongly-connected; in words, every agent in this protocol communicates directly or indirectly with every other agent infinitely often. It is said to contain a *cycle* if there are at least k agents i_1, i_2, \dots, i_k with $k \geq 3$ such that for all $m < k$, i_m communicates directly with i_{m+1} , and such that i_k communicates directly with i_1 . The communications is assumed to proceed in *rounds*; i.e., There exists a time m such that for all t , $\Pr(t) = \Pr(t+m)$. The *period* of the protocol is the minimal number of all m such that for every t , $\Pr(t+m) = \Pr(t)$.

2.4 Communication under Fuzzy Information

A *fuzzy communication system* π with revisions of agents' membership functions $(d_i^t)_{(i,t) \in N \times T}$ according to a protocol \Pr is a tuple

$$\pi = \langle \Omega, \Pr, (\Pi_i^t), f, (d_i^t) | (i, t) \in N \times T \rangle$$

² C.f.: Parikh and Krasucki [9].

with the following structures: the agents have a common prior μ on Ω , the protocol Pr among is fair and it satisfies the conditions that $r(t) = s(t+1)$ for every t and that the communications proceed in rounds. The revised information structure Π_i^t at time t is the mapping of Ω into 2^Ω for agent i . If $i = s(t)$ is a sender at t , the *message* sent by i to $j = r(t)$ is M_i^t . An n -tuple $(d_i^t)_{i \in N}$ is a revision system of individual conjectures. These structures are inductively defined as follows:

- Set $\Pi_i^0(\omega) = \Pi_i(\omega)$.
- Assume that Π_i^t is defined. It yields i 's decision $d_i^t(\omega) = d_i(\Pi_i^t(\omega))$ together with the revised fuzzy operator F_i^t on 2^Ω defined by $F_i^t(X) = \{\omega \in \Omega | \Pi_i^t(\omega) \cap X \neq \emptyset\}$. Whence the fuzzy message $M_i^t : \Omega \rightarrow 2^\Omega$ sent by the sender i at time t is defined by

$$M_i^t(\omega) = F_i^t([d_i^t(\omega)]),$$

where $[d_i^t(\omega)] = \{\xi \in \Omega | d_i^t(\xi) = d_i^t(\omega)\}$, interpreted as the event of $d_i^t(\omega)$.

Then:

- The revised partition Π_i^{t+1} at time $t+1$ is defined as follows:
 - $\Pi_i^{t+1}(\omega) = \Pi_i^t(\omega) \cap M_{s(t)}^t(\omega)$ if $i = r(t)$;
 - $\Pi_i^{t+1}(\omega) = \Pi_i^t(\omega)$ otherwise,

The specification is that a sender $s(t)$ at time t informs the recipient $r(t)$ his/her membership value of the event as a fuzzy information of X under $s(t)$'s private information. The recipient revises her/his information structure under the information. She/he revises her/his membership value according the possibility message, and she/he informs her/his the revised membership value to the other agent $r(t+1)$.

We denote by ∞ a sufficient large $\tau \in T$ such that for all $\omega \in \Omega$, $d_i^\tau(\cdot; \omega) = d_i^{\tau+1}(\cdot; \omega) = d_i^{\tau+2}(\cdot; \omega) = \dots$ Hence we can write d_i^τ by d_i^∞ .

The fuzzy communication system is said to be satisfied the *acyclic condition* if the communication protocol Pr contains no cycle.

2.5 Consensus

We note that the limit Π_i^∞ exists.³ We denote $d_i^\infty(\omega) = f(\Pi_i^\infty(\omega))$ called the *limiting membership value* of f at ω for i . We say that *consensus* on the limiting membership values can be guaranteed if $d_i^\infty(\omega) = d_j^\infty(\omega)$ for each agent i, j and in all the states ω .

3 Main Theorem

We can observe that Theorem 2 is a corollary of Theorem 3 as below.

³ In fact, Ω is a Hausdorff topological space equipped with the clopen base $\{\Pi_i(\omega) | \omega \in \Omega\}$. It can be easily seen that $F_i(X)$ is closed and $M_i(\omega)$ is also. Hence the chain $\{\Pi_i^t(\omega) | t = 0, 1, 2, \dots\}$ is the descending one of closed sets, and so it must be stationary, and so the chain has the unique limit $\cap_{t=0}^\infty \Pi_i^t(\omega)$.

Theorem 3. *If the common decision function in the above fuzzy communication system π is preserved under difference with union consistency, then consensus on the limiting values of the decision function can be guaranteed; i.e., $d_i^\infty(\omega) = d_j^\infty(\omega)$ for every ω and for all i, j .*

Remark 2. The fuzzy sets (X, d_i^∞) are equal for any agent $i \in N$.

Proof of Theorem 3 (Sketch). Let us consider the agents i, j such that $(i, j) = (s(t), r(t))$ and $(j, i) = (s(t+1), r(t+1))$. Let ω be a state, and denote $M_i = M_i^\infty(\omega)$. We can observe the two points: First that $M_i \cap M_j$ can be decomposed into the disjoint union of $\Pi_i^\infty(\xi)$ for $\xi \in M_i \cap M_j$ and it can be also done into the disjoint union of $\Pi_j^\infty(\zeta)$ for $\zeta \in M_i \cap M_j$. Therefore it follows from union consistency that $f(M_i \cap M_j) = f(\Pi_i^\infty(\omega)) = f(\Pi_j^\infty(\omega))$, and so $d_i^\infty(\omega) = d_j^\infty(\omega)$.

Next we shall proceed in the general case. On noting the protocol is fair, let us consider the protocol excluding the agent j . Because the protocol is also *acyclic*, we can choose the agents k, l that $(k, l) = (s(t'), r(t'))$ and $(l, k) = (s(t'+1), r(t'+1))$, and by the same argument as above, we obtain that $d_k^\infty(\omega) = d_l^\infty(\omega)$.

Hence, by the induction argument on the period of the protocol it can be shown that $d_i^\infty(\omega) = d_j^\infty(\omega)$ for every ω and for all i, j , as required. \square

4 Concluding Remarks

Recently researchers in such fields as economics, AI, and theoretical computer science have become interested in reasoning of belief and knowledge. There are pragmatic concerns about the relationship between knowledge (belief) and actions. Of most interest to us is the emphasis on situations involving the knowledge (belief) of a group of agents rather than that of a single agent. At the heart of any analysis of such situations as a conversation, a bargaining session or a protocol run by processes is the interaction between agents. An agent in a group must take into account not only events that have occurred in the world but also the knowledge of the other agents in the group.

In some cases we need to consider the situation that the agents has common-knowledge of an event; that is, simultaneously everyone knows the event, everyone knows that everyone knows the event, and so on. This notion also turns out to be a prerequisite for achieving agreement: In fact, Aumann [1] showed the famous agreement theorem; that is, if all posteriors of an event are common-knowledge among the agents then the posteriors must be the same, even when they have different private information. This is precisely what makes it a crucial notion in the analysis of an interacting group of agents.⁴

However, common-knowledge is actually so unfeasible a tool in helping us analyse complicated situations involving groups of agents, because the notion is defined by the infinite regress of all agents' knowledge as above, and thus we would like to remove it from our modelling.

⁴ C.f.: Fagin et al [4].

In this regard, Geanakoplos and Polemarchakis [5] investigated a communication process in which two agents announce their posteriors to each other. In the process agents learn and revise their posteriors and they reach consensus without common-knowledge of an event. Furthermore, Krasucki [6] introduced the revision process mentioned in the above. He showed that in the process, consensus on the posteriors can be guaranteed if the communication graph contains no cycle. The result is an extension of the agreement theorem of Aumann [1]. Recently, Ménager [8] extend his result for the decision function defined as the argmax of an expected utility. In the communication model of Krasucki [6] as same as Ménager, the agents sends his/her exact information on the decisions to the another agents. There is no possibility losing information.

All of the information structures in the models of Aumann [1], Geanakoplos and Polemarchakis [5] and Krasucki[6] are given by partition on a state space. Bacharach [2] showed that the partition model is equivalent to his knowledge operator model with the three axioms about the operators: **T** axiom of knowledge (what is known is true), **4** axiom of transparency (that we know what we do know) and **5** axiom of wisdom (that we know what we do not know.) He pointed out that the assumptions for the partition are problematic in decision making, and hence the model of analysing complicated situations should be also constructed without such strong assumptions.

Matsuhisa and Kamiyama [7] introduced the lattice structure of knowledge for which the requirements such as the three axioms are not imposed, and they succeeded in extending Aumann's theorem to their model. However the extension of agreement theorem are established under the common-knowledge (or common-belief) assumption.

Our concern in this article is to extend the communication model of Krasucki [6] through possibility messages. The emphasis is on that each agent sends not exact information on the membership value but fuzzy information on it . Under the circumstances we show Theorem 2, which is an extension of Krasucki [6], because the result of Krasucki [6] coincides with Theorem 2 when the message is not possibility set $F_i^t([d_i^t(X; \omega)])$ but exact one $[d_i^t(X; \omega)]$.

It is not so difficult to observe that Theorem 2 holds for the fuzzy structure associated with a reflexive and transitive information structure, where the reflexive and transitive information structure is equivalent to the Kripke semantics for the multi-modal logic $\mathbf{S4}_n$. However it is unknown at present time whether Theorem 2 is still valid without the acyclic condition. This is the agenda of future researches.

References

1. Aumann, R.J.: Agreeing to disagree. *Annals of Statistics* 4, 1236–1239 (1976)
2. Bacharach, M.: Some extensions of a claim of Aumann in an axiomatic model of knowledge. *Journal of Economic Theory* 37, 167–190 (1985)
3. Binmore, K.: *Fun and Games*, p. xxx+642. D.C. Heath and Company, Lexington (1992)

4. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about Knowledge. The MIT Press, Cambridge (1995)
5. Geanakoplos, J.D., Polemarchakis, H.M.: We can't disagree forever. *Journal of Economic Theory* 28, 192–200 (1982)
6. Krasucki, P.: Protocol forcing consensus. *Journal of Economic Theory* 70, 266–272 (1996)
7. Matsuhisa, T., Kamiyama, K.: Lattice structure of knowledge and agreeing to disagree. *Journal of Mathematical Economics* 27, 389–410 (1997)
8. Ménager, J.: Consensus, communication and knowledge: An extension with Bayesian agents. *Mathematical Social Sciences* 51, 274–279 (2006)
9. Parikh, R., Krasucki, P.: Communication, consensus, and knowledge. *Journal of Economic Theory* 52, 178–189 (1990)

Probabilistic Nogood Store as a Heuristic

Andrei Missine and William S. Havens

Simon Fraser University, 8888 University Drive, Burnaby, B.C. Canada V5A 1S6

Abstract. Nogood stores are frequently used to avoid revisiting states that were previously discovered to be inconsistent. In this paper we consider the usefulness of learned nogoods as a heuristic to guide search. In particular, we look at learning nogoods probabilistically and examine heuristic utility of such nogoods. We define how probabilistic nogoods can be derived from real nogoods and then introduce an approximate implementation. This implementation is used to compare behavior of heuristics using classic nogoods and then probabilistic nogoods on random binary CSPs and QWH problems. Empirical results show improvement in both problem domains over original heuristics.

1 Introduction

Interesting constraint satisfaction problems (CSPs) are generally hard to solve. Many infeasible states are discovered before finding a solution that satisfies all constraints. Such states are known as *nogoods*. Savings gained by keeping track of nogoods can be substantial as it may be exponentially expensive to discover a nogood, and not learning it can cause the solver to incur that cost again, possibly exponentially many times. Nogoods can be learned directly from constraint violations or by resolving nogoods from existing nogoods. There are a number of existing techniques that utilize nogoods and nogood stores. We will briefly discuss these in section 2 as well as other meta-heuristics that have been successfully applied to CSPs. Our focus in this paper will be on constructive search.

We propose keeping track of probabilistic as well as classic nogoods in order to gain better heuristic guidance from the nogood store. We will define probabilistic nogoods, how they are computed from previously learned classic nogoods and show how to generalize this computation to probabilistic nogoods in section 3. The resulting probabilistic nogood store can be used in the classical sense to prune search spaces, and can also be used to look up probability of a label to be nogood. This probability can then be used for heuristic guidance. These ideas cannot be efficiently implemented directly and thus need to be approximated. In section 4 we describe an augmented two watched literal scheme [8] that does so.

For empirical evaluation of our heuristic we have studied random binary CSPs and quasigroup with holes (QWH) completion instances, also known as Latin Squares. We have studied FCCBJ [11] and MACCBJ [10] as the core solvers. We applied FCCBJ to random binary CSPs and MACCBJ to QWH. The results are shown in section 5. Lastly, we conclude this paper with an overview of what

we have learned so far about probabilistic nogoods and describe when probabilistic nogoods appear to give the most significant improvement in section 6.

2 Related Work

Nogoods were originally introduced by Stallman and Sussman in [13] where they were used to avoid revisiting subspaces that were already deemed not to contain a solution and to enhance backtracking. More recently, an unbounded nogood store has been used to solve crossword puzzles and radio frequency allocation problems in [7], where nogoods are stored in an efficient nogood store inspired by CHAFF [8]. Keeping track of all nogoods can get expensive so techniques have been developed to only keep an interesting subset of learned nogoods. FCCBJ [11] is a simple example of such a technique where forward checking is used to both adjust future variable domains, and also to keep track of conflict sets that are used for conflict directed back jumping. These conflict sets basically represent the nogoods responsible for pruned future variable domains and since only one nogood is kept per domain reduction they are polynomially bounded in size. MACCBJ [10] is similar, but applies arc consistency instead of forward checking. More sophisticated bounded nogood stores are common in many SAT solvers such as CHAFF [8] where nogoods are recorded and discarded over time. We will outline an extended nogood store based on two watched literal scheme from CHAFF in section 4.

A simple heuristic that is frequently used in constructive search for variable selection is the first fail principle [5], i.e. *dom* heuristic, that selects the variable with the smallest domain first. There are also two existing heuristics that estimate the probability that a given singleton assignment will not be part of a solution based on modeling the constraint network as a Bayesian network and using belief propagation algorithm [9] to compute and update these probabilities. The first approach chooses to do loopy belief propagation to obtain probabilistic arc consistency [6] while the second approach decomposes the constraint network into multiple spanning trees [14] over which belief propagation is applied and combined to produce the final heuristic value. More recent heuristics try to take into account information that may be learned during search. For example, impact heuristic [12] computes the impact that a given singleton assignment has on the reduction of search space. In a recent paper, inspired by impacts, nogoods are used to approximate the effect of a particular assignment on the domain of other variables [2]. Lastly, *dom* heuristic has also been extended to *dom* / *wdeg* [1] to take into account the information about constraints violated during search and has been shown to perform very well.

We will now describe what probabilistic nogoods are and how they can be derived from classic and probabilistic nogoods. We then use probabilistic nogood estimates to augment first fail and impact heuristics and will empirically evaluate these augmentations in section 4.

3 Theory

We are interested in exploring the utility of probabilistic nogoods as a heuristic. We will begin by giving a few essential definitions and then outline the key ideas that describe what probabilistic nogoods are and how they can be used to provide heuristic guidance.

Definition 1. *Constraint Satisfaction Problem (CSP)* is defined as a tuple $\langle \mathbb{C}, \mathbb{V}, \mathbb{D} \rangle$ where \mathbb{C} is a set of constraints, \mathbb{V} is a set of variables and \mathbb{D} is a set of domains. Each variable $V \in \mathbb{V}$ is associated with a domain $V_D \in \mathbb{D}$. Each constraint $C \in \mathbb{C}$ is defined on a subset of variables $C_V \subseteq \mathbb{V}$ and is a function from values assigned to each variable in C_V to a boolean value that states whether or not the given set of values satisfies this constraint. The goal of solving a CSP is to find an assignment to all variables such that all constraints are satisfied. We will only consider CSPs with bounded discrete domains in this paper.

Definition 2. *Assignment*, or, more precisely, a singleton assignment, is an assignment of a value $i \in V_D$ to a variable $V \in \mathbb{V}$. We will use notation $V = i$ to represent assignments.

Definition 3. *Label* is a set of assignments, with each variable appearing at most once. A partial label is one where there is at least one variable $V \in \mathbb{V}$ that is not assigned; a complete label is one where all variables are assigned; lastly an empty label is one where no variables are assigned. A label L_1 is said to extend label L_2 if $L_1 \supset L_2$. For brevity, set operations on labels can be applied both on assignments, as in $L \cup \{A = 1\}$, meaning we extend L with assignment $A = 1$, and on variables, as in $L \setminus B$, meaning we remove the assignment involving B from L . Labels can be used to represent search subspaces; note that labels with fewer assignments thus correspond to larger search subspaces. For instance, the empty label represents the entire search space.

Definition 4. *Span of Label / Subspace Covered by a Label* is the subspace of the overall search space that the label subsumes. Given a label L , its span, or subspace covered by it, is defined as all complete labels that extend L .

$$\text{span}(L) = \left\{ \bigcup_{V \in \mathbb{V}, V \notin L, i \in V_D} \{V = i\} \cup L \right\} \quad (1)$$

Definition 5. *Probabilistic Nogood* is a tuple $\gamma = \langle L, p \rangle$ where L is a label and p is an estimate of the fraction of $\text{span}(L)$ that is nogood. Set operations on a nogood will refer to its label and arithmetic operations on a nogood will refer to its probability. We will use $\text{label}(\gamma)$ and $\text{prob}(\gamma)$ to refer to the label and probability of γ when the context is less obvious. We will use Γ to refer to the nogood store and it is simply a set of probabilistic nogoods.

Definition 6. *Intersection of Labels* is defined as the intersection of spans of the given labels. It is, itself, also a label. Given two labels L_1 and L_2 their intersection is given by the union of their assignment sets.

$$\text{intersect}(L_1, L_2) = L_1 \cup L_2 \quad (2)$$

Note that intersection is not defined when the given labels do not share any common subspaces. This occurs when there exists a variable that is assigned to different values in the argument labels to intersect. For example $\{A = 1\}$ and $\{B = 1\}$ intersect on $\{A = 1, B = 1\}$, but $\{C = 1, D = 1\}$ and $\{C = 2, D = 1\}$ do not intersect.

Let us consider what a particular probabilistic nogood γ tells us about some subspace L that is included in $\text{label}(\gamma)$. Based on the definition, $\text{prob}(\gamma)$ is the estimate of the fraction of subspace spanned by γ that is known to be nogood. In this paper we will only consider the uniform distribution assumption that states that γ contributes the same probability over its entire span¹. This is shown in the Equation 3.

$$\text{contrib}(\gamma, L) = \text{prob}(\gamma) \quad (3)$$

Classic nogood resolution occurs when for each value i in the domain of some variable V there is a nogood, containing $V = i$ as part of its label. A new nogood is resolved from these nogoods via Equation 4 - in other words the resulting label is the union of all singleton assignments of the argument nogoods, minus the assignment to V .

$$\text{resolve}_L(V, \Gamma) = \text{intersect}(\{\gamma_i \setminus V | \gamma_i \in \Gamma, i \in V_D\}) \quad (4)$$

For the probabilistic case, in addition to computing a label we also need to compute the probability of the resulting nogood based on the arguments². The label is computed as before, via Equation 4, and the probability is computed via Equation 5. This equation applies *contrib* operator to determine how much each nogood contributes separately, and then takes the average.

$$\text{resolve}_P(V, \Gamma) = \sum_{\gamma_i \in \Gamma, i \in V_D} \text{contrib}(\gamma_i, L \cup \{V = i\}) / |V| \quad (5)$$

Next, we need to define how to combine contributions from multiple nogoods to the same subspace. We chose to use maximum as it is less prone to overestimation. This is because we cannot rely on independence of nogoods since multiple nogoods may be derived from the same set of original nogoods, so using something like probabilistic *or* operator would result in overestimation.

$$\text{combine}(S) = \max(p \in S) \quad (6)$$

Lastly, we can use these to define how to compute the $\text{lookup}(L, \Gamma)$ operator that will be used to compute probability of label L given a collection of nogoods Γ in Algorithm 1. Algorithm 1 works by first identifying the subset S of Γ that has a non-empty intersection with L - these are the nogoods that actually contribute to L 's probability to be nogood. At this point, if all elements of S include L then

¹ Two other alternatives are to compute the lower or upper bound contribution, rather than assume uniform distribution.

² It is possible that there isn't a nogood for each of the values of V , i.e. the variable being "resolved out". In such cases we model these using a dummy probabilistic nogood with label $\{V = i\}$ and probability of zero, where i is the value of V for which we did not have a nogood before.

the result can be computed directly by calling *combine* on all contributions of $\gamma \in S$ onto L . Otherwise, there must exist one or more elements of S that overlap with L , but do not contain L . The algorithm picks an unassigned variable and resolves γ_V by recursively enumerating all assignments to V . Lastly, $prob(\gamma_V)$ is combined with contributions from nogoods that already include L and this value is returned. This value will be used to provide heuristic guidance.

Algorithm 1. *lookup*(L, Γ)

```

1:  $S \leftarrow \{\gamma | \gamma \in \Gamma, \text{intersect}(\gamma, L) \neq \text{nil}\}$ 
2: if  $\forall_{\gamma \in S} \text{contains}(\gamma, L)$  then
3:   return combine( $\{p | \gamma \in S, p = \text{contrib}(\gamma, L)\}$ )
4: else
5:    $V \leftarrow \text{unassignedVar}(L)$ 
6:   for all  $i \in V_D$  do
7:      $\gamma_{V_i} = \text{lookup}(L \cup \{V = i\}, S)$ 
8:   end for
9:    $\gamma_V = \text{resolve}(\{\gamma_{V_i} | i \in V_D\}, V)$ 
10:  return combine( $\{prob(\gamma_V)\} \cup \{p | \gamma \in S, \text{contains}(\gamma, L), p = \text{contrib}(\gamma, L)\}$ )
11: end if
```

The overall result of *lookup* is a probabilistic estimate of L to be nogood produced by attempting to resolve it from a probabilistic nogood store Γ . This value can then be used for heuristic guidance. For variable ordering the *first fail* principle states that we should pick the variable with the smallest live domain. Probabilistic estimates from *lookup* can be used to estimate the remaining live fraction of a particular value-variable pair to refine *dom* heuristic. For value selection *lookup* can be used to choose a value that is the least likely to be nogood; this is referred to as *promise* principle. These augmentations to *dom* heuristic plus an augmentation of impact heuristic [12] are empirically evaluated in section 5.

4 Implementation

The techniques we have discussed in the previous section describe how to compute the probability of a given label to be nogood given a probabilistic nogood store. Unfortunately, an exact implementation of these techniques would be too computationally expensive. Due to this fact we resort to making an approximation as follows: only nogoods that include the given label L and those that are off by one assignment will be considered. This is equivalent to bottoming out recursion of Algorithm 1 at depth one.

In order to account for nogoods that are off by two or more assignments we introduce nogood learning into the overall solver, thus when a variable is unassigned a new probabilistic nogood, summarizing the information from the previous search state, is learned. Such a nogood will only be learned if its probability is over a specified threshold, thus a classic nogood store can be obtained

by setting this threshold to 1 and a probabilistic nogood store by setting it to a value less than 1. This threshold prevents the nogood store from learning too many weak probabilistic nogoods. Nogoods that are subsumed by learned nogoods will be removed if their probability is less than or equal to the probability obtained from the learned nogood via *contrib* function (Equation 3).

We make the assumption that the search algorithm will always assign and unassign variables one at a time³ in order to efficiently maintain this set of contributing nogoods. We also assume that the search will use *lookup*, bottoming out at depth one, in order to obtain heuristics and will only call *lookup* with labels that extend the current search state with exactly one assignment.

Our implementation uses the two watched literal scheme [8] directly, but in addition we also allow nogoods to be in an *active* state, meaning that the search subspace we are currently exploring is nogood, with some probability. The reason for allowing this is because if this probability is not equal to 1 then there may be a solution in this subspace and it should be explored. Transition to this state occurs from *implied* state⁴ when the remaining watched literal of the nogood is assigned to the same value as the current state and thus the search state is now fully included in the nogood label. Transition out of this state occurs when the last assigned variable becomes unassigned and the nogood returns to *implied* state. This augmented scheme is then used to maintain what state each nogood is in. As nogoods enter *implied* and *active* states they begin to contribute to *lookup*(L, Γ): nogoods in *implied* state contribute to a single variable and single value (the one that is implied in the nogood) and nogoods in *active* state contribute to all possible extensions of the current solver state.

In order to reduce the run time and memory footprint of the nogood store we also limit its size. Each nogood is associated with a score that reflects its usefulness to the search. The score prefers shorter nogoods over longer nogoods and nogoods with high probability over those with low probability as these tell us more about the search space. The score is increased whenever the nogood is used to resolve another nogood. These scores are then used to determine which nogoods should be thrown out once the nogood store size limit is reached.

The nogood store also supports classic nogood resolution and keeps track of which nogoods are classic, and which are not. Classic nogoods are used to prune domains of unassigned variables, in addition to supplying heuristic guidance.

5 Experiments

The implementation we have described in the previous section has been used to study the effect of recording probabilistic nogoods versus regular nogoods on heuristic guidance provided by *lookup*. In this section we will describe the problem domains we have examined to date and will then present our results.

³ Multiple assignments / unassignments can be executed sequentially as singletons.

⁴ A nogood is in *implied* state when its label matches state of the solver C on all assignments but one, which is unassigned in C .

5.1 Problem Domains

The first problem is that of random binary CSPs. We studied random binary CSPs generated using flawless random problem generator (Model B) based on work of Ian Gent et al [3]. We studied problems with 40 variables, each having domain size 10 using FCCBJ. Parameters for this problem model beyond the number of variables, domain sizes of variables and a random seed are two probabilities that control how constrained the resulting problem will be: p_1 and p_2 . The first, p_1 , is the probability that there will be a constraint between a pair of variables. The second, p_2 , is the probability that for a given constraint and a pair of values the constraint will not be satisfied. The difficulty of such problems is typically summarized by a single parameter κ that is a function of the other parameters. Resulting problems can be either satisfiable or unsatisfiable. This problem domain has a phase transition where problems transition from satisfiable to unsatisfiable; in this phase transition region it is difficult to find a solution or to prove that one does not exist. We traverse this region by keeping p_1 fixed and varying p_2 .

The second problem domain is quasigroup with holes completion problems. These problems are defined by partially filled in Latin squares that need to be fully filled in. A Latin square is an n by n square where each cell can hold a number from 1 to n and a number must appear exactly once in each row and column. The problem parameters for the generator are the size of the square, a random seed, and a number of holes to produce in the resulting square. Instances were generated via `lsencode` generator [4]. Resulting problems are always satisfiable. This problem domain has a “easy-hard-easy” transition where problems with few empty cells are easy to solve, problems with approximately half of the cells filled in are hard, and problems with many empty cells are again easy to solve. We generated balanced instances, i.e. those where the number of holes in a given row or column is approximately the same as these are harder to solve [4].

We used the two watched-literal based implementation with nogood store size bound of 500 nogoods. For binary random CSPs we used FCCBJ [11] as the underlying search algorithm with *dom* [5] as the variable heuristic. We compared the performance of *dom* to our augmented *dom* heuristic, as outlined in section 3. For Latin squares we used MACCBJ and impact heuristic [12]; impact heuristic performs significantly better with MACCBJ than FCCBJ as stronger pruning produces better impact measures. Impact heuristic is used for both variable and value ordering on Latin Squares. We compared the performance of regular impacts to our augmented impact heuristic.

For each of the experiments we measured performance of the given heuristic, heuristic augmented with classic nogoods only, and heuristic augmented with probabilistic nogoods. Classic nogoods are always used for pruning, as well as for heuristic guidance supplied by *lookup*. Probabilistic nogoods were derived in one of two ways. The first is to learn a probabilistic nogood when at least half of values of a variable’s live domain are removed due to a consistency check. The second is to learn a nogood during backtracking; this nogood may be probabilistic

if restarts are employed and such nogoods are only learned if the probability is at least 0.01. In both cases, conflict sets were used to learn more compact (and thus more general) nogoods.

5.2 Results

We considered random binary CSPs with 40 variables, each having domain size of 10. We used FCCBJ with *dom* heuristic⁵ as the baseline and compared it against augmented first fail heuristic that also takes classic and probabilistic nogoods into account for variable ordering as discussed in section 3. Value ordering was obtained for probabilistic heuristic only by picking the value with the smallest probability to be nogood; random value ordering was used elsewhere. Restarts did not improve performance in this problem domain and thus were not used.

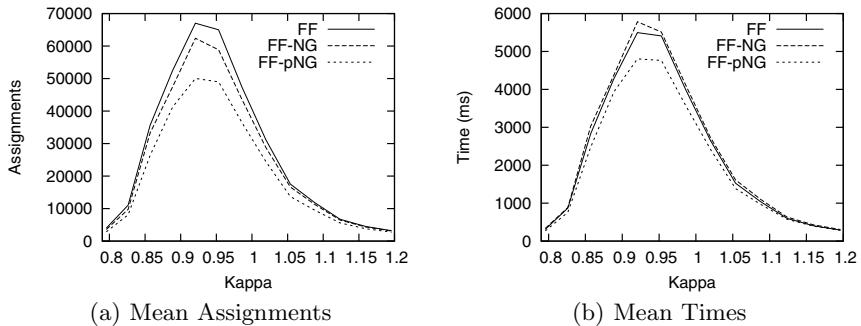
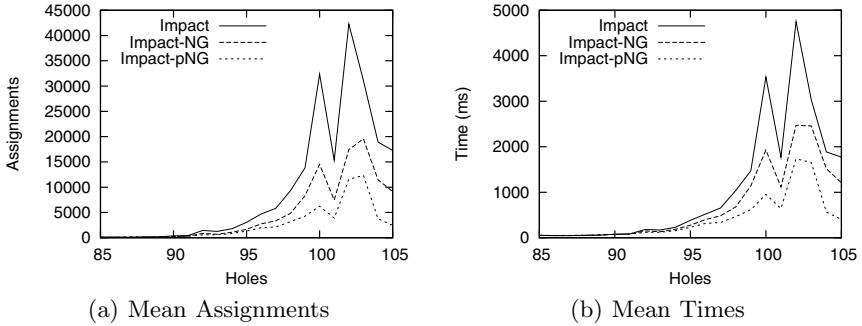


Fig. 1. FCCBJ and *dom* Heuristic

Figure 1 shows the mean number of assignments and times as the problem goes through the phase transition around $\kappa = 0.9$; each point is an average over 300 runs. The labels are as follows: “FF” stands for first fail, “FF-NG” stands for first fail using heuristic with classic nogoods only, and lastly “FF-pNG” stands for first fail using heuristic with probabilistic nogoods. From the first figure we can see that FF-NG requires fewer assignments than FF, and FF-pNG requires fewer assignments still. From the second figure we see that the overhead of maintaining nogoods actually makes FF-NG slower than FF, but since FF-pNG makes significantly fewer assignments than both FF and FF-NG its run time is better than both FF-NG and FF.

The other problem domain we examined is that of Latin squares. For this problem domain we looked at 14x14 Latin squares with number of holes ranging from 85 to 105. The underlying search algorithm used is MACCBJ with restarts and impact heuristic with initialization, as suggested in the original paper [12]. In this domain we found that the best performance gain was obtained by combining impact measure with result of *project* and assigning each equal weight of 0.5

⁵ We are currently investigating *dom* / *wdeg* heuristic [1]. Preliminary results on radio link frequency allocation problem instances are promising.

**Fig. 2.** MACCBJ and Impact Heuristic

whenever variable or value heuristic is required. The reason for doing so is to give impacts a sense of context, provided by the probabilistic nogood store, which is lost if impacts are summarized and only the average impact measure is used.

Our results on this problem domain are promising, with impacts augmented with probabilistic nogoods running more than twice as fast on the hard problem region (starting around 100 holes). Figure 2 shows mean assignments and times. Each point represents the average of 200 runs. It is interesting to note that the heuristic gives best guidance on instances where regular impacts tend to get lost in unpromising parts of the search tree. This suggests that probabilistic nogoods that are learned over time are more accurate than impacts on harder instances. Similar, but less extreme behavior is observed on random binary CSPs.

6 Conclusions

In this paper we have introduced the idea of probabilistic nogoods, outlined how they can be implemented and have shown that based on our empirical results they do appear to provide valuable heuristic guidance. Based on what we have observed so far and our intuition, we believe that probabilistic nogoods are best used on hard problem instances. Under such circumstances simpler heuristics lack learning, or oversimplify, as is the case with impact heuristic, and we believe it is exactly this intelligent summarization and learning that helps our heuristic outperform simpler heuristics. We are now exploring how probabilistic nogoods can be used to augment other known heuristics such as *dom* / *wdeg* heuristic [1] and are also experimenting on more structured, real-world based problems.

Acknowledgements. The authors would like to thank anonymous reviewers for their comments and suggestions. This work was supported by National Sciences and Engineering Research Council of Canada.

References

1. Boussemart, F., Hemery, F., Lecoutre, C., Sais, L.: Boosting systematic search by weighting constraints. In: ECAI, pp. 146–150 (2004)
2. Cambazard, H., Jussien, N.: Identifying and exploiting problem structures using explanation-based constraint programming. *Constraints* 11(4), 295–313 (2006)
3. Gent, I., MacIntyre, E., Prosser, P., Smith, B., Walsh, T.: Random constraint satisfaction: Flaws and structure (1998)
4. Gomes, C.P., Shmoys, D.: Completing quasigroups or latin squares: A structured graph coloring problem. In: Proc. Computational Symposium on Graph Coloring and Generalizations (2002)
5. Haralick, R.M., Elliott, G.L.: Increasing tree search efficiency for constraint satisfaction problems. *Artif. Intell.* 14(3), 263–313 (1980)
6. Horsch, M.C., Havens, W.S.: Probabilistic arc consistency: A connection between constraint reasoning and probabilistic reasoning. In: UAI 2000: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, pp. 282–290. Morgan Kaufmann Publishers Inc., San Francisco (2000)
7. Katsirelos, G., Bacchus, F.: Unrestricted nogood recording in csp search (2003)
8. Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S.: Chaff: Engineering an Efficient SAT Solver. In: Proceedings of the 38th Design Automation Conference, DAC 2001 (2001)
9. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
10. Prosser, P.: MAC-CBJ: maintaining arc consistency with conflict-directed back-jumping. Technical Report Research Report/95/177, Dept. of Computer Science, University of Strathclyde (1995)
11. Prosser, P.: Hybrid algorithms for the constraint satisfaction problem. *Computational Intelligence* 9(3), 268–299 (1993)
12. Refalo, P.: Impact-based search strategies for constraint programming. In: Wallace, M. (ed.) CP 2004. LNCS, vol. 3258, pp. 557–571. Springer, Heidelberg (2004)
13. Sussman, G.J., Stallman, R.M.: Forward reasoning and dependency directed backtracking in a system for computer aided circuit analysis. *Artificial Intelligence* 9, 135–196 (1977)
14. Vernooy, M., Havens, W.S.: An examination of probabilistic value-ordering heuristics. In: Australian Joint Conference on Artificial Intelligence, pp. 340–352 (1999)

Semantic Filtering for DDL-Based Service Composition

Wenjia Niu^{1,2}, Zhongzhi Shi¹, Peng Cao^{1,2}, Hui Peng^{1,2}, and Liang Chang^{1,2}

¹ Key Laboratory of Intelligent Information Processing, Institute of Computing Technology,
Chinese Academy of Sciences, 100190, Beijing, China

{Niuwenjia, Shizz, Caopeng, Pengh, Changl}@ics.ict.ac.cn

² Graduate School of the Chinese Academy of Sciences, 100039, Beijing, China
{Niuwenjia, Caopeng, Pengh, Changl}@ics.ict.ac.cn

Abstract. Dynamic description logic (DDL) provides a good logic-level solution among the few emerging service composition solutions through reasoning in AI area. To make DDL reasoning infrastructure practical, there is still unaddressed needs of increasing reasoning efficiency. We proposed a semantic filtering approach aiming to increase reasoning efficiency through decreasing reasoning space. The filtering approach was decomposed into two consecutive steps: context-based semantic retrieval with iRDQL —an imprecise query model, and processing of retrieval results under the control of workflow before forming final filtering results. Experimental results show that the method is well suitable for the volatile context-aware environment and yields good performance over DDL-based service composition.

1 Introduction

The combination of Semantic Web and Web services leads to a new direction of the next Web generation, i.e. Semantic Web Services (SWS), in which the semantic interoperability of Web Services is the emergence and evolution of the SWS. Semantic Web Services promise to change the way knowledge and business services are consumed and provided on the Web by augmenting the service description with rich semantics, which can be utilized by applications with less human assistance or less highly constrained agreements on interfaces or protocols.

As with Web services, Semantic Web enables ontology standards built on the foundation of URIs and XML Schema for describing the semantics of Web resources. The current components of the Semantic Web framework include RDF, RDF Schema (RDF-S) and the Web Ontology Language – OWL, among which OWL has been the W3C-recommended Web ontology language and adopted description logic (DL) as its logic basis. The DLs offer considerable expressive power going far beyond propositional logic and ensure reasoning decidable in the mean time. However, in Semantic Web Services, DLs cannot effectively represent and reason dynamic knowledge (e.g., a book-selling service itself). To overcome this shortcoming, researchers are working towards the characterization of dynamic information, with most research focusing on two approaches. The first is to construct upper ontology OWL-S [7]. The second approach aims to integrate action theory into DLs and construct a proper logic basis

and reasoning mechanism. Along the latter direction, an important effort—dynamic description logic (DDL) was proposed [2, 10] to represent and reason knowledge of static and dynamic, which integrate description logics with action formalisms. The DDL-based service composition proves to be a proper approach, which can achieve semantic composition of Web services through DDL reasoning.

As the service composition relies on the inferences drawn by the DDL reasoner, the practicality of the DDL-based solution crucially depends on the efficiency of the DDL reasoning. However, the time complexity of DDL reasoning is higher than the exponential-time complete. Due to low efficiency, it is hard to apply DDL-based composing approach for a large number of services. To decrease this high time complexity of DDL reasoning, one approach is that reasoner itself need improve reasoning algorithm based on heuristic rule ; And the other approach is decreasing the DDL reasoning space. In this paper, to decrease reasoning space we propose a bottom-up context-based semantic filtering approach, on which a new DDL-based model can be built to realize service composition in a more intelligent and efficient way.

The remainder of this paper is organized as follows. Section 2 presents a brief introduction of dynamic description logic (DDL). In Section 3 we explain our semantic filtering approach. In Section 4, we illustrate the usefulness of our approach through experimental analysis. Section 5 draws conclusions.

2 Backgrounds

The concept “action” of DDL is to be utilized for the filtering approach. For this purpose, it suffices to introduce the DDL in brief.

The DDL knowledge base consists of a TBox, an ABox and an ActionBox. The Tbox contains assertions about concepts (e.g., *Person*) and roles (e.g., *hasAge*). The ABox contains assertions about individuals (e.g., *PETER*). The ActionBox contains assertions about actions (e.g., *BookMovieTicket(PETER, TICKET)*). Concepts are inductively defined by a set of constructors, starting with a set N_C of concept names, a set N_R of role names, a set N_I of individual names. By these constructors, concepts are formed with the syntax rule: $C, C' \rightarrow C_i | \{p\} | <\pi>C | \neg C | C \cup C' | \exists R.C | \forall R.C$, where $C_i \in N_C, p \in N_I, R \in N_R, \pi$ is an action. Formulas of dynamic description logic are formed with the syntax rule: $\varphi, \varphi' \rightarrow C(p) | R(p, q) | \neg \varphi | \varphi \vee \varphi' | <\pi>\varphi$, where C is a concept, $p, q \in N_I, R \in N_R$, and π is an action. An atomic action is a pair (P, E) , where, P, E are two finite set of formulas used to describe precondition and effect accordingly. Actions are formed with the syntax rule: $\pi, \pi' \rightarrow (P, E) | \varphi? | \pi \cup \pi' | \pi, \pi' | \pi^*$, where (P, E) is an atom action, φ is a formula. Actions of the form $\varphi?$, $\pi \vee \pi'$, π , π^* are respectively named as testing, choice, sequential, and iterated actions. For a state u , the semantics are given via interpretation $I = <\Delta^{I(u)}, \bullet^{I(u)}>$, where $\Delta^{I(u)}$ is a non-empty set of objects, and $\bullet^{I(u)}$ maps each individual name to an element, each concept to a subset of $\Delta^{I(u)}$, and each role name to a binary relation on $\Delta^{I(u)}$. The semantics of concept constructors are as follows:

- $(\neg C)^{I(u)} = \Delta^{I(u)} - (C)^{I(u)}$
- $(\neg R)^{I(u)} = \Delta^{I(u)} \times \Delta^{I(u)} - (R)^{I(u)}$
- $(C \cap D)^{I(u)} = C^{I(u)} \cap D^{I(u)}$
- $(C \cup D)^{I(u)} = C^{I(u)} \cup D^{I(u)}$

- $(\exists R.C)^{I(u)} = \{x|\exists y, (x,y) \in R^{I(u)} \wedge y \in C^{I(u)}\}$
- $(\forall R.C)^{I(u)} = \{x|\forall y, (x,y) \in R^{I(u)} \rightarrow y \in C^{I(u)}\}$
- $(\langle \pi \rangle C)^{I(u)} = \{x|uT_\pi v, x \in C^{I(u)}\}$

The purpose of reasoning is to discover the sequence of services, while service can be put into a one-to-one correspondence with action in DDL, so action reasoning plays an important role in DDL reasoning. There are four kinds of action reasoning: realizability, executability, projection and plan.

Realizability: An action π is realizable w.r.t. the RBox D_R and TBox D_T iff there exists a model $M=(W,I)$ of both D_R and D_T such that there exists some states $w, w' \in W$ with $(w, w') \in \pi^I$.

Executability: An action π is executable on states described by D_S iff for any model $M=(W,I)$ of both D_R and D_T , and for any state $w \in W$ with $(M,w) \models D_S$, there exists a model $M'=(W',I')$ of both D_R and D_T , such that $W \subseteq W', I'(Wi) = I(Wi)$, for each $(M', w) \models D_S$, and $(w, w') \in \pi^I$ for some state $w' \in W'$.

Projection: A formula ψ is a consequence of applying π on states described by D_S iff for any model $M=(W,I)$ of both D_R and D_T , and for any states $w, w' \in W$: if $(M,w) \models D_S$ and $(w, w') \in \pi^I$, then $(M, w') \models \psi$.

Plan: Let ψ be a formula and Σ be a set of actions. Let π_1, \dots, π_n be a sequence of actions with each action coming from Σ . Then, the sequence π_1, \dots, π_n is a plan for ψ relative to D_S iff (i) the sequence-action π_1, \dots, π_n is executable on states described by D_S and (ii) ψ is a consequence π_1, \dots, π_n of applying on states described by D_S .

3 Context-Based Semantic Filtering

Context-based semantic filtering refers to filtering those services unsuitable for current contexts before built into DDL reasoning space. For example, one user's context is that he does not have passport, so those services which need user passport do not have to participate in reasoning.

3.1 Context Modeling

According to the role in Web service composition, we generalize three contexts: user context, provider context and broker context. Zimmermann defines an operational definition of context [11], in which the context is any information that can be used to characterize the situation of an entity. Elements for the description of this context information fall into five categories: individuality, activity, location, time, and relations. The activity predominantly determines the relevancy of context elements in specific situations, and the location and time primarily drive the creation of relations between entities and enable the exchange of context information among entities. According to the operational definition above, the service description (e.g. input, output etc.) can be considered as an activity context. By adding activity context into context

family, the context information can be used not only for personalized application but also for functionally composing Web services.

Traditionally, most context formalization approaches, such as comprehensive structured context profiles (CSCP) [4], only focus on common context such as time, location, preference and profile etc.. In order to better characterize context information especially dynamic context information such as activity context for the specific background of Web service, we propose a new context formalization approach by extending the context predicate model in [8] based on DDL's action representation. The context is denoted as three tuples:

ContextType(Action,Time,Role), Where:

- Action: a pair (P,E), where, P ,E are two finite set of formulas used to describe precondition and effect
e.g. $\text{buyCD}(\text{Tom}, \text{Love}) \equiv \{\text{customer}(\text{Tom}), \text{cd}(\text{Love}), \text{instore}(\text{Love}), \neg\text{bought}(\text{Tom}, \text{Love})\}, \{\text{bought}(\text{Tom}, \text{Love}), \neg\text{instore}(\text{Love})\}$;
- Time: the time point at which the action ends;
- Role: the entity with which the context keeps true;
- ContextType: a name of context type.

For examples:

- 1) Location(enterRoom(Tom, Room), 14:20, User1);
- 2) Activity(buyCD(Bob, Love), 18:00, User2);

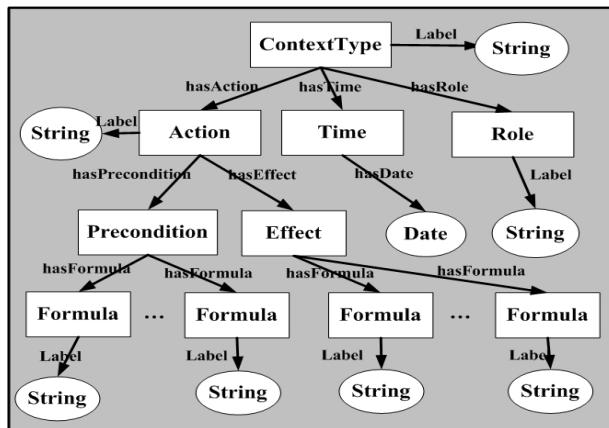


Fig. 1. Context Ontology Representation

Finally, these contexts are storied in OWL files and the ontology representation of context is shown in Figure 1.

3.2 Semantic Retrieval with iRDQL

Since services are modeled as activity contexts, the filtering problem is how to retrieve those satisfied activity contexts and filtering unsatisfied ones based on other context

information(e.g. location, user profile). The action precondition and effect in activity context are both made up of strings of formulas. As a result, precise querying the activity context will probably make the user buried in results or with no results whatsoever. In the case of too many answers or no answer, it is a common approach that using imprecise retrieval based on similarity when no precise matches to the query.

We exploit the iRDQL[1], a semantic web query language with support for similarity joins[3]. The iRDQL aims to extend RDF Data Query Language (RDQL) [9] in order to enable the user to query for similar resources of ontology.

Among similarity measures implemented in the generic java library *SimPack* of iRDQL, the similarities between vectors, strings, trees and objects have been realized well. In this paper, we focus on utilizing the sequence-based measure which makes use of underlying ontology graph to realize semantic retrieval of activity context and experiments show this measure is suitable for our filtering approach. In the sequence-based measure, the resource precondition or effect is considered as starting node to traverse the graph along its edges where edges are properties and then mapped into a feature set according to the traverse sequence. Figure 2 shows the sample of activity context and preference context.

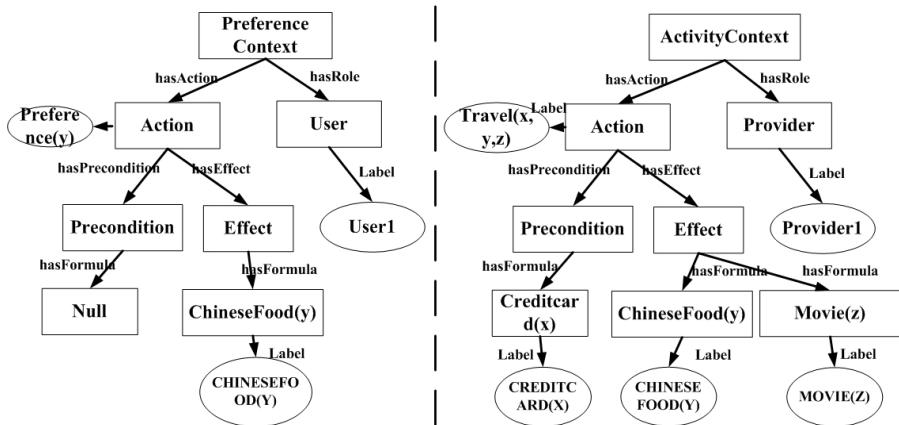


Fig. 2. Partial Ontology Graph Representation of Preference Context and Activity Context

The action effect in preference context and the one in activity context results in the following two feature sets:

$$Rx = \{\text{Effect}, \text{hasFormula}, \text{ChineseFood}(y), \text{Label}, \text{CHINESEFOOD}(Y)\}$$

$$Ry = \{\text{Effect}, \text{hasFormula}, \text{ChineseFood}(y), \text{Label}, \text{CHINESEFOOD}(Y), \text{hasFormula}, \text{Movie}(z), \text{Label}, \text{MOVIE}(Z)\}$$

The similarity calculation *Levenshtein edit distance*[6] between strings are leveraged to evaluate the similarity between the vectors of strings. The number of insert, remove, and replacement operations to transform vector \mathbf{x} to vector \mathbf{y} is defined as $xform(\mathbf{x}, \mathbf{y})$, the worst case transformation cost(replace all concept parts of \mathbf{x} with parts of \mathbf{y} , then delete the remaining parts of \mathbf{x} , and insert additional parts of \mathbf{y}) is defined as $xform_{wc}(\mathbf{x}, \mathbf{y})$. Supposing the operations $c(delete) + c(insert) \geq c(replace)$ and

each operation has equal weight **1**, the similarity between the vectors of strings is calculated as follows:

$$\text{Sim}_{\text{levenshtein}}(R_x, R_y) = 1 - \frac{\text{xform}(x, y)}{\text{xform}_{wc}(x, y)} \quad (1)$$

For the example in Figure 2, the similarity between the vector x and y is $1-4/9=5/9=0.56$.

To illustrate the querying process, we consider activity context and preference context (see Figure 2) are stored in a Jena model and then give our two queries using extended iRDQL shown in Figure 3:

<pre> SELECT ?C1 ?R1 ?C2 ?R2 WHERE (C1 hasAction ?A1) (A1 hasPrecondition ?P1) (P1 hasFormula Null) (C1 hasEffect ?E1) (C1 hasRole ?R1) (C2 hasAction ?A2) (A2 hasPrecondition ?P2) (C2 hasRole ?R2) IMPRECISE ?E1 ?P2 SIMMEASURE Levenshtein OPTIONS IGNORECASE false THRESHOLD 0.5 </pre>	<pre> SELECT ?C1 ?R1 ?C2 ?R2 WHERE (C1 hasAction ?A1) (A1 hasPrecondition ?P1) (P1 hasFormula Null) (C1 hasEffect ?E1) (C1 hasRole ?R1) (C2 hasAction ?A2) (A2 hasEffect ?E2) (C2 hasRole ?R2) IMPRECISE ?E1 ?E2 SIMMEASURE Levenshtein OPTIONS IGNORECASE false THRESHOLD 0.5 </pre>
(a)	(b)

Fig. 3. The iRDQL Queries for Activity Context Retrieval (a) Using Action Precondition Matching (b) Using Action Effect Matching

The queries above look for activity contexts whose precondition or effect is similar with the action effect of preference context. The similarity between $?E1$ and $?P2$, $?E1$ and $?E2$ is computed using the Levenshtein string edit distance and is returned with the possible combinations of $?C1$, $?R1$, $?C2$ and $?R2$ as shown in Table 1 and Table 2. String comparison is case sensitive. A threshold of 0.5 is used in this example expressing that two vectors are equal if their similarity is at least 0.5.

Table 1. Output of the Query in Fig.3(a)

C1	R1	C2	R2	Sim
Preference Context	User	Preferecne Context	User	1

Table 2. Output of the Query in Fig.3(b)

C1	R1	C2	R2	Sim
Preference Context	User	Preferecne Context	User	1
Preference Context	User	Activity Context	Provider	0.56

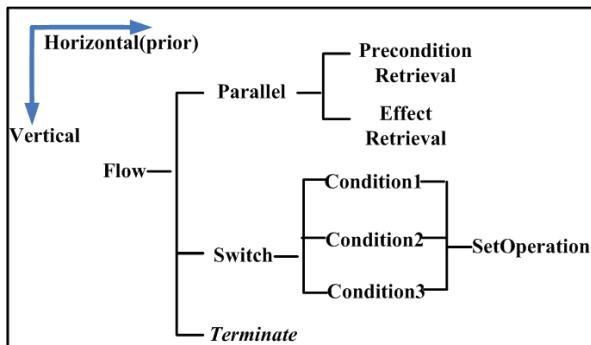
3.3 Filtering Flow

As mentioned in Section 3.1, activity context can be functionally described by action precondition and effect. Filtering activity contexts can be realized based on semantic retrieval of action precondition or effect. To ensure filtering flexibility and efficiency, we divide filtering process into several sub tasks and utilize workflow to orchestrate these tasks to realize filtering in concurrent and logic-based way.

In our filtering process, there are three primitive tasks named *PreconditionRetrieval*, *EffectRetrieval* and *SetOperation* separately. *PreconditionRetrieval* and *EffectRetrieval* are concurrently executed, which aim to use iRDQL to retrieve activity context based on precondition matching or effect matching separately. The flow control “Switch” is used to decide which activity context to be built into reasoning space depending on the condition. Task *SetOperation* decides the final set of activity contexts to be not filtered. We define the retrieval set based on precondition matching as PS, the retrieval set based on effect matching as ES, and then the *SetOperation* can be defined as follows:

$$\text{SetOperation} = \begin{cases} \text{Result-set}=\emptyset & \text{if } \text{PS}=\emptyset \\ \text{Result-set}=\emptyset & \text{if } \text{ES}=\emptyset \\ \text{Result-set}=\text{PS} \cup \text{ES} & \text{if } \text{PS} \neq \emptyset \text{ and } \text{ES} \neq \emptyset \end{cases}$$

Workflow should be designed and the structure of process is shown in Figure 4. The process starts from the root activity and then goes to the lower levels until reaching the terminate point.

**Fig. 4.** The Filtering Workflow

4 Experiments

In order to evaluate the usefulness of our semantic filtering approach, we need to assess it by (1) choosing a set of services in a specific knowledge domain and (2) specifying retrieval queries based on specific context and (3) finally having the filtering results and DDL reasoning results. By statistical analysis of these results, we provide a sense of the utility of our approach.

4.1 Dataset of Activity Contexts

We choose the OWL-S-TC-v1 [5], a collection specifying a set of 406 OWL-S services of six different domains (i.e., communication, economy, education, food, medical, and travel). Based on the 58 sample services of the travel domain in [5], 90 services of this domain are manually constructed by adding some context information as our knowledge dataset. For the 90 services of travel domain, they fall into three sub-domains: food services, hotel services and traffic services. Then the totally 90 services are going to be transformed into activity contexts for supporting the evaluation of the performance of our filtering approach. For instance, a service called `City_Broker_Service` that should provide the hotel reservation service for a given city will be transformed into an activity context named `City_Broker_Service_Activity.owl` file-type. Finally we give the specific context-based query for filtering activity contexts corresponding to each sub-domain.

4.2 Experimental Setup

The following steps are necessary to evaluate our approach:

- 1) All activity contexts of the same sub-domain are loaded into a Jena model through a one by one way.
- 2) Corresponding to each sub-domain, we give a specific query file based on context (such as profile context or preference context), which is also loaded into Jena model.
- 3) Two kinds of iRDQL statement are automatically generated based on precondition and effect matching separately. For each iRDQL query the Levenshtein similarity measure as explained in Section 3.2 is used. As a result, when searching for activity contexts of a specific sub-domain, the query is intended to find and rank all satisfied activity contexts suitable for current query context.
- 4) After Step 3, for each sub-domain, there exists two semantic retrieval sets PS and ES mentioned in Section 3.3, though the processing on the result sets under the control logic of workflow we designed, the final filtering result set is formed.
- 5) After semantic filtering, we run the DDL reasoner to find the composed service plans through DDL reasoning.

4.3 Filtering Result Analysis

The activity context number of food, hotel and traffic domain is 20, 30 and 40 separately. In the semantic filtering process, the strength of filtering may have two influences. On one hand, filtering too much can make the reasoning hard to generate

enough service composition plans for optimal selection later, and on the other hand filtering too little still puts more efficiency burden on reasoning.

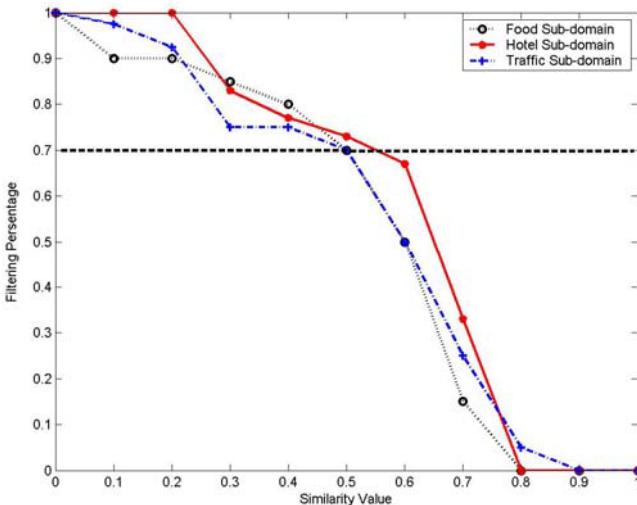


Fig. 5. The Filtering Percentage for Each Sub-domain

To ensure a proper similarity value threshold, we evaluate the filtering percentage (unfiltered number/total number) of each sub-domain with similarity value changing from 0 to 1. As Figure 5 shown, the percentage will decrease rapidly and drop to 70% when the similarity value is larger than 0.5. This states that 0.5 is a proper similarity value for the experiment by our approach. Furthermore, by analyzing the results we also found similarity 0.5 as our retrieval threshold used in our approach can lead to satisfied filtering results from case study view and the final filtering results show that 7 activity contexts in the food domain were filtered out, 8 in the hotel domain were filtered out and 12 were filtered out in the traffic domain. As a result, the dataset of 90 activity contexts has 63 left.

Without filtering, the DDL reasoning for 90 services costs 16565 milliseconds. However, after filtering out 27 services, the reasoning time drops to 5034 milliseconds. Ignoring the filtering process, which takes a smaller scale time than large scale reasoning, our approach can decrease 69.61% time cost than the reasoning without filtering.

5 Conclusions

In this paper, we proposed a semantic filtering approach to decrease DDL reasoning space for increasing the reasoning efficiency. In our approach, a context model is utilized to modeling the context information around Web service composition for providing context-based filtering. By semantic query with iRDQL and then the results refinery under the workflow control, we build a filtered DDL reasoning space. Experiments show our approach provides a good solution.

Acknowledgement. This work is supported by the National Science Foundation of China (No. 90604017, 60435010, 60775035,), 863 National High-Tech Program (No.2007AA01Z132), National Basic Research Priorities Programme (No. 2003CB317004, 2007CB311004) and National Science and Technology Support Plan (No.2006BAC08B06).

References

1. Bernstein, A., Kiefer, C.: iRDQL-Imprecise RDQL Queries Using Similarity Joins. In: K-CAP 2005 (2005)
2. Chang, L., Lin, F., et al.: Dynamic Description Logic for Representation and Reasoning About Actions. In: Zhang, Z., Siekmann, J.H. (eds.) KSEM 2007. LNCS (LNAI), vol. 4798, pp. 115–127. Springer, Heidelberg (2007)
3. Cohen, W.W.: Data Integration Using Similarity Joins and a Word-Based Information Representation Language. ACM Trans. Inf. Syst. 18(3), 288–321 (2000)
4. Held, A., Buchholz, S., et al.: Modeling of context information for pervasive computing applications. In: Proceedings of SCI 2002/ISAS 2002 (2002)
5. Klusch, M.: OWLS-TC-v1: OWL-S Service Retrieval Test Collection (2005),
<http://projectssemmwebcentral.org/projects/owlstc/>
6. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady 10, 707–710 (1966)
7. OWL-S: Semantic Markup for Web Services (OWL-S 1.0),
<http://www.daml.org/services/owl-s/1.0/owl-s.html>
8. Ranganathan, A., Campbell, R.H., et al.: ConChat: A Context-Aware Chat Program. IEEE Pervasive Computing 1, 51–57 (2002)
9. Seaborne, A.: RDQL – A Query Language for RDF (2004),
<http://www.w3.org/Submission/RDQL/>
10. Shi, Z., Dong, M., et al.: A Logic Foundation for the Semantic Web. Science in China, Series F 48(2), 161–178 (2005)
11. Zimmermann, A., Lorenz, A., et al.: An Operational Definition of Context. In: Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R., Vieu, L. (eds.) CONTEXT 2007. LNCS (LNAI), vol. 4635, pp. 558–571. Springer, Heidelberg (2007)

Prediction of Protein Functions from Protein Interaction Networks: A Naïve Bayes Approach

Cao D. Nguyen¹, Kathleen J. Gardiner², Duong Nguyen³, and Krzysztof J. Cios^{1,4}

¹ Virginia Commonwealth University, USA

² University of Colorado Denver, USA

³ Raytheon, USA

⁴ IITiS PAN, Poland

{cdnguyen, kcios}@vcu.edu, kathleen.gardiner@uchsc.edu,
dnguyen1@raytheon.com

Abstract. Predicting protein functions is one of most challenging problems in bioinformatics. Among several approaches, such as analyzing phylogenetic profiles, homologous protein sequences or gene expression patterns, methods based on protein interaction data are very promising. We propose here a novel method using Naïve Bayes which takes advantage of protein interaction network topology to improve low-recall predictions. Our method is tested on proteins from the Human Protein Reference Database (HPRD) and on the yeast proteins from the BioGRID and compared with other state-of-the-art approaches. Analyses of the results, using several methods that include ROC analyses, indicate that our method predicts protein functions with significantly higher recall without lowering precision.

Keywords: protein function, protein interaction networks, Bayes methods.

Supplementary Materials: www.egr.vcu.edu/cs/dmb/Bayesian.

1 Introduction

To discover how proteins function within the living cell is one of the central goals for life scientists. Genome sequences have been published at a dramatic rate but a large fraction of newly discovered genes have no functional characterization. For example, in a simple organism, such as baker's yeast, approximately one third of the proteins have no functional annotation. For more complex organisms, functional annotation is lacking for a much larger fraction of the proteome. Because experimental determination of protein function is expensive, successful predictive methods have an important role to play.

Several computational methods for protein function prediction have been developed. Conceptually simplest, homologous proteins are identified in protein databases by using protein sequence similarity and functions are assigned to the query protein based on the known functions of the matches [8]. Another approach [9] infers protein interactions from genomic sequences using the observation that some pairs of interacting proteins have homologies in another organism fused into a single protein chain; the functional relatedness of

some such protein pairs has been confirmed. [1] and subsequently other groups [2,3,4,5], inferred functional similarity among proteins based on phylogenetic profiles of orthologous proteins. Grouping proteins by correlated evolution, correlated messenger RNA expression patterns plus patterns of domain fusion have been successfully applied to yeast proteins [6]. Correlations between genes that have similar expression patterns are used to detect similar functions [7]. Other approaches integrate similarly heterogeneous types of high-throughput biological data for protein function prediction. Bayesian reasoning has been used to combine large-scale yeast two-hybrid (Y2H) screens and multiple microarray analyses [10]. Gene functional annotations were identified from a combination of protein sequence and structure data by support vector machines [11]. While each approach has had some success, in general these methods are severely limited by low reliability.

Use of protein-protein interaction (PPI) data to annotate protein function has been extensively studied. Proteins interact with each other for a common purpose and thus a protein may be annotated by information on the functions of its interaction neighbors. Large scale protein physical interaction data have been generated by high-throughput experiments for *worm* [12], *fly* [13], *yeast* [14,15,16,17], and *human* [18]. Applied to the yeast PPI network, the *Majority* method assigns functions to a protein using the most frequent annotations among its nearest neighbors [15]. The drawback of the *Majority*, however, is that some functions may have a very high frequency in the network but are not assigned if they do not occur in the nearest neighbor set. The *Majority* method was extended in [19] to predict functions by exploiting indirect neighbors and using a topological weight to estimate functional similarity. Another approach [20] makes use of χ^2 statistics by looking at all proteins within a specified radius thus taking into account the frequency of all proteins having a particular function. However, the χ^2 statistics does not take into account the underlying topology of the PPI network. Global optimization approaches based on Markov random fields (MRF) and belief propagation in PPI networks [21,22,23] assign functions based on a probabilistic analysis of graph neighborhoods in the network. These methods assume that the probability distribution for the annotation of any node is conditionally independent of all other nodes, given its neighbors. The methods are sensitive to the neighborhood size and the parameters of prior distribution. *FunctionalFlow* [24] considers each protein of known function as a source of functional flow for that function. The functional flow spreads through the neighborhoods of the sources. Proteins receiving the highest amount of flow of a function are assigned that function. This algorithm does not consider indirect flow of functions to other proteins after labeling the functions. Other global approaches integrate PPI network with more heterogeneous data sources. *ClusFCM* [25] assigns biological homology scores to interacting proteins in a PPI network and performs an agglomerative clustering on the weighted network to cluster the proteins by known functions and cellular location. Functions then are assigned to proteins by a fuzzy cognitive map technique. The MRF methods were extended by combining PPI data, gene expression data, protein motif information, mutant phenotype data, and protein localization data to specify which proteins might be active in a given biological process [26,27].

In this work we use the Naïve Bayes method that takes into account the underlying topology of a PPI network. For each protein we analyze the predicted functions by using association rules to discover interesting relationships among the assigned functions, i.e., when one set of functions occurs in a protein then the protein may be annotated with an additional set of other specific functions at some confidence level. We test our method on the PPI networks of yeast and human proteins, and compare its performance with the *Majority* and χ^2 *statistics* methods.

2 Materials

Yeast Interaction Dataset

We use the yeast molecular interaction network from the BioGRID database [28] (release 4/2008, version 2.0.39). After eliminating direct interactions based solely on high-throughput Y2H assays because of noise levels [29], the yeast dataset includes 39,128 direct molecular interactions. The yeast dataset without Y2H comprises a total of 3,727 unique yeast proteins, of which 3,724 proteins are annotated with 3,000 distinct GO functions from the GO database [30]. Among the 3,000 GO terms there are 1,182 molecular functions, 494 cellular component functions, and 1,324 biological process functions.

Human Interaction Dataset

The human interaction data was obtained from the HPRD [31] (release 9/2007). The entire dataset contains 37,106 direct molecular interactions from three types of experiments (*in vivo*, *in vitro* and *in Y2H*). There are 9,463 distinct proteins annotated with 424 GO functions in three categories: 201 molecular functions, 150 biological process functions and 73 cellular component functions. We limit the HPRD data by excluding direct interactions supported only by Y2H experiments. The HPRD dataset without the Y2H comprises 28,148 interactions from *in vivo* and *in vitro* experiments. In this dataset, of 411 GO functions annotating the 7,764 unique proteins, there are 195 molecular functions, 143 biological process functions and 73 cellular component functions. The statistics are shown in *Table 1* in Supplementary Materials.

3 Methods

3.1 Definitions

The PPI network is described as an undirected graph $G=(V,E)$, where V is a set of proteins and E is a set of edges connecting proteins u and v if the corresponding proteins interact physically. We use the following notation. K : the total number proteins in the PPI network, F : the whole GO function collection set, $|F|$: the cardinality of the set F , f_i : a function in the set F ($i=1..|F|$), C_u : the cluster coefficient of protein u , N_u : the neighbor set of protein u (proteins interacting directly with protein u), $N_u^{f_i}$: the number of proteins annotated with

function f_i in N_u and $N_u^{\bar{f}_i}$: the number of proteins un-annotated with function f_i in N_u where $|N_u| = N_u^{f_i} + N_u^{\bar{f}_i}$.

3.2 Predictive Modeling

Multinomial Naïve Bayes

We first explain the basic idea of our approach. For a function of interest f_i , we want to annotate the function f_i to the proteins in a PPI network. We pose the functional annotation problem as a classification problem. The training data are available in the form of observations $d \in \mathbb{R}^k$ (k dimensions) and their corresponding class. For each protein u in the network, a function of interest f_i is considered as a class label 1 if the protein u is annotated with f_i , and otherwise as 0. Below we discuss how to select features to deduce class information. Exploiting the fact that proteins of known functions tend to cluster together [15], the first feature we take into account, A_1 , is the number of proteins annotated with the function f_i in the neighborhood set of protein u (i.e. $A_1 = N_u^{f_i}$). The second feature (A_2) is the number of proteins unannotated with the function f_i in the neighborhood set of the protein u (i.e. $A_2 = N_u^{\bar{f}_i}$). Figure 1 illustrates an observation in the HPRD without Y2H data for *Acetylcholine acetylhydrolase* protein (gene symbol: *ACHE*, HPRD id: 00010) with function *Extracellular region (GO: 0005576, cellular component)*.

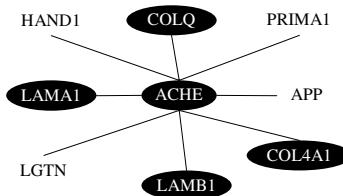


Fig. 1. Proteins annotated with GO:0005576 are black. The observation for the ACHE is $A_1=4$, $A_2=4$ and class=1 because there are 4 proteins annotated with GO:0005576 in the neighborhood set, the other 4 proteins are not annotated and the ACHE is itself annotated with GO:0005576.

Several studies indicate that other features can be useful to predict functions and drug targets for a protein, such as the number of functions annotated in proteins in the neighborhood set at *level 2* of the protein [19,20], *the connectivity* (the total number of incoming and outgoing arc of a protein, which is equal to $N_u^{f_i} + N_u^{\bar{f}_i}$), *the betweenness* (the number of times a node appears in the shortest path between two other nodes) and *the clustering coefficient C_u* (the ratio of the actual number of direct connections between the neighbors of protein u to the maximum possible number of such direct arcs between its neighbors) [31]. We include those features in our experiments and perform classification using Radial Basis Function network, Support Vector Machine and Logistic Regression (data not shown). To our surprise, the Multinomial Naïve Bayes using only three features $A_1 = N_u^{f_i}$, $A_2 = N_u^{\bar{f}_i}$ and $A_3 = C_u$ performed best. Next, we briefly explain how the Multinomial Naïve Bayes method is used in

our study. If $d = \langle A_1, A_2, A_3 \rangle$ is an observation for a protein u and we decide a class membership for the observation d (corresponding to a function of interest f_i) by assigning d to the class with the maximal probability computed as follows:

$$\mu(d, f_i) \propto \operatorname{argmax}_{c \in \{0,1\}} \hat{P}(c | d, f_i) \propto \operatorname{argmax}_{c \in \{0,1\}} \frac{\hat{P}(d | c, f_i) \hat{P}(c | f_i)}{\hat{P}(d | f_i)} \quad (1)$$

Note that since $P(d | f_i)$ can be ignored as it is the same for all classes:

$$\mu(d, f_i) \propto \operatorname{argmax}_{c \in \{0,1\}} \hat{P}(d | c, f_i) \hat{P}(c | f_i) \quad (2)$$

The likelihood $P(d | c, f_i)$ is the probability of obtaining the observation d for a protein u in class c and is calculated as:

$$\hat{P}(d | c, f_i) = (N_u^{f_i} + N_u^{\bar{f}_i} + C_u)! \frac{\hat{P}(A_1 | c, f_i)^{N_u^{f_i}}}{N_u^{f_i}!} \frac{\hat{P}(A_2 | c, f_i)^{N_u^{\bar{f}_i}}}{N_u^{\bar{f}_i}!} \frac{\hat{P}(A_3 | c, f_i)^{C_u}}{C_u!} \quad (3)$$

Thus equation (2) becomes:

$$\begin{aligned} \mu(d, f_i) &\propto \operatorname{argmax}_{c \in \{0,1\}} (N_u^{f_i} + N_u^{\bar{f}_i} + C_u)! \frac{\hat{P}(A_1 | c, f_i)^{N_u^{f_i}}}{N_u^{f_i}!} \frac{\hat{P}(A_2 | c, f_i)^{N_u^{\bar{f}_i}}}{N_u^{\bar{f}_i}!} \\ &\quad \frac{\hat{P}(A_3 | c, f_i)^{C_u}}{C_u!} \hat{P}(c | f_i) \end{aligned} \quad (4)$$

Since the factorials in equation (4) are constant, we can rewrite the *maximum a posteriori* class c as follows:

$$\mu(d, f_i) \propto \operatorname{argmax}_{c \in \{0,1\}} \hat{P}(A_1 | c, f_i)^{N_u^{f_i}} \hat{P}(A_2 | c, f_i)^{N_u^{\bar{f}_i}} \hat{P}(A_3 | c, f_i)^{C_u} \hat{P}(c | f_i) \quad (5)$$

Two key issues arise here. First, the problem of zero counts can occur when given class and feature values never appear together in the training data. It can be problematic because the resulting zero probabilities will wipe out the information in all other probabilities. We use the Laplace correction to avoid the problem [32]. Second, in equation (5), the conditional probabilities are multiplied and this can result in a floating point underflow. Therefore, it is better to perform the computation by using logarithms of probabilities instead of simply multiplying the probabilities. Equation (5) can be calculated as:

$$\begin{aligned} \mu(d, f_i) &\propto \operatorname{argmax}_{c \in \{0,1\}} \exp [N_u^{f_i} \log \hat{P}(A_1 | c, f_i) + N_u^{\bar{f}_i} \log \hat{P}(A_2 | c, f_i) + \\ &\quad C_u \log \hat{P}(A_3 | c, f_i)] \hat{P}(c | f_i) \end{aligned} \quad (6)$$

The parameters of the model, in our case, $\hat{P}(A_1 | c, f_i)$, $\hat{P}(A_2 | c, f_i)$, $\hat{P}(A_3 | c, f_i)$ and $\hat{P}(c)$ can be estimated as follows:

$$\begin{aligned} \hat{P}(A_1 | c=1, f_i) &= (\sum N_u^{f_i} + l_{c_1}) / (\sum N_u^{f_i} + N_u^{\bar{f}_i} + C_u + l_{c_1}) \\ &\text{where } u \in \{\text{proteins annotated with } f_i\} \end{aligned} \quad (7)$$

$$\hat{P}(A_1 \mid c=0, f_i) = (\sum N_u^{f_i} + lc_1) / (\sum N_u^{f_i} + N_u^{\bar{f}_i} + C_u + lc_1) \quad (8)$$

where $u \in \{\text{proteins not annotated with } f_i\}$

$$\hat{P}(c=1 \mid f_i) = (\mid \text{proteins annotated with } f_i \mid + lc_1) / (K + lc_2) \quad (9)$$

$$\hat{P}(c=0 \mid f_i) = (\mid \text{proteins not annotated with } f_i \mid + lc_1) / (K + lc_2) \quad (10)$$

where $lc_1=1$ and $lc_2=2$ are the Laplace corrections and the attributes A_2 and A_3 can be similarly estimated.

Below we briefly describe the *Majority* and χ^2 statistics methods used for comparison with the Multinomial Naïve Bayes method.

Majority: For each protein u in a PPI network, we count the number of times each function $f_i \in F$ occurs in neighbors of the protein u . The functions with the highest frequencies are assigned to the query protein u .

χ^2 statistics: For each function of interest f_i we derive the fraction π_{f_i} (number of proteins annotated with function f_i / K). Then, we calculate e_{f_i} as the expected number for a query protein u annotated with f_i : $e_{f_i} = N_u \pi_{f_i}$. The query protein u is annotated with the function with the highest χ^2 value among the functions of all proteins in its neighbors, where $\chi^2 = (N_u^{f_i} - e_{f_i})^2 / e_{f_i}$.

3.3 Assessment of Prediction

We use the *leave-one-out* method to evaluate predictions performed by each method. For each query protein u in a PPI network we assume that it is unannotated. Then, we use the above methods to deduce the protein functions for protein u . Let A be the annotated function set and P be the predicted function set. We calculate the number of true positive (TP), false positive (FP), true negative (TN) and false negative (FN) as follows: $TP = |A \cap P|$, $FP = |P \setminus A|$, $FN = |A \setminus P|$ and $TN = |F \setminus (A \cup P)|$. The following measures are used for assessing performance of the methods: *precision*, *recall*, *Matthews correlation coefficient (MCC)* and *harmonic mean (HM)* [33,35,36,37].

4 Results and Discussion

We implemented the *Multinomial Naïve Bayes*, *Majority* and χ^2 statistics methods in Java and tested them on three datasets: yeast and human, with and without interactions determined by Y2H. We examine predictions based on the entire GO function set, separately for each of the three categories: *biological process*, *cellular component* and *molecular function*. To compare the performance of our method we use implicit thresholds, namely, we normalize the posterior probability of a query protein u annotated with the function f_i : $P(c=1 \mid d, f_i)$ and decide the protein u to be annotated with the function f_i if the normalized $P(c=1 \mid d, f_i) > \tau$, where τ assumes a value between 0 and 1, in increments of 0.1.

In our method, we assume that a newly annotated protein can flow its newly acquired function(s) to its direct neighbors. Thus, the method is repeated in two iterations. In the second iteration, to calculate the value $A_i = N_u^{f_i}$ for a protein u , we count both the number of proteins in its neighborhood annotated with f_i and *predicted with f_i in the first iteration*. For the *Majority* and χ^2 *statistics* methods we select top k functions having the highest scores (k ranges from 0, 1, ... to 20) and assign these functions to the query protein. For each method, we choose the threshold which yields the highest HM measure. Interestingly, we found that with the selected thresholds the MCC values also achieve the highest value for each method. Figure 2 shows relationship between precision and recall using different thresholds for the normalized probabilities of query proteins in the HPRD *without Y2H* dataset.

The thresholds resulting in the highest HM measures in the Yeast *without Y2H*, HPRD and HPRD *without Y2H* datasets are .2, .3, and .3, respectively. Table 2 in Supplementary Materials shows performance of the algorithm. Note that functional annotations for the proteins are incomplete at present. Therefore, a protein may have a function that has not yet been experimentally verified. We wish to decrease the number of annotated functions that are not predicted and increase the number of predicted functions that are actually annotated. The fact that values of recall are always higher than the values of precision in all datasets increases confidence in our method.

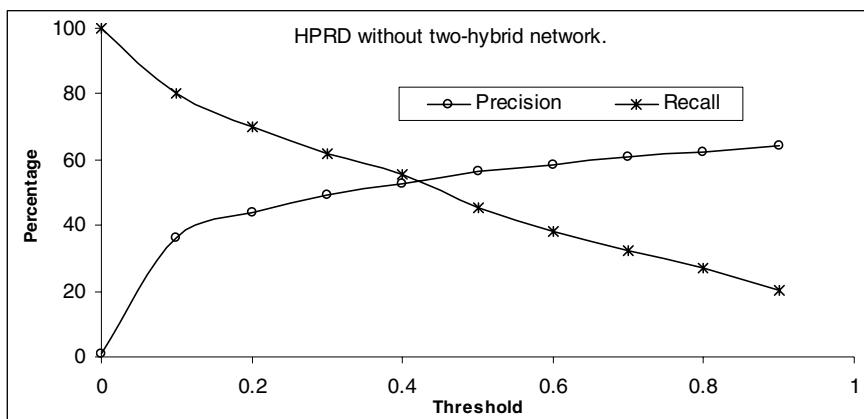


Fig. 2. Precision and recall results for the multinomial Naïve Bayes prediction on the HPRD *without Y2H* dataset

The results of Naïve Bayes in the three categories: *biological process*, *cellular component* and *molecular* are shown in Figures 3, 4 and 5 for the three datasets (see Supplementary Materials). We observe that in the *cellular component* and *biological process*, the Naïve Bayes performs better than in the molecular functions. This confirms the fact that PPI networks are more reflective of *cellular component* and *biological process*. Performance measures of our method at the selected thresholds are shown in Table 3 in Supplementary Materials, for each category. The comparison of Multinomial Naïve Bayes method with *Majority* and χ^2 *statistics* is shown in Figure 6. It

shows that for any given precision, the recall of Naïve Bayes outperforms the recalls of the other methods. At the selected thresholds (*Majority*'s are 8,3,4 and χ^2 statistics's are 13,10,10 for Yeast without Y2H, HPRD and HPRD without Y2H networks, respectively) performance measures of the three methods are shown in Table 4. Comparison of the ROC curves is shown in Figure 7 in Supplementary Materials. The closer the curve follows the top left-hand area of the ROC space the more accurate the classifier. A random classifier would have its ROC lying along the diagonal line connecting points (0,0) and (1,1). The Naïve Bayes performs consistently better on the three datasets for functional prediction.

Next, we ask the question: if the predicted functions $\{f_X\}$ appear together in a protein can we derive other functions $\{f_Y\}$? To answer this question we use association rule learning [38] to discover potentially interesting relationships between functions in PPI networks. Association rules are statements of the form $\{f_X\} \Rightarrow \{f_Y\}$, meaning that if we find all of $\{f_X\}$ in a protein, we have a good chance of finding $\{f_Y\}$ with some user-specified confidence (the probability of finding $\{f_Y\}$ given $\{f_X\}$) and support (the proportion of proteins containing functions $\{f_X\}$ in the entire networks). With 0.1% support and 75% confidence, we found 837 association rules in Yeast without Y2H, 1,504 in HPRD, and 1,837 in HPRD without Y2H (see Table 5 in Supplementary Materials for details). We derive new functions from the predicted functions of the three methods by using the mined rules and three axioms [39]: 1. if $X \supseteq Y$ then $X \rightarrow Y$, 2. if $X \rightarrow Y$ then $XZ \rightarrow YZ$ for any Z , and 3. if $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$. Interestingly, the performance measures for *Majority* and χ^2 statistics improved, as can be seen in Table 6 (see Supplementary Materials), while this is not the case for the Naïve Bayes. Based on that observation we believe that our method is able to find hidden correlations among functions. As mentioned above the functional annotation of proteins is incomplete, particularly for human protein data. This suggests possibility that the predicted functions for proteins that are now false positive may actually be yet-to-be-discovered true positive. We list in Table 7 in Supplementary Materials all the proteins from HPRD without Y2H network classified with functions at very high probabilities (>.9) but termed as "false" at the present.

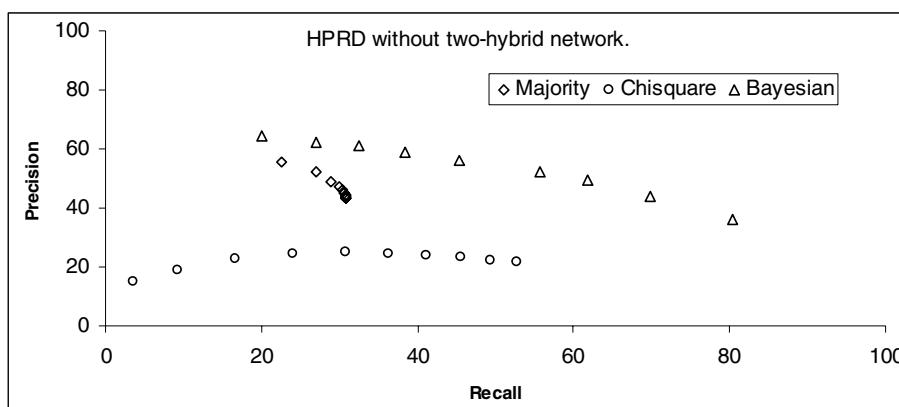


Fig. 6. Precision and recall of the three methods on the HPRD without Y2H network

Table 4. Performance of the three methods on three datasets using leave-one-out validation; (1): *Multinomial Naïve Bayes* (2): *Majority* (3): χ^2 statistics

	Yeast without Y2H			HPRD			HPRD without Y2H		
	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
Precision	0.29	0.17	0.13	0.47	0.42	0.20	0.49	0.47	0.22
Recall	0.54	0.15	0.31	0.58	0.25	0.51	0.62	0.30	0.53
MCC	0.40	0.16	0.20	0.52	0.32	0.31	0.55	0.37	0.33
HM	0.38	0.16	0.19	0.52	0.32	0.28	0.55	0.37	0.31

4 Conclusions

We introduce a novel method based on the Multinomial Naïve Bayes for protein function predictions. Our algorithm uses a global optimization approach that takes into account several characteristics of interaction networks: direct and indirect interactions, underlying topology (cluster coefficients), and functional protein clustering. We have shown robustness of our method by testing it on three interaction datasets using the leave-one-out cross-validation. Results show that the Multinomial Naïve Bayes consistently outperforms the *Majority* and the χ^2 -statistics methods for prediction of protein functions. In addition, we discovered hidden relationships among the predicted functions by using association rule learning; we believe that it finds new functions of proteins.

Acknowledgments. The authors acknowledge support from the National Institutes of Health (1R01HD05235-01A1) (KG and KC) and Vietnamese Ministry of Education (CN).

References

1. Pellegrini, M., Marcotte, E., Thompson, M., Eisenberg, D., Yeates, T.: Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. Proc. Natl Acad. Sci. USA 96, 4285–4288 (1999)
2. Bowers, P., Cokus, S., Eisenberg, D., Yeates, T.: Use of logic relationships to decipher protein network organisation. Science 306, 2246–2259 (2004)
3. Pagel, P., Wong, P., Frishman, D.: A domain interaction map based on phylogenetic profiling. J. Mol. Biol. 344, 1331–1346 (2004)
4. Sun, J., Xu, J., Liu, Z., Liu, Q., Zhao, A., Shi, T., Li, Y.: Refined phylogenetic profiles method for predicting protein–protein interactions. Bioinformatics 21, 3409–3415 (2005)
5. Ranea, J., Yeats, C., Grant, A., Orengo, C.: Predicting Protein Function with Hierarchical Phylogenetic Profiles: The Gene3D Phylo-Tuner Method Applied to Eukaryotic Genomes. PLoS Comput. Biol. 3(11), e237 (2007)
6. Marcotte, E., Pellegrini, M., Thompson, M., Yeates, T., Eisenberg, D.: A combined algorithm for genome-wide prediction of protein function. Nature 402, 83–86 (1999)
7. Zhou, X., Kao, M., Wong, W.: Transitive functional annotation by shortest-path analysis of gene expression data. Proc. Natl. Acad. Sci. USA 99, 12783–12788 (2002)
8. Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25, 3389–3402 (1997)

9. Marcotte, E., Pellegrini, M., Ng, H., Rice, D., Yeates, T., Eisenberg, D.: Detecting protein function and protein–protein interactions from genome sequences. *Science* 285, 751–753 (1999)
10. Troyanskaya, O., Dolinski, K., Owen, A., Altman, R., Botstein, D.: A Bayesian framework for combining heterogeneous data sources for gene function prediction (in *Saccharomyces cerevisiae*). *Proc. Natl. Acad. Sci. USA* 100, 8348–8353 (2003)
11. Lewis, D., Jebara, T., Noble, W.: Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure. *Bioinformatics* 22, 2753–2760 (2006)
12. Li, S., et al.: A map of the interactome network of the metazoan *C.elegans*. *Science* 303, 540–543 (2004)
13. Giot, L., et al.: A protein interaction map of *Drosophila melanogaster*. *Science* 302, 1727–1736 (2003)
14. Fromont-Racine, M., et al.: Toward a functional analysis of the yeast genome through exhaustive Y2H screens. *Nat. Genet.* 16, 277–282 (1997)
15. Schwikowski, B., Uetz, P., Fields, S.: A network of protein-protein interactions in yeast. *Nature Biotechnology* 18, 1257–1261 (2000)
16. Uetz, P., et al.: A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. *Nature* 403, 623–627 (2000)
17. Ho, Y., et al.: Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature* 415, 180–183 (2002)
18. Peri, S., et al.: Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Research* 13, 2363–2371 (2003)
19. Chua, H., Sung, W., Wong, L.: Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. *Bioinformatics* 22, 1623–1630 (2006)
20. Hishigaki, H., Nakai, K., Ono, T., Tanigami, A., Takagi, T.: Assessment of prediction accuracy of protein function from protein-protein interaction data. *Yeast* 18, 523–531 (2001)
21. Deng, M., Zhang, K., Mehta, S., Chen, T., Sun, F.: Prediction of protein function using protein-protein interaction data. *Journal of Computational Biology* 10, 947–960 (2003)
22. Letovsky, S., Kasif, S.: Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics* 19, 197–204 (2003)
23. Vazquez, A., Flammi, A., Maritan, A., Vespignani, A.: Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology* 21(6), 697–670 (2003)
24. Nabieva, E., Jim, K., Agarwal, A., Chazelle, B., Singh, M.: Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics* 21, 302–310 (2005)
25. Nguyen, C., Mannino, M., Gardiner, K., Cios, K.: ClusFCM: An algorithm for predicting protein functions using homologies and protein interactions. *J. Bioinform. Comput. Biol.* 6(1), 203–222 (2008)
26. Deng, M., Chen, T., Sun, F.: An integrated probabilistic model for functional prediction of proteins. *Journal of Computational Biology* 11, 463–475 (2004)
27. Nariai, N., Kolaczyk, E.D., Kasif, S.: Probabilistic Protein Function Prediction from Heterogeneous Genome-Wide Data. *PLoS ONE* 2(3), e337 (2007)
28. Stark, C., Breitkreutz, B.J., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M.: BioGRID: a general repository for interaction datasets. *Nucleic Acids Res.* 34, D535–D539 (2006)
29. Sprinzak, E., Sattath, S., Margalit, H.: How reliable are experimental protein–protein interaction data? *Journal of Molecular Biology* 327, 919–923 (2003)

30. Ashburner, M., et al.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* 25, 25–29 (2000)
31. Yao, L., Rzhetsky, A.: Quantitative systems-level determinants of human genes targeted by successful drugs. *Genome Res.* 18(2), 206–213 (2008)
32. Niblett, T.: Constructing decision trees in noisy domains. In: *Proceedings of the Second European Working Session on Learning*, pp. 67–78. Sigma, Bled, Yugoslavia (1987)
33. van Rijsbergen, C.: Information retrieval: theory and practice. In: *Proceedings of the Joint IBM/University of Newcastle upon Tyne Seminar on Data Base Systems*, pp. 1–14 (1979)
34. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.: Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics* 16, 412–424 (2000)
35. Kurgan, L., Cios, K., Scott, D.: Highly Scalable and Robust Rule Learner: Performance Evaluation and Comparison. *IEEE Transactions on Systems Man and Cybernetics, Part B* 36(1), 32–53 (2006)
36. Cios, K., Kurgan, L.: CLIP4: Hybrid Inductive Machine Learning Algorithm that Generates Inequality Rules. *Information Sciences* 163(1-3), 37–83 (2004)
37. Cios, K., Pedrycz, W., Swiniarski, R., Kurgan, L.: *Data Mining A Knowledge Discovery Approach*. Springer, Heidelberg (2007)
38. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. In: *SIGMOD Conference*, pp. 207–216 (1993)
39. Armstrong, W.: Dependency Structures of Data Base Relationships. In: *Information Processing* 74. North Holland, Amsterdam (1974)

Multi-class Support Vector Machine Simplification

DucDung Nguyen¹, Kazunori Matsumoto¹, Kazuo Hashimoto²,
Yasuhiro Takishima¹, Daichi Takatori², and Masahiro Terabe²

¹ KDDI R&D Laboratories Inc.

Fujimino-city, Saitama, 356-8502 Japan

{dd-nguyen,matsu,takisima}@kddilabs.jp

² Graduate School of Information Sciences, Tohoku University

Sendai-city, Miyagi, 980-8579 Japan

takatori@shino.ecei.tohoku.ac.jp,

{terabe,kh}@aiet.ecei.tohoku.ac.jp

Abstract. In support vector learning, computational complexity of testing phase scales linearly with number of support vectors (SVs) included in the solution – support vector machine (SVM). Among different approaches, reduced set methods speed-up the testing phase by replacing original SVM with a simplified one that consists of smaller number of SVs, called *reduced vectors* (RV). In this paper we introduce an extension of the bottom-up method for binary-class SVMs to multi-class SVMs. The extension includes: calculations for optimally combining two multi-weighted SVs, selection heuristic for choosing a good pair of SVs for replacing them with a newly created vector, and algorithm for reducing the number of SVs included in a SVM classifier. We show that our method possesses key advantages over others in terms of applicability, efficiency and stability. In constructing RVs, it requires finding a single maximum point of a one-variable function. Experimental results on public datasets show that simplified SVMs can run faster original SVMs up to 100 times with almost no change in predictive accuracy.

Keywords: kernel-based methods, support vector machines, reduced set method.

1 Introduction

In support vector learning [1], [2], reduced set method is an effective solution for speeding up support vector machine (SVM) in testing phase. The main reason is that the complexity of SVMs scales linearly with number of SVs included in their solution: to test a new data the main computational power is spent for comparing it with all support vectors (SVs) via a kernel function. Reducing this number of comparison leads to a speeding-up rate for the testing phase.

There have been several algorithms proposed for SVM simplification. Their common objective is to approximate the original norm of SVM's hyper plane by a new one that involves a small number of new vectors, called *reduced vectors* (RVs). The first algorithm introduced in [3] constructs the RVs incrementally: it finds the first

vector to approximate the whole SVM solution, or the norm vector of the hyper plane in feature space, and then iteratively finds other RVs to approximate the difference between the original solution and reduced solution. The authors in [4] extend this method for multi-class SVM classifier: using the same tactic to construct RVs for each binary-class SVM, and then share it to others by retraining all of them. To avoid the local minima problem the author proposed to apply a differential evolution algorithm in finding RVs. However, as mentioned in the paper, the algorithm may fail to converge quickly. In our point of view, this instability might be caused by the nature of local minima problem.

In this work we propose to extend the bottom-up method for simplifying binary SVM in [5] to multi-class case. The main idea is iteratively selecting two SVs and replacing them with a newly constructed RV. We will show that the RVs could be optimally constructed via finding a single maximum point of a one variable function. This property ensures the stability and efficiency of the bottom-up approach to multi-class SVMs simplification. Experimental results on different datasets indicate that the proposed algorithm can reduce a big number of SVs while keeping predictive performance of simplified SVMs unchanged.

The rest part of this paper is organized as follows. In section 2 we review different methods for constructing RVs, for both binary-class and multi-class cases. We present our proposed method for simplifying SVMs in section 3. In section 4 we report our experiment for evaluation and comparison with other method on benchmark datasets. Summarization and open research issues are given in section 5.

2 Simplification of Support Vector Machine

2.1 Support Vector Machine

SVMs work in some feature space F via a kernel function $K(x, y) = \Phi(x) \cdot \Phi(y)$ where $\Phi : D \rightarrow F$ is a map from input space D to feature space F . To classify a new object vector x , SVMs compares x with all SVs included in the decision function via kernel function K

$$f(x) = \text{sign} \left(\sum_{i=1}^{N_s} \alpha_i K(x_i, x) + b \right) \quad (1)$$

As support vector learning is principally designed for binary-class classification, more than one SVM are required to form a classifier for a multi-class application. The most popular way is to use T binary one-versus-rest SVMs or $T(T-1)/2$ one-versus-one SVMs, where T is the number of classes. In other words, the final decision is based on not one, but a set of T functions

$$f_t(x) = \sum_{i=1}^{N_s} \alpha_{ti} K(x_i, x) + b_t, t = 1, \dots, T \quad (2)$$

In (2), $\alpha_{ti} \neq 0$ means vector x_i is one SV of the t^{th} SVM and its corresponding weight is α_{ti} , otherwise $\alpha_{ti} = 0$ means x_i is not related to the t^{th} SVM.

For both binary-class and multi-class SVM, the most expensive procedure in testing a new object vector x is to compare it with the whole set of SVs via kernel function K . This computation scales linearly with the number of SVs N_s . To reduce this computation cost, or to speed-up the testing phase, reduced set method tries to replace N_s , number of original SVs, by N_z , a smaller number of new vectors, called *reduced vector* set. The decision functions then become ($T = 1$ for the binary-class case)

$$f'_{t_i}(x) = \sum_{i=1}^{N_z} \beta_{ti} K(z_i, x) + b_t, t = 1, \dots, T \quad (3)$$

In the following sub-sections we review different methods for constructing reduced vectors, in both binary-class and multi-class applications.

2.2 Top-Down Approach to SVM Simplification

The first reduced set method described in [3] builds RVs incrementally. It starts from finding the first vector to replace the whole original solution - a linear combination of SVs in feature space. It then iteratively finds another RV to approximate the difference between original SVM and reduced SVM. Call $\Psi = \sum_{i=1}^{N_s} \alpha_i \Phi(x_i)$ be the norm vector of the original hyper plane, each reduced vector z_m is found by solving the following optimization problem

$$(z_m, \beta_m) = \arg \min_{z, \beta} \rho = \|\Psi_m - \beta \Phi(z)\| \quad (4)$$

where Ψ_m are complement vectors of the reduced norm

$$\Psi_m = \Psi - \sum_{i=1}^{m-1} \beta_i \Phi(z_i) \quad (5)$$

The main drawback of this method is that it may suffer from numerical instability and get trapped in a local minimum of function ρ . To prevent this circumstance, the finding for each new vector must be repeated many times with different initial values [3], [6].

The method described in [4] extends this approach to multi-class SVM. Its main steps include:

- i) Create one reduced vector for each SVM
- ii) Combine all reduced vectors into a single list and retrain their coefficients for each SVM using all reduced vector
- iii) While more vector are desired
 - Use a heuristic to determine which SVM to improve. Find a new reduced vector for this SVM
 - Share all reduced vector and retrain all coefficients for all SVMs

To avoid the local minima problem, the authors proposed to use a differential evolution algorithm in finding reduced vectors. However, as mentioned in the paper, it occasionally fails to converge quickly. In our opinion, this phenomenon is due to the nature of multi minima problem in RV construction.

2.3 Bottom-Up Approach to SVM Simplification

To overcome the local minima problem, the bottom-up method described in [5] proposed to iteratively select two SVs and replace them with one combined vector. Call (x_i, x_j) be the selected pair of SVs, the combined vector z for replacing x_i and x_j is found by solving

$$\min \leftarrow \|\beta\Phi(z) - (\alpha_i\Phi(x_i) + \alpha_j\Phi(x_j))\|^2 \quad (6)$$

The main advantage of this approach is that solving (6) is much easier than (4). However, the method is only applicable to binary-class SVM, or each SV is related to only one coefficient. For multi-class applications, multi SVM must be used and therefore, one SV may relate to more than one SVM with different weights. In the next section we describe how to extend the bottom-up method for multi-class SVM simplification.

3 Bottom-Up Method to Multi-class SVM Simplification

3.1 Multi-weighted Support Vector Combination

Supposing that we want to replace two multi-weighted SVs (x_i, α_{ii}) and (x_j, α_{jj}) with a single new vector (z, β_t) , $t = 1, \dots, T$, the 2-norm optimal solution for all single SVMs will be the one that minimizes

$$\min \leftarrow \sum_{t=1}^T \|\beta_t\Phi(z) - (\alpha_{ti}\Phi(x_i) + \alpha_{tj}\Phi(x_j))\|^2 \quad (7)$$

The next two propositions will show how to find the optimal reduced vector z for the two most widely used kernels: Gaussian and polynomial. All it requires is to find an unique maximum point of a one-variable function on $[0,1]$.

Proposition 1. For Gaussian kernel $K(x, y) = \exp(-\gamma\|x - y\|^2)$, the 2-norm optimal vector z in (7), given fixed values of β_t , $t = 1, \dots, T$, is calculated as following:

$$z = \kappa x_i + (1 - \kappa)x_j, \quad (8)$$

where κ is the maximum point of

$$h(k) = \sum_{t=1}^T \left(m_t K_{ij}^{(1-k)^2} + (1 - m_t) K_{ij}^{k^2} \right) \quad (9)$$

with $m_t = \alpha_{ti}/(\alpha_{ti} + \alpha_{tj})$, $t = 1, \dots, T$, $K_{ij} = K(x_i, x_j)$.

Proof: Call $M_t = m_t\Phi(x_i) + (1 - m_t)\Phi(x_j)$, $\alpha_t^m = (\alpha_{ti} + \alpha_{tj})$, problems (7) becomes

$$\min \leftarrow \sum_{t=1}^T \|\beta_t\Phi(z) - \alpha_t^m M_t\|^2 \quad (10)$$

For Gaussian kernel, Φ maps input vectors into surface of the unit hyper sphere in feature space, or $\|\Phi(z)\| = 1$ for every z . $\|M_t\|$ are constants because (x_i, α_{ti}) and (x_j, α_{tj}) are pre-determined. If the scalars β_t is fixed then problem (10) is equivalent to finding vector z such that

$$\max \leftarrow \sum_{t=1}^T \Phi(z) \cdot M_t \quad (11)$$

For the extremum, we set derivative of the objective function on the right side of (11) to be zero

$$\begin{aligned} 0 &= \nabla_z \left(\sum_{t=1}^T \Phi(z) \cdot M_t \right) \\ &= 2 \sum_{t=1}^T m_t K(z, x_i)(x_i - z) + 2 \sum_{t=1}^T (1-m_t) K(z, x_j)(x_j - z) \end{aligned} \quad (12)$$

This leads to

$$z = kx_i + (1-k)x_j \quad (13)$$

where

$$k = \frac{\sum_{t=1}^T \alpha_{ti} K(x_i, z)}{\sum_{t=1}^T \alpha_{ti} K(x_i, z) + \sum_{t=1}^T \alpha_{tj} K(x_j, z)} \quad (14)$$

Equation (12) means that the optimal z is linearly dependent with two given vectors x_i and x_j in the input space. Replacing (13) into (11) and defining,

$$h(k) = \sum_{t=1}^T \Phi(z) M_t = \sum_{t=1}^T \left(m_t K_{ij}^{(1-k)^2} + (1-m_t) K_{ij}^{k^2} \right) \quad (15)$$

we arrive at the result that the optimal vector z of (7) is calculated via the maximum point κ of $h(k)$

$$\begin{aligned} z_{opt} &= \kappa x_i + (1-\kappa)x_j, \\ \kappa &= \arg \max_k h(k) \end{aligned} \quad (16)$$

Proposition 2. For polynomial kernel $K(x, y) = (x \cdot y)^p$, the 2-norm optimal vector z in (7), given fixed values of β_t , $t = 1, \dots, T$, is calculated as following:

$$z = \left(\frac{\sum_{t=1}^T \|M_t\|}{T} \right)^{1/p} \frac{z^*}{\|z^*\|} \quad (17)$$

Where $z^* = \kappa x_i + (1-\kappa)x_j$ and κ is the maximum point of

$$\begin{aligned} \kappa &= \arg \max_k g(k) = \sum_{t=1}^T \|M_t\| u(k) v_t(k), \\ u(k) &= \frac{1}{[x_i^2 k^2 + 2x_i x_j k(1-k) + x_j^2 (1-k)^2]^{p/2}}, \\ v_t(k) &= \frac{\alpha_{ii} [x_i^2 k + x_i x_j (1-k)]^p + \alpha_{jj} [x_i x_j k + x_j^2 (1-k)]^p}{(\alpha_{ii} + \alpha_{jj})} \end{aligned} \quad (18)$$

Proof: In (10), we want to find one input vector z that its image $\Phi(z)$ approximates T feature space vectors M_t , $t=1,\dots,T$. Without any loss in generality we assume that $\Phi(z)$ has an average length of M_t

$$\|\Phi(z)\| = \frac{1}{T} \sum_{t=1}^T \|M_t\| \quad (19)$$

and find the solution for (11). According to Lemma 1 in [5], the optimal vector z is also linearly dependent with x_i and x_j in the input space. Replacing this relation into (11) and solve (11) with constraint (19), we found that optimal vector z is calculated via the maximum point κ of function $g(k)$ in (18).

The key point of proposition 1 and 2 is that $h(k)$ in (9) and $g(k)$ in (18) are both single variable functions with unique maximum point on $[0,1]$. In our implementation it only takes several iterations for the parabolic interpolation method [8] to find their maximum value. It means that SVs can be constructed very efficiently.

Once reduced vector z is found, its optimal coefficients for each individual SVM are calculated (kernel independent) by the following proposition.

Proposition 3[5]. Given z , the optimal coefficients β_t , $t = 1,\dots,T$, for approximating the sum of two feature space vectors $\alpha_{ii}\Phi(x_i) + \alpha_{jj}\Phi(x_j)$ by $\beta_t\Phi(z)$ in (7) is

$$\beta_t = \frac{\alpha_{ii} K(x_i, z) + \alpha_{jj} K(x_j, z)}{K(z, z)} \quad (20)$$

3.2 Adjusting All Reduced Vectors and Recalculating Coefficients

The combination scheme described in section 3.1 aims at constructing one new vector to replace one selected pair of SVs. The combination criterion, or the objective function in (7), is locally optimized for the two SVs in pair. However, the ultimate goal of the simplification is to keep the simplified solution as similar to the original solution as possible. To improve this similarity, we can adjust all reduced vectors globally with respect to norms of solution's hyper planes by minimizing the difference between them:

$$\min \leftarrow \rho = \sum_{t=1}^T \left\| \sum_{i=1}^{N_s} \alpha_{ii} \Phi(x_i) - \sum_{i=1}^{N_z} \beta_{ii} \Phi(z_i) \right\|^2 \quad (21)$$

In our implementation, we apply the gradient descent and bisection methods [8] for minimizing ρ with respect to all RVs z_i , $i=1,\dots,N_Z$. The search directions for Gaussian RBF and polynomial kernels are:

$$\frac{\partial \rho_{RBF}}{\partial z_i} = \sum_{t=1}^T \left(\sum_{j=1}^{N_Z} -2\gamma\beta_{ti}\beta_{tj}K(z_i, z_j)(z_i - z_j) - \sum_{j=1}^{N_S} -2\gamma\beta_{ti}\alpha_{tj}K(z_i, x_j)(z_i - x_j) \right) \quad (22)$$

$$\frac{\partial \rho_{Poly}}{\partial z_i} = \sum_{t=1}^T \left(\sum_{j=1}^{N_Z} p\beta_{ti}\beta_{tj}(z_i \cdot z_j)^{p-1}z_j - \sum_{j=1}^{N_S} p\beta_{ti}\alpha_{tj}(z_i \cdot x_j)^{p-1}x_j \right) \quad (23)$$

The optimal coefficients of RVs are then recalculated by solving the following equations for all T binary-class SVMs [6]

$$\boldsymbol{\beta}_t = (\mathbf{K}^{\text{zz}})^{-1} \mathbf{K}^{\text{zx}} \boldsymbol{\alpha}_t, t = 1, \dots, T \quad (24)$$

where $K_{ij}^{zz} = K(z_i, z_j)$, $K_{ik}^{zx} = K(z_i, x_k)$, $i, j = 1, \dots, N_Z$, $k = 1, \dots, N_S$, $\boldsymbol{\beta}_t = (\beta_1, \beta_2, \dots, \beta_{N_Z})$, $\boldsymbol{\alpha}_t = (\alpha_1, \alpha_2, \dots, \alpha_{N_S})$.

As mentioned in [6], solving (24) always yields optimal coefficients given a fixed reduced set. In our experiments, this (24) is performed twice: after all RVs are constructed and after optimizing ρ in (21).

3.3 Simplification Algorithm

The simplification algorithm iteratively selects two SVs x_i and x_j and replaces them with a newly constructed vector z . For selecting a good pair of SVs, we use the first selection heuristic based on Euclidian distance between two vectors in the input space [5]. In a multi-class application, one SV may be a positive SV (having positive weight) in one SVM and negative in another, the selection of a pair candidate (x_i, x_j) becomes:

$$x_j = \arg \max_{k=1, \dots, N_S, k \neq i, \alpha_{ti}\alpha_{tk} \geq 0, t=1, \dots, T} \|x_i - x_k\| \quad (25)$$

The condition $\alpha_{ti}\alpha_{tj} \geq 0, t = 1, \dots, T$ means that two SVs, x_i and x_j , are constrained to belonging to the same positive or negative class in every binary-class SVMs, or one of them is not involved in that SVM. As we found that the construction of RVs is simple, then we propose to go further by using pre-calculating reduced vectors for all pair candidates, and the final selection is based on the difference between the combined vector z and the sum of two vector x_i , and x_j in feature space (7).

The simplification process will stop when the size of reduced set equal or less than a predefined number which is given as a parameter for the algorithm. This predefined number determines speed-up rates of simplified SVM in testing phase: a smaller size of the reduced set results faster SVM. However, as shown by experimental results that

with the same speed-up rate predictive accuracy of simplified SVM may degrade depending on application.

4 Experiment

We selected different multi-class datasets publicly available [9] for evaluating our proposed method. Summarization of these datasets is described in Table 1. Original SVM classifiers are firstly trained by LibSVM [7], a well known SVM training implementation, and then simplified by the algorithm described in Section 3. Training parameters were selected so that the trained (original) classifiers have good predictive accuracy on independent test data. For the reproducibility of experiments we also report parameter setting in Table 1.

In the first experiment, we run the proposed algorithm with different values of N_z , indicating different speed-up rates of simplified SVM. Then we compare accuracy of original SVMs and reduced SVMs on the prepared test data. Experimental results reported in Table 2 show that simplified SVMs with only 10% number of SVs have almost identical performance with original SVMs. Specially, on the “dna” and “shuttle” datasets, speed-up rate can reach up to 100 times without any loss in predictive accuracy.

Table 1. Dataset and parameter setting used in experiment

Name	Dimension	# Class	# Training	# Testing	Parameter
dna	180	3	2,000	1,186	$\gamma = 0.01, C = 10$
satimage	36	6	4,435	2,000	$\gamma = 0.1, C = 10$
shuttle	9	7	43,500	14,500	$\gamma = 0.1, C = 10$
usps	256	10	7,291	2,007	$\gamma = 0.0078, C = 10$

Table 2. Predictive accuracy of reduced SVMs with different speed-up rates on the selected datasets. Classifiers with 100% of SV(s) are original SVMs trained on training data. Other SVMs are simplified solutions having different number of RVs. With 10% of SVs, or running 10-time faster, reduced SVMs have almost identical performance with original classifiers.

Data	dna		satimage		shuttle		usps	
	Percentage of SV	# SV	Acc. (%)	# SV	Acc. (%)	# SV	Acc. (%)	# SV
100%	843	95.62	1215	89.75	4191	99.03	1670	94.77
50%	422	95.62	608	89.75	2096	99.03	835	94.77
10%	84	95.53	122	89.45	419	99.03	167	94.67
5%	42	95.19	61	89.25	210	99.03	84	93.92
1%	8	95.03	12	78.00	42	99.04	45	89.59

In the second experiment, we compare our algorithm and the multi-class SVM simplification method described in [4] on the “usps” handwritten recognition dataset. In our experiment, LibSVM [7] produces 45 one-versus-one SVMs with total number of (shared) SV are 1422 for Gaussian kernel and 1543 for polynomial kernel. The corresponding accuracies are 95.4% and 95.1%. Comparison in Figure 1 shows that the

bottom-up approach for multi-class SVM simplification produces very competitive performance in term of speeding-up rate and preserving predictive accuracy of simplified SVMs. Note that the classifier in [4] consists of ten one-versus-rest SVMs with the total number of SV is 2620 and the test accuracy is 95.6%. We took the result (accuracy) of [4] for comparison and did not re-conduct its experiments because reproduction of experimental results in [4] is difficult to achieve due to: i) it was conducted on a preprocessed data (smoothing, random sampling); ii) the nature of local minima problem.

Fig. 1. Comparison of reduction rate and loss in predictive accuracy between simplified SVMs produced by the method described in [4] and our propose algorithm (for original Gaussian and Polynomial SVMs) on the handwritten “usps” dataset. With the same SV reduction rate the bottom-up method produces simplified SVMs with very competitive accuracy.

5 Conclusion

We have presented our extension of the bottom-up approach to multi-class SVM simplification. We showed that the proposed method possesses several key advantages over other reduced set methods: applicable for multi-class SVMs, theoretically stable and efficient, competitive performance in terms of speed-up rate and predictive accuracy of reduced SVMs. One remaining problem is the calculation of RV construction is kernel-dependent. In this paper we have introduced formulae for constructing RVs for two most widely used Gaussian and polynomial kernels. With other types of kernel it apparently requires other formulae.

References

1. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory. COLT 1992, pp. 144–152. ACM, New York (1992)

2. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2(2), 955–974 (1998)
3. Burges, C.J.C.: Simplified support vector decision rules. In: Proc. 13th International Conference on Machine Learning, San Mateo, CA, pp. 71–77 (1996)
4. Tang, B., Mazzoni, D.: Multiclass reduced-set support vector machines. In: Proceedings of the 23rd international Conference on Machine Learning, ICML 2006, Pittsburgh, Pennsylvania, June 25–29, 2006, vol. 148, pp. 921–928. ACM, New York (2006)
5. Nguyen, D., Ho, T.: An efficient method for simplifying support vector machines. In: Proceedings of the 22nd international Conference on Machine Learning, ICML 2005, Bonn, Germany, August 07–11, 2005, vol. 119, pp. 617–624. ACM, New York (2005)
6. Schölkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Müller, K.-R., Rätsch, G., Smola, A.: Input Space vs. Feature Space in Kernel-Based Methods. *IEEE Trans. Neural Networks* 10, 1000–1017 (1999)
7. Fan, R.-E., Chen, P.-H., Lin, C.-J.: Working set selection using second order information for training SVM. *Journal of Machine Learning Research* 6, 1889–1918 (2005), <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
8. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical recipes in C++: the art of scientific computing. Cambridge University Press, Cambridge (2002)
9. <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html>

A Syntactic-based Word Re-ordering for English-Vietnamese Statistical Machine Translation System

Hong-Nhung Nguyen Thi and Dien Dinh

Faculty of Information Technology

University of Science - National University of Ho Chi Minh City
227 Nguyen Van Cu St, District 5, Ho Chi Minh City, Vietnam

nthnhung_ddien@fit.hcmuns.edu.vn

<http://www.fit.hcmuns.edu.vn/>

Abstract. In machine translation, the re-ordering of word from source to target language is one of the major steps that affect mainly the performance of the system. Among many approaches for this type of problem, syntactic is an effective method for handling word-order in a statistical machine translation (SMT) system. In this paper, we introduce a word re-ordering approach that makes use the syntactic rules extracted from parse tree for the English-Vietnamese SMT system. Our word re-ordering rule set includes rules in noun phrase, verb phrase and adjective phrase. According to the experiment result, the noun phrase rules are the most significant rules of all. Compared with the MOSES phrase-based SMT system [1], these rules can improve BLEU score of 3.24 on our testing corpus. Moreover, we also conduct other experiments by using different combinations of rules to study their effectiveness. And we find that the translation performance for each corpus can be tuned by different ways of combination.

Keywords: Statistical machine translation, word re-ordering, parse tree, syntactic-based word re-ordering rule.

1 Introduction

The machine translation task basically consists of two sub tasks: predicting the translation of word, and deciding the order of predicted words in target language. For some language pairs such as English-Chinese, English-Japanese, and English-Vietnamese, the re-ordering problem is especially hard, because the target word order differs significantly from the source word order.

Statistical machine translation (SMT) system also solves the word order problem in the decoding phase. It simply used distance-based re-ordering model. In MOSES toolkit, they use the lexicalized re-ordering model and integrate in decoding phase effectively. These re-ordering methods are only based on pure statistic information. Moreover, for a different language pairs, syntactic-based re-ordering is considered as an effective and comparative method for handling

word order in SMT context in literature [2], [3] and [4]. In this approach, the source language sentences are first parsed into a parse tree. A set of syntactic rules is then applied to the tree in order to transform the sentence of source language into a word order that is much closer to which of the target sentence. The advantages of using parse tree are (i) the constituents move as a whole and keep the phrasal cohesion constraints and (ii) we can model broad syntactic re-ordering phenomena. For example: the most significant difference between English and Vietnamese is the order of noun and adjective. From this difference we create a re-ordering rule: **if** a noun appears after an adjective **then** re-order noun and adjective. As a result, the phrase like **the/DT first/JJ characters/NNS** will be transform into **the/DT characters/NNS first/JJ** in Vietnamese.

Theoretically, by using parse tree as syntactic information, we can generalize our rule **if** a noun or noun phrase appears after an adjective or adjective phrase **then** their order will be swapped. When applying this rule to a complex case, we can transform the whole adjective phrase with noun without mentioning what adjective phrase contains. In case word re-ordering appears in the adjective phrase, our rule is still applicable. Fig. 1 exemplifies the above cases.

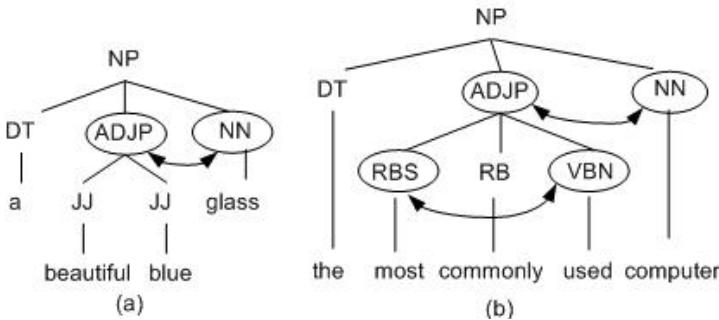


Fig. 1. Re-ordering noun and adjective phrase without mentioning what adjective phrase contains. (a) No re-ordering in adjective phrase. (b) Re-ordering in adjective phrase.

Recently some syntactic re-ordering methods are carried out on syntactic parse trees. Collins et al. [5] describe a method to re-order German sentence in German-English translation, where six transformations are applied to the surface string of parsed source sentence. Quang et al. [4] present an syntactic re-ordering approach for Chinese-English translation. They used three categories of re-ordering rules including verb phrase, noun phrase and localize. Nguyen and Shimazu [6] develop a phrase-based system better dealing with re-ordering for English-Vietnamese translation. The re-ordering is done by using morpho-syntactic analysis and transformation rules. The set of rules are automatically extracted from corpus. Since the syntactic transformation task is ambiguity, they have to use a lexicalized probabilistic context free grammar (LPCFG). Our approach is a little similar to this paper, but we focused on three kinds of phrases: noun phrase (NP), verb phrase (VP) and adjective phrase (ADJP); and

we didn't use probabilistic to simplify our work. Additionally we analysis deeply the impact of each category on the performance of translation.

The purpose of word re-ordering is transforming the source language sentences into a word order that is closer to that of the target language. In fact, this action will decrease the number of cross align, on the other hand flatten the word-align between source and target language. Fig. 2 illustrates this purpose. It can be seen that before word re-ordering all alignments are cross alignments, and after re-ordering, all alignments are flattened.

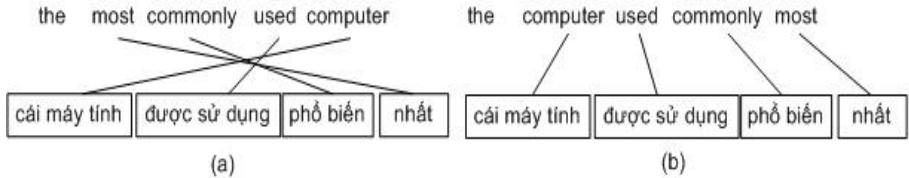


Fig. 2. (a)Before re-ordering: There have many cross-alignments. (b)After re-ordering: All of them are flattened.

The fewer cross alignments are, the better translation SMT performs. So, for evaluating more exactly about the effectiveness of rule sets, besides the BLEU score [16] we also investigate the effectiveness of our rule set in the term of cross alignments count.

The rest of this paper is organized as followed. Section 2 presents our system overview. In section 3 we introduce the details of re-ordering rules, how we derive and present them. Section 4 shows our experiments, evaluations and our analysis about the impact of different types of rules. Finally we conclude our approach and discuss the future work in section 5.

2 Background

In this section we describe the phrase-based SMT system which we use as baseline in our experiments. Then we review some approaches in SMT re-ordering and our approach.

2.1 Phrase-Based SMT System

SMT is now more and more common in machine translation. The basic idea of this approach is using probability of translation a sentence from source language sentence into target language. SMT searches the most probable English sentence given a foreign language sentence $P(e|f)$:

$$\arg \max_e P(e|f) = \arg \max_e P(e) * P(f|e)$$

From above equation, it can be seen that SMT includes two main models:

- *Language model - $P(e)$* : assign a higher probability to fluent/grammatical sentences. This probability is estimated by using monolingual corpus.
- *Translation model - $P(f|e)$* : assign a higher probability to sentences that have corresponding meaning by using bilingual corpus.

Based on the unit to count the probability in translation model, SMT has three main approaches, including word-based, phrase-based and syntax-based. Among them the phrase-based is currently becomes the state of the art approach in machine translation.

In this approach [7], e is segmented into phrases e_1, e_2, \dots, e_i (assuming a uniform probability distribution over all possible segmentations). e is re-ordered according to some distortion model. The translation model is the probability that a phrase e transfer to phrase f . Though other phrase-based approaches use different models, the basic architecture of phrase segmentation, phrase re-ordering, and phrase translation is the same.

The standard SMT system currently use distance-based re-ordering model but its performance is not so good. This has encouraged researchers to propose many re-ordering methods to improve the translation quality.

2.2 Re-ordering Approaches

There are two main possible directions when facing the re-ordering problem for SMT: output or target language sentence re-ordering and input or source language sentence re-ordering.

Output Sentence Re-ordering. Recently, Kumar and Byrne [8] use this approach by involving learning weighted finite state transducers that account for local re-ordering of two or three positions, allowing each word to jump a maximum of one or two positions. And they get encouraged results from the experiments on Arabic-English and Chinese-English.

Input Sentence Re-ordering. The main idea of this approach is to avoid non-monotonous translation problem by re-ordering the input sentence. This helps the translation model not need to account for possible word re-ordering; on the other hand, to help the decoding can directly translate word-by-word or phrase-by-phrase. The main advantage of this method is that there is more information for re-ordering methods when the source language sentences are always given. This is the reason why there are many works related to source sentence re-ordering up to now. Sanchis and Casacuberta [9] consider the re-ordered input sentence as a new language and create a model like translation model in SMT. They then generate n-best list re-ordering; translate them all and finally select an output sentence which obtained the highest probability of the translation model. Zhang et al. [10] do re-ordering at chunk-level by using automatically learned rules. Their result is reported for Chinese-English task, it gains about 0.5 - 1.8 BLEU score. [2], [3], [6] and [15] also use this approach and claim that their model improves the BLEU score for each SMT system in their experiments.

We choose the second one in our proposed model; it is the re-ordering from the source language side. We apply syntactic rules to each input parse tree and then send it to baseline SMT system.

3 Syntactic Word-Reordering Rules

We use the Penn Tree Bank's POS and phrase tag set for a suitable set of re-ordering rules. In this paper, we focused on three categories: noun phrase (NP), verb phrase (VP) and adjective phrase (ADJP) that we considered to be the most prominent candidates for re-ordering.

For creating rules, we exploit systematic differences between English and Vietnamese word order [11]. A re-ordering rule consists of a left-hand-side (LHS) and a right-hand-side (RHS). The LHS is a syntactic rule, it composes of POS tag, syntactic tag of phrases and words. The RHS is the re-ordering sequence, we use zero-based index in this side. Table 1 contains some rule examples. Rule number 5 and 6 are examples of lexical rule. Rule 5 means that if NP has DT NP and the word at DT position is *this*, then re-order *this* and NP. We need to use this kind of rule because there are many exceptions in English and Vietnamese word order. In our proposed rules we have just carried out some exceptions.

Table 1. Sample of our rule set

No.	Category	LHS	RHS
1.	NP	ADJP NP	1 0
2.	NP	DT NP POS	2 1; 1 0
3.	ADJP	NP VP	1 0
4.	VP	MD RB VP	1 0
5.	NP	DT/ <i>this</i> NP	1 0
6.	ADJP	RB/ <i>much</i> JJR	1 0
7.	NP	NNS	0 0
8.	ADJP	JJ	0 0
9.	ADJP	JJ ADJP	0 0 ; 1 1
10.	VP	VBN	0 0

Rules from number 7 to 10 are pseudo-rules. Pseudo-rules are monotone rules; it means if any phrases satisfy these rules, their word order will not be changed. This form of rules create a hierarchical structure in our rule sets. This structure helps NP rule take into account ADJP rules, or ADJP rules take into account VP rules, etc. Beside transformation in a phrase, we also transfer between two phrases if they satisfy the rule. And with a small number of rules we can automatically check many word re-ordering cases. For example, with a noun phrase like **NP (JJ JJ NNS)**; although we can't see any rules that the LHS is the same with its NP, our system can check and transform some terms in NP to make its order closer to Vietnamese order. Let go from right to left, first replace NNS by NP (rule 7), then JJ by ADJP (rule 8), then JJ ADJP by ADJP (rule

9), finally we have **NP (ADJP NP)**. This satisfies rule 1 then transform ADJP and NP, notice that **ADJP** includes (**JJ JJ**).

After extracting our set of rules, we made a series of experiments to analysis the impact of each category of rules. We will present more detail about this in the following section.

4 Evaluation

4.1 Corpus Characteristics

The corpus we used for training and testing our system is the English-Vietnamese parallel corpus belongs to the Vietnamese Computation Linguistic group [12]. We only use two subsets that are Cadasa corpus (*C*) and IBM corpus (*I*). Both of them are in computing.

Table 2. Our corpus characteristics

Corpus	Sent. Pairs	Average Length		Number of Tokens	
		English	Vietnamese	English	Vietnamese
<i>C</i>	8963	18.97	22.44	8388	5808
<i>I</i>	4997	16.41	15.56	2214	2464

Table 2 presents the characteristic of two corpora. We randomly divide the corpus to 10 parts, got 9 parts for training, 0.5 parts for developing and other 0.5 part for testing.

4.2 Experiments

Translation model is created by using GIZA++ and MOSES toolkit, while language model is generated by SRILM toolkit. For Vietnamese preprocessing, we apply the word segmentation tool in [13]. On the English side, we parse our experiment data by Stanford parser [14] in advance of applying re-ordering rules. The rules described in the previous section are then applied to the parse tree of each input. After re-ordering phase, the output will be sent to SMT for training, tuning and testing.

We evaluate the performance of our system by showing the improved performance for the machine translation system in a well-known metrics - BLEU score. Our experiments compare the translation quality of three below approaches:

- The standard SMT system or the baseline system used distance-based re-ordering.
- The MOSES toolkit used the lexicalized re-ordering model.
- Our system made use of syntactic re-ordering rules.

All three systems are trained by MOSES phrase-based toolkit [1] but with different configurations.

Table 3. Translation result of three approaches in BLEU score

BLEU-4	Baseline	MOSES' model	Our re-order	MOSES' gain	Our gain
<i>C</i>	50.09	51.47	52.62	+1.38	+2.53
<i>I</i>	51.57	58.91	60.75	+1.4	+3.24

As shown in table 3, our approach outperforms the other ones. It improve the BLEU-4 score by 2.53 on *C* corpus, 3.24 on *I* corpus compared with the baseline system. These are the best result when we in turn applied NP; NP and VP; NP and ADJP; and all rules to our corpora. The next section will describe the impact of each type of rules in more detail.

4.3 Frequency of Rules

We collect statistics to evaluate how often the re-ordering rules are applied to the corpus, remember that we only counted on real rules, not on pseudo-rules. The most frequent rule is NP rules, which count over 97% in both *C* and *I* corpus. However, the role of each NP rule is different. With *C* corpus the most frequent rule is NP (ADJP NP) while in *I* corpus it is NP (NN NP).

The VP and ADJP frequency in *C* differ in *I*. In *C* corpus, VP and ADJP frequency are a bit equal (1.03% and 1.64%); but in *I* corpus VP phrases overweight ADJP phrases (2.51% and 0.15%). This can be explained by the content of each corpus. Although they are both computing, their sub fields are not the same. The *C* corpus is got from a computing book while *I* corpus is the IBM's manual. And the manual including more verb phrases than a book is understandable. And this small difference will affect significantly the translation quality.

4.4 Impact of Individual Re-ordering Rules

In order to assess the relative effectiveness of each type of re-ordering rules, we conduct an experiment in which we trained and evaluated systems using data that were re-ordered using different subsets of the rules. Because of the small number of VP and ADJP frequency, instead of applying VP and ADJP individually, we combined them with NP.

Table 4 shows that the NP rules are the most effective rules, when applying them the result is improve significant. However, when we add VP rules or ADJP rules, it doesn't consequently improve scores, even in *C* corpus, it creates lower scores. Refer to table 4 and you will find the differences between *C* and *I* corpus. With *C* the rules give the best result include NP and ADJP rules, while in *I*, NP combining with ADJP is the best. In both corpora, combining NP, VP and ADJP led a worse result; especially with *C*, it decreases below the baseline's score.

The explanations in previous section gave us some ideas for this result. Because *I* corpus contains more VP than *C* corpus, when we apply VP re-ordering rules it makes the translation quality increase. We also make some error analysis and find that beside the parse error, the rule errors take an important role

Table 4. Translation result in BLEU score when applying each kinds of rules

BLEU-4	Baseline	Moses	NP	NP+VP	NP+ADJP	All
<i>C</i>	50.09	51.57	52.27	48.34	52.62	48.35
<i>I</i>	57.51	58.91	60.16	60.75	60.41	60.1

in decreasing the performance. Although VP includes two rules, it has created most of errors. Because the VP is very complex, we can only suggest two rules. And they are so general that they make so much noise. For example, with rule VP (MD AVDP VP), it is applied 214 times in *C* training corpus, but only 40 correct times (about 18.7% correct). The other is 81.7% errors, a large scale of error, which decreases our system below the baseline. The same things happen with ADJP for *I* corpus, however when combining all categories, it doesn't make the performance below the baseline as *C* corpus. So depending on each corpus, we should have a suitable combination to improve the translation performance.

4.5 Decreasing Cross Alignment

The purpose of word re-ordering is transforming the source language sentences into a word order that is closer to that of the target language. In fact, this will decrease the number of cross alignments, on the other hand flatten the word-align between source and target language.

Table 5. Cross alignments of each approach

Cross Align	Baseline	Moses	NP	NP+VP	NP+ADJP	All
<i>C</i>	135,660	135,660	101,749	103,505	100,506	103,566
<i>I</i>	28,524	28,524	15,042	16,361	15,409	16,149

To measure our approach's effectiveness, we calculate the number of cross alignments after applying rules to training corpus. The result is reported in table 5.

The MOSES' model makes use of the lexicalized re-ordering model integrated in decoding phase; so its cross alignments are equal with baseline. The other results also reflect all above analysis. Because the NP is the most effective rules, it makes the cross alignment decrease significantly. From this table we can also say that compared with NP, NP + ADJP and All approaches, NP+VP creates the largest number of cross alignments in *C* corpus. Following is an example illustrating this kind of noise. We have an English sentence: *you can also carry your mp3 recordings*. In Fig. 3, after applying VP rule: VP (MD ADVP VP), number of cross alignments increase from 3 to 5.

This noise makes NP+VP perform worst, and the score when we combine all kind of rules lower than the NP. In evident, the translation performance is opposite with the number of cross alignments. Therefore, we conclude that the fewer cross alignments are, the better our translation system performs.

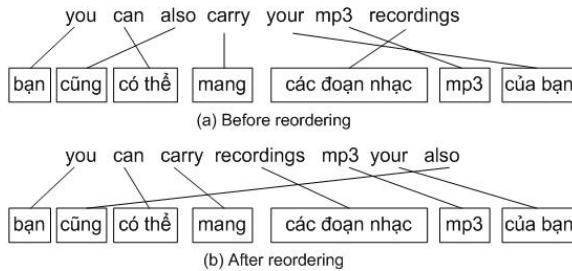


Fig. 3. (a) Before re-ordering, number of cross alignment is 3. (b) After re-ordering, number of cross alignment is 5.

5 Conclusion

In this paper, we make use a set of syntactic word re-ordering rules to transform English sentences into a much closer to Vietnamese form in terms of their word re-order. We evaluate the re-ordering approach within the MOSES phrase-based SMT system. Our re-ordering approach improve the BLEU score from 50.09 to 52.62 with *C* corpus, from 57.51 to 60.75 with *I* corpus. In three kinds of rule we exploit, NP rules play the most important roles. It is above 97% of all kinds and it makes our system's quality increase significantly. We also analyze the impact of each rule when combining in turn NP with ADJP, NP with VP and all of them. Our experimental result shows that each corpus suits with different combination when tuning its performance. We also claim that the fewer cross alignments are, the better our system performs.

Our proposed rule set in this paper can be broadened to cover more re-ordering cases in English and Vietnamese. For a better translation system, we can not only introduce more re-ordering rules, but also exploit other kinds of rules such as question, adverbial phrase, prepositional phrase, etc.

References

1. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Moses, E.H.: Open source toolkit for statistical machine translation. In: Proceedings of ACL, Demonstration Session (2007)
2. Xia, F., McCord, M.: Improving a statistical MT system with automatically learned rewrite patterns. In: Proceedings of COLING (2004)
3. Quang, P.-C., Tuoutanova, K.: A Discriminative syntactic word order model for machine translation. In: Proceedings of ACL 45th, pp. 9–16 (2007)
4. Wang, C., Collins, M., Koehn, P.: Chinese syntactic re-ordering for statistical machine translation. In: Proceedings of 2007 Joint Conference on Empirical Methods in NLP and CL NLP, pp. 737–745 (2007)
5. Collins, M., Koehn, P., Kucerova, I.: Clause restructuring for statistical machine translation. In: Proceedings of the 43rd Annual Meeting of the Assoc. for Computational Linguistics (ACL), Ann Arbor, Michigan, pp. 531–540 (2005)

6. Nguyen, T.P., Shimazu, A.: A syntactic transformation model for statistical machine translation. In: Matsumoto, Y., Sproat, R.W., Wong, K.-F., Zhang, M. (eds.) *ICCPOL 2006. LNCS (LNAI) vol. 4285*, pp. 63–74. Springer, Heidelberg (2006)
7. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proc. of the HLT-NAACL 2003 conference, Edmonton, Alberta, Canada, pp. 127–133 (2003)
8. Kumar, S., Byrne, W.: Local phrase re-ordering models for statistical machine translation. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in NLP, pp. 161–168 (2007)
9. Sanchis, G., Casacuberta, F.: N-best re-ordering in statistical machine translation. *Jornadas en Techlogia del Habla*, pp. 99–104 (2006)
10. Zhang, Y., Zens, R., Ney, H.: Chunk-level re-ordering of source language with automatically learned rules for statistical machine translation. In: Proceedings of SSST, NAACL-HLT, pp. 1–8 (2007)
11. Dien, D.: Comparision word order of attributions in English and Vietnamese. In *Journal of Social Sciences and Humanities. University of Social Sciences and Humanities. HCM City* (2001)
12. Dinh, D.: Building an Annotated English-Vietnamese parallel Corpus. In *MKS: A Journal of Southeast Asian Linguistics and Languages* 35, 21–36 (2005)
13. Dien, D., Thuy, V.: A maximum entropy approach for Vietnamese word segmentation. In: *Proceedings of 4th IEEE International Conference RIVF 2006*, Ho Chi Minh City, Vietnam, February 12-16, 2006, pp. 247–252 (2006)
14. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: *Proceedings of ACL 2003* (2003)
15. Li, C.-H., Zhang, D., Li, M., Zhou, M., Li, M., Guan, Y.: A probabilistic approach to syntax-based re-ordering for statistical machine translation. In: *Proceedings of 45th ACL*, pp. 720–727 (2007)
16. Papineni, K.A., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: *The Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318 (2002)

A Multi-modal Particle Filter Based Motorcycle Tracking System

Phi-Vu Nguyen¹ and Hoai-Bac Le²

¹ Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
npuv@fit.hcmuns.edu.vn

² Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
lhbac@fit.hcmuns.edu.vn

Abstract. Object tracking in computer vision is an attractive research field due to its widespread application area and challenges. In the recent years, Particle filter is known as a prominent solution for the state estimation problems in point tracking and successfully applied in a wide range of applications. But one of its limitations is the weakness at constantly maintaining the multi-modal target distribution that may arise due to occlusion, clutter or the presence of multiple objects. Lately, that weak point has been overcome in a multi-modal Particle filter (MPF). This paper aims to build some most basic functions of a motorcycle surveillance system using MPF and basing on the color observation model. Accompanied with a simple but effective detecting strategy, the application has the processing rate equivalent to a real time tracking system and high performance.

Keywords: Particle filter, Sequential Monte Carlo, multi-modal Particle filter, MPF, motorcycle tracking.

1 Introduction

Object tracking is becoming a more and more attractive research field because of not only its widespread applications in both military (missile defense, air traffic control, ocean surveillance, ...) and civilian (public or secret surveillance) areas but also its challenging problems. In computer vision, object tracking takes the crucial role since it is an essential function of motion understanding.

In [14], object tracking in computer vision is divided into three main categories. Point tracking, which regards an object as a point and focuses on its position and motion, is among them. Filtering is a class of methods that is suited for solving the dynamic state estimation problems in point tracking. In literature, Kalman filter is known as an optimal approach for recursive Bayesian estimation in case of linear and/or Gaussian dynamic and observation models. Unfortunately, in practice, these assumptions are rarely satisfied, and this caused the birth of many non-linear filtering methods such as: extended Kalman filter, unscented Kalman filter, approximate grid-based methods, Gaussian sum filters ([12]). In the recent years, Particle filter has appeared as a strong non-linear filter in a wide range of tracking applications. As other non-linear filters, Particle filter also gets the recursive Bayesian estimation to be its conceptual solution, but it uses the Monte Carlo idea to approximate this framework. Specifically, Particle filter uses a large set of samples - each sample is considered as a particle – drawn from

a proposal distribution to represent the posterior distribution, then these samples with their weights enclosed are aggregated to produce the current state estimation of targets. The strong points which make Particle filter become a prominent method for non-linear filtering problems are: the requirement of few system assumptions, the convergence rate does not depend on the number of state vector dimensions, its simplicity and generality in implementation; furthermore, its accuracy can be increased by enlarging the sample set. Those are the reasons which made Particle filter be successfully applied in many contexts: single object (car, face) tracking using color-based measurements [9], multi-target tracking in passive sonar context [5], vehicle tracking by radar signal [6], [7], hockey player tracking [10], traffic estimation [8], ...

One of the limitations of Particle filter is its weakness at constantly maintaining the multi-modal target distribution that may arise due to occlusion, clutter or the presence of multiple objects. Recently, Vermaak et al. [13] proposed a mixture particle representation to overcome this shortcoming. In that work, the authors put forward a mixture model for the joint posterior distribution, each mixture component has its own posterior distribution and interacts only via its mixture weight, then gave the particle approximation to the joint model.

Motorcycle tracking in particular and traffic tracking in general, is an interesting but challenging application. Its main difficulties can be enumerated as: the severe occlusions when traffic density is high (especially in rush hours), the shadows of big vehicles, and the real time processing requirement. This paper aims to build some most basic functions of a motorcycle surveillance system using multi-modal Particle filter (MPF) in [13] and basing on the color observation model. This is also an improvement of [4] in detecting stage, tracking algorithm and surveillance facilities. The system works well with the number of targets less than 10 per frame and high processing rate (20-30 frames/s), which is suited for a real time application.

The remains of this paper is outlined as follows, section 2 gives the brief theoretical background of Particle filter, section 3 summarizes the idea of multi-modal Particle filter in [13], section 4 presents the applied MPF in motorcycle tracking, the experimental results are showed in section 5 and the Conclusions is at the final section.

2 Particle Filter

Given a dynamic system that satisfies all the assumptions of first-order Markov model. At time k , let \mathbf{x}_k be the system state, \mathbf{z}_k be the observation, \mathbf{X}_k be the sequence of system states from initiation, \mathbf{Z}_k be the sequence of all available observations. In order to analyze and make inferences about a dynamic system, at least two models are required: the dynamic model, which describes the evolution of state with time and the observation model, which describes the relationship between the observation and the system state at the same time.

The Bayesian solution states that the posterior density could be attained through two steps:

Prediction:

$$p(\mathbf{x}_k | \mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{Z}_{k-1}) d\mathbf{x}_{k-1} \quad (1)$$

Update:

$$p(\mathbf{x}_k | \mathbf{Z}_k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Z}_{k-1})}{\int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Z}_{k-1}) d\mathbf{x}_k} \quad (2)$$

However, except the very restrictive cases as Kalman filter assumptions, this is just a conceptual solution because there is no analytical way to evaluate the integrations in (1) and (2). For the wide range of the other cases, non-linear filters are employed to approximate this solution. With the same purpose, Particle filter represents the posterior density by a large set of samples drawn from a proposal density function: $\mathbf{x}_k^i \sim q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$. Each sample is enclosed with a weight which is recursively updated as follows:

$$\tilde{w}_k^i = \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)} \times w_{k-1}^i \quad (3)$$

where w_{k-1}^i is a normalized weight at time $k-1$: $w_{k-1}^i = \frac{\tilde{w}_{k-1}^i}{\sum_{j=1}^N \tilde{w}_{k-1}^j}$ and \tilde{w}_k^i is an unnormalized weight at time k . Hence, the posterior density is approximated as:

$$p(\mathbf{x}_k | \mathbf{Z}_k) \approx \sum_{i=1}^N w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (4)$$

where $\delta(\cdot)$ is the Delta Dirac function.

The foregoing forms the main content of the Sequential Importance Sampling (SIS) method, which is the simplest version of Particle filter. However, there is a puzzle of SIS arising after a definite number of iterations: all but one particle will have negligible normalized weights, this is namely the Degeneracy problem ([2], [3]). To solve this problem, we need to adapt to SIS the Resampling step once the degeneracy phenomenon is found out - by keeping track of the Effective Sample Size

$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^N (w_k^i)^2}$. The tasks of this step is to eliminate samples with low importance

weights and multiply samples with high importance weights. After the Resampling step, all of the importance weights are uniformed. The Particle Filter can be formalized as: $[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^N] = \text{PF}[\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^N, \mathbf{z}_k]$ ([12]).

3 Multi-modal Particle Filter

3.1 Bayesian Framework and Particle Approximation

With the consideration as an M -component mixture, our system's posterior density is modeled as a composition of M componential posterior densities:

$$p(\mathbf{x}_k | \mathbf{Z}_k) = \sum_{m=1}^M \pi_{m,k} p_m(\mathbf{x}_k | \mathbf{Z}_k) \quad (5)$$

where $\pi_{m,k}$ is the mixture weight of each component at time k and $\sum_{m=1}^M \pi_{m,k} = 1$.

Assume that $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$ is known, under some rigorous transformations [13], we can get $p(\mathbf{x}_k|\mathbf{Z}_k)$ in (5) with:

$$p_m(\mathbf{x}_k | \mathbf{Z}_k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p_m(\mathbf{x}_k | \mathbf{Z}_{k-1})}{\int p(\mathbf{z}_k | \mathbf{x}_k) p_m(\mathbf{x}_k | \mathbf{Z}_{k-1}) d\mathbf{x}_k} \quad (6)$$

and

$$\pi_{m,k} = \frac{\pi_{m,k-1} \int p(\mathbf{z}_k | \mathbf{x}_k) p_m(\mathbf{x}_k | \mathbf{Z}_{k-1}) d\mathbf{x}_k}{\sum_{n=1}^M \pi_{n,k-1} \int p(\mathbf{z}_k | \mathbf{x}_k) p_n(\mathbf{x}_k | \mathbf{Z}_{k-1}) d\mathbf{x}_k} \quad (7)$$

where the prior density $p_m(\mathbf{x}_k | \mathbf{Z}_{k-1})$ of each component is achieved by (1).

After having Bayesian framework for the mixture representation, Particle filters are used to approximate this model.

Let us denote $P_k = \{N, M_k, \Pi_k, X_k, W_k, C_k\}$ the Particle representation of the joint posterior density in (5), where N is the number of particles, and the parameters at time k : M_k is the number of mixture components, $\Pi_k = \{\pi_{m,k}\}_{m=1}^{M_k}$ is the set of mixture weights, $X_k = \{\mathbf{x}_k^i\}_{i=1}^N$ is the particle set, $W_k = \{w_k^i\}_{i=1}^N$ is the weight set corresponding to particles, $C_k = \{c_k^i\}_{i=1}^N$ is the particle index set, which means: $c_k^i = m$ if the i^{th} particle belongs to the m^{th} mixture component.

Using the Particle representation, the joint posterior density can be approximated as:

$$p(\mathbf{x}_k | \mathbf{Z}_k) \approx \sum_{m=1}^{M_k} \pi_{m,k} \sum_{i \in I_m} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (8)$$

where I_m is the set of indexes of particles that belongs to component m and $\sum_{i \in I_m} w_k^i = 1$. Hence, in this approximation, each mixture component will evolve on its own particle set and just interact via the mixture weights. Therefore, each component acts exactly the same of a Particle filter with SIS algorithm and resampling step. The mixture weights, then, are updated over time as follows:

$$\pi_{m,k} \approx \frac{\pi_{m,k-1} \sum_{i \in I_m} \tilde{w}_k^i}{\sum_{n=1}^{M_k} \pi_{n,k-1} \sum_{i \in I_n} \tilde{w}_k^i} \quad (9)$$

where \tilde{w}_k^i is the un-normalized weight of each particle.

3.2 Multi-modal Maintaining

Due the number of particles for the joint model is fixed whereas the number of mixture components can be changed over time, we need a procedure to re-assign the

particles for each current component, that means to share particles for new occurring objects and revoke those from disappearing objects.

Let: $F: (N, M_k, \Pi_k, X_k, W_k, C_k) \mapsto (N, M'_k, \Pi'_k, X_k, W'_k, C'_k)$ (or $F: P_k \rightarrow P'_k$) be such a re-clustering procedure, where C'_k determines the new particle assignments; Π' is the new mixture weight set and W'_k is the new particle weight set, which are recomputed as follows:

$$\pi'_{m,k} = \sum_{i \in I'_m} \pi_{c_k^i, k} w_k^i, \quad (w')_k^i = \frac{\pi_{c_k^i, k} w_k^i}{\pi_{(c')_k^i, k}} \quad (10), (11)$$

The new Particle representation P'_k neither changes the joint posterior density nor affects the convergence of Particle filter.

4 Multi-modal Particle Filter in Motorcycle Tracking

4.1 Dynamic Model

The state vector of each mixture component is defined as: $\mathbf{x}_k = (x_k, y_k)$, this implies the (continuous) Descartes coordinate of a motorcycle in the image at time k . Following is the dynamic model described via each element of a state vector:

$$x_k = f(x_{k-1}, v_{k-1}^x) = x_{k-1} + v_{k-1}^x; \quad y_k = f(y_{k-1}, v_{k-1}^y) = y_{k-1} + v_{k-1}^y \quad (12)$$

where the noises v_{k-1}^x and v_{k-1}^y are normally distributed, respectively, $N(0, \delta_x^2)$ and $N(0, \delta_y^2)$.

4.2 Observation Model

This paper adopts the color-based observation model, which has been successfully experimented in many target tracking systems ([4], [10], [11]), here is the brief description of this model.

Bhattachayya distance is employed to calculate the “distance” between a reference color model $\mathbf{K}^* \triangleq \{k^*(n; \mathbf{x}_0)\}_{n=1, \dots, N}$ and a candidate color model $\mathbf{K}(\mathbf{x}_k) \triangleq \{k(n; \mathbf{x}_k)\}_{n=1, \dots, N}$:

$$\xi[\mathbf{K}^*, \mathbf{K}(\mathbf{x}_k)] = \left[1 - \sum_{n=1}^N \sqrt{k^*(n; \mathbf{x}_0) k(n; \mathbf{x}_k)} \right]^{1/2} \quad (13)$$

To increase the accuracy, the reference model and candidate model are divided into two sub-regions, then the likelihood of a candidate model is produced:

$$p(\mathbf{z}_k | \mathbf{x}_k) \propto e^{\sum_{j=1}^2 -\lambda \xi^2 [\mathbf{K}_j^*, \mathbf{K}_j(\mathbf{x}_k)]} \quad (14)$$

4.3 Moving Object Detection

4.3.1 Background Learning

The background learning method will generate a sequence of $n-1$ masks $M_i(x)$ ($i = 2, \dots, n$) corresponding to $n-1$ pairs of adjacent frames, each $M_i(x)$ specifies whether the x pixel in the $i-1^{\text{th}}$ frame is a moving point by subtracting the two adjacent frames I_i and I_{i-1} . Then pixels of the background B will be updated basing on $M_i(x)$ sequence. The main content of the background learning algorithm is shown in Figure 1 (left).

Fig. 1. The background learning algorithm (left) and detecting algorithm (right)

4.3.2 Object Detection

After having the background image, moving points in the current frame are determined by combining background difference and inter-frame difference methods. The first one is to calculate the difference between background and input frame while the second performs the same work on consecutive frames. Then, cooperating with some Heuristic techniques such as: eliminating the image regions having low densities of moving points, separating the objects stuck each other due to shades, adjusting the detecting frames to catch the objects exactly. Accompanied with the “spreading oil” algorithm for finding connective regions, the list of moving objects could be produced. The main content of moving object detecting algorithm is shown in Figure 1 (right).

4.4 Object Tracking

In this section, we propose an algorithm to combine two tools: moving objects detection and multi-modal Particle filter to form a completed object tracking algorithm. This algorithm gets the results of the moving object detection to be the premise of Multi-modal Particle filter. During the tracking process, a motorbike passing through the observed area would be detected once only when it starts entering this area, that means the detecting algorithm just needs to perform in a small region at the beginning of the observed area - called “Beginning area” - to detect new object appearances. Therefore, the detecting algorithm does not need to care objects that left (partly or

entirely) the beginning area. So, the detecting algorithm should be modified a little to be applied in the beginning area, renamed BeginningAreaDetect.

The Tracking algorithm is presented in Figure 2, where P_k was defined in section III and NOList_k is the list of new objects entering the beginning area at time k . Note that in this case $p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i)$ is used as the proposal density function.

Fig. 2. The Tracking algorithm

Beside the tracking ability, this tracking algorithm could recognize the motorbikes driven in wrong way. After passing through the beginning area, wrong-wayed motorbikes disappear from the observed area, this will result the likelihood of their candidate regions in rapidly decreasing ($|p(\mathbf{z}_k \mid \mathbf{x}_k) - p(\mathbf{z}_k \mid \mathbf{x}_{k-1})| > \Delta L$ with $|k - k'| < \Delta k$). Taking advantage of this property, wrong-wayed motorbikes could be recognized.

Furthermore, using the tracking results, the average velocity of a motorcycle passing through the observed area could be computed basing on the length of the road line and the number of frames in which this motorcycle presents. The trajectories of motorbikes in the observed area are also saved to notice the dodging and encroaching cases.

5 Experimental Results

Figure 3 below is some results of moving object detection using the background and inter-frame difference method, the above image of each pair is the observed area and the below one is the detecting result. Almost all of the motorbikes entering the

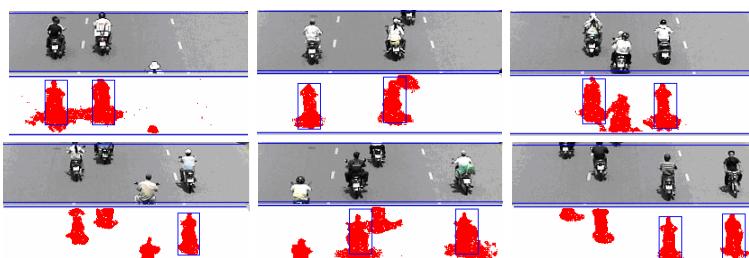
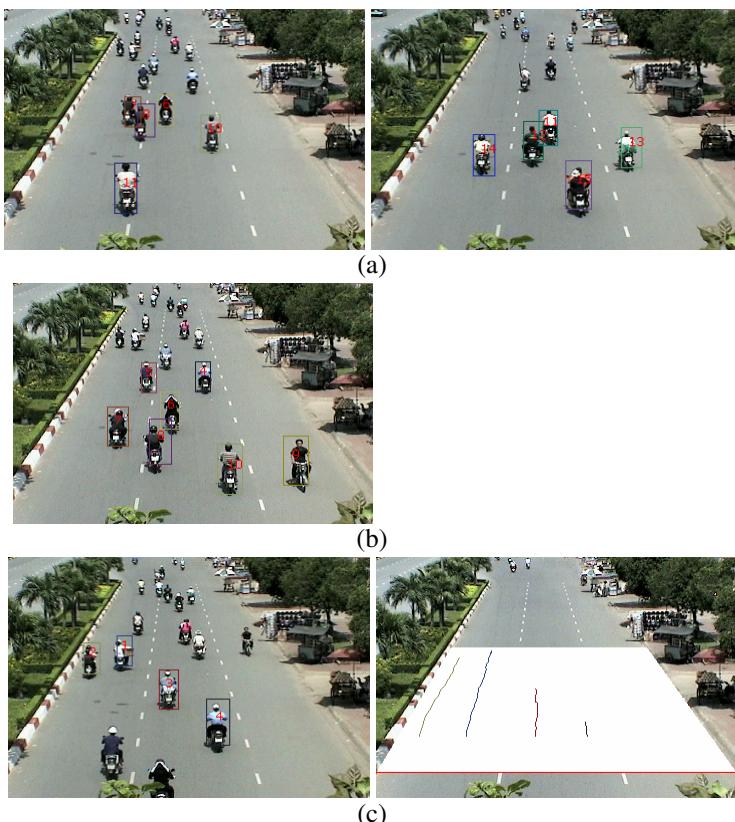


Fig. 3. Some results of detecting moving objects in the beginning area (the above image of each pair is the observation area and the below one is the detecting result)

Table 1. The statistics of detecting results**Fig. 4.** (a) Some tracking results; (b) Wrong-wayed motorcycle detecting and velocity computing; (c) Trajectory saving

beginning area are detected and accurately grasped, which would make a good basis for the tracking stage. However, due to the simplicity of the detecting algorithm, the detecting performance is a little sensitive to illumination change, but the results are still well although there are noises in some cases. Once having good detecting results, the tracking results will become good after. The system works well with the number of targets less than 10 per frame. Some tracking results are shown in figure 4(a), the system could keep the objects well tracked till the end of the observed area. Figure 4(b) is an example of the ability to detect wrong-wayed motorcycles, the object 9 was recorded as driving in wrong way when it traveled through the observed area. The quantitative results are created by testing 11 video clips, each one is about 10-second long, taken in the motorcycle lane in a cloudy weather, the statistics is very satisfactory (Table 1). This system is experimented on a Pentium IV 2.4Ghz, 512 MB RAM, the processing rate is equivalent to 20-30 frames/s, which is suited for a real-time application.

6 Conclusion

This paper is an application of the multi-modal Particle filter to the tracking function of a motorcycle tracking system basing on the color observation model. With the Particle filter background, the system has the strong ability to track objects which is identified with points. Accompanied with the simple but effective detecting strategy, the application has the processing rate equivalent to a real time tracking system and high performance. The authors hope that this work will make a good premise for an automatical traffic surveillance system in big cities of Vietnam in the future.

References

1. Arulampalam, S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. In: IEEE Transactions on Signal Processing, vol. 50(2) (February 2002)
2. Doucet, A., Freitas, N., Gordon, N.: Sequential Monte Carlo Methods in Practice. Springer, Heidelberg (2001)
3. Doucet, A., Godsill, S.J., Andrieu, C.: On sequentialMonte Carlo samplingmethods for Bayesian filtering. Statistics and Computing 10(3), 197–208 (2000)
4. Le, H.-B., Pham, N.-T., Le-Nguyen, T.-V.: Applied Particle Filter in Traffic Tracking. In: Proceedings of IEEE International on Research, Innovation & Vision for the Future in Computing and Communication Technologies, RIVF 2006 (2006)
5. Hue, C., Cadre, J.P., Perez, P.: Tracking multiple objects with Particle Filtering. IEEE Transactions on Aerospace and Electronic Systems 38(3), 791–812 (2002)
6. Kravaritis, G., Mulgrew, B.: Variable-Mass Particle Filter for Road-Constrained Vehicle Tracking. EURASIP Journal on Advances in Signal Processing 2008, Article ID 321967,(1967)
7. Mallick, M., Maskell, S., Kirubarajan, T., Gordon, N.: Littoral tracking using Particle filter. In: Proceedings of the 5th International Conference on Information Fusion, Annapolis, Md, USA, vol. 2, pp. 935–942 (July 2002)

8. Mihaylova, L., Boel, R.: A Particle Filter for Freeway Traffic Estimation. In: Proceedings of 43rd IEEE Conference on Decision and Control (CDC) (2004)
9. Giebel, J., Gavrila, D.M.: Multimodal Shape Tracking with Point Distribution Models. In: Van Gool, L. (ed.) DAGM 2002. LNCS, vol. 2449. Springer, Heidelberg (2002)
10. Okuma, K., Taleghani, A., de Freitas, N., Little, J.J., Lowe, D.G.: A Boosted Particle Filter: Multitarget Detection and Tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
11. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
12. Ristic, B., Arulampalam, S., Gordon, N.: Beyond the Kalman Filter, Artech House (2004) ISBN: 1-58053-631-X
13. Vermaak, J., Doucet, A., Perez, P.: Maintaining multi-modality through mixture tracking. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, ICCV 2003 (2003)
14. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Computing Surveys 38(4) (December 2006)

Bayesian Inference on Hidden Knowledge in High-Throughput Molecular Biology Data

Viet-Anh Nguyen¹, Zdena Koukolíková-Nicola², Franco Bagnoli³,
and Pietro Lió¹

¹ Computer Laboratory, University of Cambridge, Cambridge CB3 0FD, UK

² Fachhochschule Nordwestschweiz, Hochschule fr Technik, Steinackerstrasse 5,
CH-5210 Windisch, Switzerland

³ Department of Energy, University of Florence, S. Marta, 3 50139 Firenze.

Also CSDC and INFN, sez. Firenze

van25@cam.ac.uk, z.nicola@bluewin.ch,

franco.bagnoli@unifi.it, p1219@cam.ac.uk

Abstract. Along with the information overload brought about by the Internet in communication, economics, and sociology, high-throughput biology techniques produce vast amount of data, which are usually represented in form of matrices and considered as knowledge networks. A spectral based approach has been proved useful in extracting hidden information within such networks to estimate missing data. In this paper, we propose the use of a simple nonparametric Bayesian model to fully automate this approach and better utilize the available data at each stage of the learning process. Although the algorithm is developed with a general purpose in mind, within the scope of this paper, we evaluate its performance by applying on three different examples from the field of proteomics and genetic networks. The comparison with other general or data-specific methods has shown favor to ours. Systematic tests on synthetic data are also performed, showing the approach's robustness in handling large percentage of missing data both in term of prediction accuracy and convergence rate. Finally, we describe a procedure to explore the nature of different types of noise containing within investigated systems.

1 Introduction

There is currently a tremendous growth in the amount of life science high through-put data including completely sequenced genomes, 3D protein structures, DNA chips, and mass spectroscopy data. Large amounts of data are distributed across many sources over the web, with high degree of semantic heterogeneity and different levels of quality. These data must be combined with other data and processed by statistical tools for patterns, similarities, and unusual occurrences to be observed. The results of many experiments can be summarized in a large matrix, in which rows represent repetitions of the experiment in different context, and the columns are the output of a single measurement. Within the scope of this paper we consider the following cases: microarray sampling, protein-substrate affinity, and genetic networks.

A DNA microarray (gene chip) can be seen as an ordered collection of spots, on each of which there is a different probe formed by known sequences of cDNA. A sample of mRNA, supposed to represent the gene expressed in a given tissue under investigation, is let hybridize with the probes. Fluorescent techniques allows to detect the hybridized spots. The idea is that of using probes specific for a unique region of a gene, therefore detecting the gene expressed in a tissue. The experiment is repeated for many tissues, from different part of the body, or from different patients, or from a different phase of the cellular cycle. The data can therefore be arranged using the probe numbering as column index, and tissue numbering as row index.

The interaction of proteins with an intra/extracellular ligand, protein or with the outer world (in particular concerning the immune response) depends on the shape. At present, it is not possible to reconstruct the three-dimensional shape of a protein from its primary sequence (easily obtained by mRNA sequence). Moreover, proteins are very often glycosylated, and these sugar chains attached to the outer surface may be the most important factor for inflammation. On the other hand, direct visualization of protein surface, using NMR, electronic microscopy, etc. is a very slow and costly process. A method for obtaining information about this shape is that of using proteins or antibody arrays, similar to DNA microarrays. Again, in this case, the pattern of matches can be represented as a matrix, with columns corresponding to substrates (probing proteins or antibodies) and rows to different proteins under investigation.

Biologists represent biochemical and gene networks as state transition diagrams with rates on transition. This approach provides an intuitive and qualitative understanding of the gene/protein interaction networks underlying basic cell functions through a graphical and database-oriented representation of the models. More mathematical approaches focus on modeling the relationships between biological parameters using a connectivity network, where nodes are molecules and edges represent weighted ontological/evolutionary connections [1]. Therefore a genetic network can be represented by an adjacency matrix showing the value for each gene-gene interaction.

In a very recent paper, Koukolíková-Nicola et al. [7] has demonstrated the power of the iterative spectral algorithm proposed by Maslov and Zhang [9] in inferring missing values of protein contact maps and cytokine networks. While the estimation ability of the method depends considerably on the hidden feature dimension M , there is not yet a solid theoretical way to determine its value. In this paper, we propose the use of Bayesian statistics to automatically determine the most appropriate M value at each iteration of the learning process. As the result, the value of M is changed accordingly to the amount of information available at each step, and approaches a fixed value when the predictions start to converge. We also investigate the role of nonlinearities and noise in the matching phase. In particular, it is shown that nonlinearities appear as noise when linear investigation tools are used. We introduce the use of the distribution of data correlations in combination with our Bayesian spectral approach to make prediction on the noise nature within the investigated systems.

2 Systems and Methods

We consider a set of N genetic objects. A typical biological experiment will measure their responses towards a set of D different objects or experimental conditions. As the result, we obtain a size $N \times D$ data matrix \mathbf{Y} .

We assume that each object could be well represented by a hidden feature vector of dimension M , and there exists a linear transformation from the input data space to the hidden space of interest:

$$\mathbf{y}_i = \mathbf{W}\mathbf{x}_i + \boldsymbol{\epsilon} \quad (1)$$

where \mathbf{y}_i has length D , \mathbf{x}_i has length M for $i = 1..N$. \mathbf{W} is the transformation matrix of size $D \times M$, and $\boldsymbol{\epsilon}$ is the inherent noise from data and/or information loss of linear transformation. In other words, the transformation matrix contains the contribution of each condition towards the basic M features of each object.

2.1 Inference on Missing Values

It is normal that each object would adopt a different scale of responses towards experimental triggers. To explicitly consider this fact, our prediction upon a missing value is as follows:

$$\tilde{y}_{ij} = \bar{y}_i + \sum_{k=1}^M \mathbf{w}_k(x_{ik} - \bar{x}_i) \quad (2)$$

where \bar{y}_i is the average responses of object i to D conditions, and \bar{x}_i is its corresponding average of the hidden features.

Our next step is to learn \mathbf{W} from a given data matrix \mathbf{Y} . Tipping and Bishop [15] has proved that given a specific M , the maximum likelihood estimation for \mathbf{W} under the assumption of Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ is:

$$\tilde{\mathbf{W}} = \mathbf{U}_M(\Lambda_M - \sigma^2 \mathbf{I})^{1/2} \quad (3)$$

where the $N \times M$ matrix \mathbf{U}_M is constructed by M principal eigenvectors of $\mathbf{Y}\mathbf{Y}^T$, the $M \times M$ matrix Λ_M contains M largest eigenvalues of $\mathbf{Y}\mathbf{Y}^T$. The arbitrary rotation matrix as in [15] was effectively selected as \mathbf{I} for simplicity. The square root operation is safe with the corresponding estimation of σ^2 .

However, M is not a known and fixed property in real systems. Maslov and Zhang [9] has proposed a conjecture to effectively estimate M using the knowledge of portion of missing values of symmetric data matrix. Although the conjecture sometimes proved useful [7], its usability limits to the case of symmetric input data. Here, we propose the use of Bayesian statistics to estimate M from its posterior distribution. In other word, we want to calculate:

$$p(M|\mathbf{Y}) = \frac{p(\mathbf{Y}|M)p(M)}{\int p(\mathbf{Y}|M)p(M)dM} \quad (4)$$

where the likelihood of the data given M is computed by integrating over two unknown parameters \mathbf{W} and σ^2 .

By assuming a standard Gaussian distribution over \mathbf{x}_i , the likelihood of the data given all parameters is:

$$p(\mathbf{Y}|\mathbf{W}, \sigma^2) = (2\pi)^{-ND/2} |\mathbf{V}|^{-N/2} \exp\left(-\frac{1}{2} \text{tr}((\mathbf{V})^{-1} \mathbf{P})\right) \quad (5)$$

where $\mathbf{V} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$ and $\mathbf{P} = \mathbf{Y}\mathbf{Y}^T$.

There exists a few papers [4,12,10] proposing different ways to estimate the sufficient number of principal components to keep when performing PCA, which is very close in nature to our problem. To keep the lightweight characteristic of the spectral algorithm, we base our calculation on the Laplace approximation of $p(\mathbf{Y}|M)$ proposed by [10], which leads to the following estimation:

$$p(\mathbf{Y}|M) \approx N^{-M/2} (2\pi)^{(D-M+1)M/2} \left(\prod_{i=1}^M \lambda_i\right)^{-N/2} \left(\frac{\sum_{i=M+1}^D \lambda_i}{D-M}\right)^{-N(D-M)/2} |\mathbf{A}|^{-1/2} \quad (6)$$

where $|\mathbf{A}| = \prod_{i=1}^M \prod_{j=i+1}^D N(\lambda_j^{-1} - \lambda_i^{-1})(\lambda_i - \lambda_j)$ and $\lambda_i, i = 1..D$ are the square root of the eigenvalues of $\mathbf{Y}\mathbf{Y}^T$.

Maslov and Zhang [9] provided an estimation of M_{eff} - the sufficient number of eigenvalues to keep during matrix reconstruction, taking into account the proportion of missing values of the data matrix. Here we adapt this conjecture to the case of asymmetric data, and use it as the suggestion for the upper boundary of M given m missing elements. We define the prior distribution $p(M)$ as:

$$p(M) = \begin{cases} \frac{k}{(k-1)M_{eff}+D} & \text{for } M \leq M_{eff} \\ \frac{1}{(k-1)M_{eff}+D} & \text{for } M_{eff} < M \leq D \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where $M_{eff} = \frac{ND-m}{N}$ and the empirical value $k = 3$.

Our proposed approximation algorithm to infer on missing values of data matrix \mathbf{Y} is as follows:

- (1) Construct the initial estimation $\tilde{\mathbf{Y}}$ of \mathbf{Y} by assigning 0 to all unknown positions.
- (2) Estimate the sufficient \tilde{M} for $\tilde{\mathbf{Y}}$ using equations (4), (6), and (7), the denominator in (4) is ignored since it is a constant to M .
- (3) Perform SVD on $\tilde{\mathbf{Y}}$, and construct the matrix $\tilde{\mathbf{Y}}'$ by keeping the \tilde{M} largest singular values and corresponding eigenvectors.
- (4) Reconstruct $\tilde{\mathbf{Y}}$ from $\tilde{\mathbf{Y}}'$ by filling known positions with their original values.
- (5) Go to step (2).

We repeat this process until either there is no significant change on estimated values or a maximum number of iterations has been reached. The final predictions are then constructed using (2). Although we haven't come up with a formal proof about the convergence of the algorithm, the numerical experiments suggest very good convergence rate under various proportions of missing data (section 3.1).

2.2 Inference on Data Noise

Noise affects measurements, technologies and pervades all science areas. Biological sequences are affected by several types of noise. For example functional annotation of biological sequences are often inferred by sequence similarity to homologous sequences due to the time and cost constraints of the experimental practice. Therefore biological databases contains annotation errors, and chains of misannotation due to the annotations simply copied from similar sequences in other databases [6]. Information present in genome sequence is often partitioned in vertical signals (evolutionary dependence), non-vertical signals and phylogenetic noise [3].

Given our assumption in equation (1), it is intuitive to see that if we are able to extract the true hidden features of each object, the remaining information left in the systems would be due to data noise. Here we propose the use of object correlations as the variable to explore systems noise nature.

The correlation between two objects is defined as follows:

$$C_{i,j} = \frac{\sum_{n=1}^N (y_{in} - \bar{y}_i)(y_{jn} - \bar{y}_j)}{\sqrt{\sum_{n=1}^N (y_{in} - \bar{y}_i)^2 \sum_{n=1}^N (y_{jn} - \bar{y}_j)^2}} \quad (8)$$

It ranges from -1 to 1, where 1 means the responses of the two objects are identical, 0 implies no relation, and -1 means a completely negative relation between the two objects.

The procedure to explore the noise nature within the system is as follows:

- (1) Consider each data point as the missing position, and re-estimate it given all other known values using our Bayes spectral algorithm; iterate this step through out the data set.
- (2) Construct the error matrix from the difference between the predicted and original data values.
- (3) Compute the $N \times N$ correlation matrix among object errors. Since correlations measure the similarity among objects, the distribution of error correlations provide us with suggestion about the nature of the noise within our systems. A demonstration on synthetic data comes in section 3.2.

3 Experiments and Discussions

3.1 Inference on Missing Values

Synthetic Data. We firstly tested the performance of our proposed Bayesian spectral using synthetic data. Using N , D , M as free parameters, we randomly generated two matrices \mathbf{W} and \mathbf{X} with each components to be either 1 or -1. Data matrix Y was then computed using equation (1) with randomly generated Gaussian noise. The variance of noise was set to be as large as 60% of the maximal data values. The missing positions of each data matrix were picked randomly with the percentage of missing ranges from 5% to 50%. The performance of

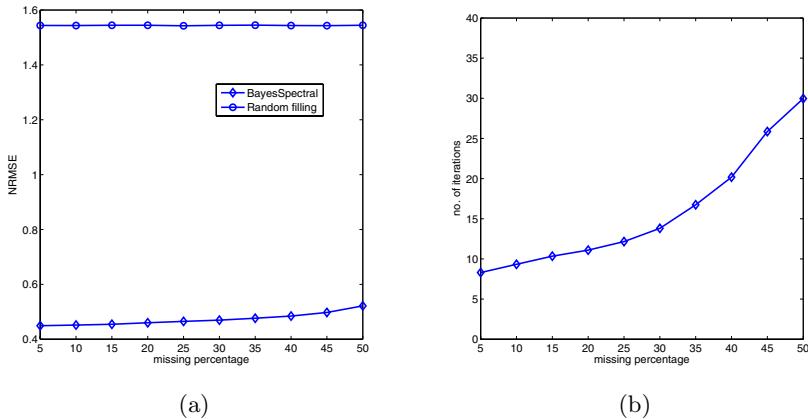


Fig. 1. Effect of various missing data proportions on BayesSpectral algorithm. (a) Average NRMSE errors of BayesSpectral and random fillings. (b) Average number of iterations needed for BayesSpectral to converge.

the algorithm on missing value prediction was evaluated by the normalized root mean square error (NRMSE), which measures the average deviation of predicted values over all missing positions.

Figure 1(a) shows the average performance from 1000 runs for increasing missing percentages of BayesSpectral compared to random filling. It could be seen that the algorithm performs reasonably well even with noisy and relatively sparse data, with a NRMSE of 0.52 for the case of 50% missing data (compared to an average of 1.55 error for random case). The algorithm is also robust in term of convergence rate with only 30 iterations needed for the case of 50% missing positions (Figure 1(b)), which takes only a few seconds with our Matlab implementation.

Cytokine Networks. Given the good performance on synthetic data, our next step was to evaluate the algorithm performance on real biological data. We first used the cytokine network from [5], which contains 29 cells communicating with each other through different kinds of cytokines. This dataset was also used by [7] to test the spectral algorithm (MZ-Spectral) proposed by Maslov and Zhang [9].

Using the same testing strategy as [7], we iteratively evaluated how well the algorithm could reconstruct each matrix position from all other known values. Table 1 compares the performance of our Bayesian version with the best result achieved by MZ-Spectral. The most basic difference between the two algorithms is that while MZ-Spectral requires a user input on the value of M and uses it throughout the process, our Bayesian Spectral automatically proposes the most appropriate M at each iteration step to best utilize the available data.

Enzyme-Ligand Data. We applied our algorithm on the enzyme-ligands data generated by [8], which contains the binding energies of 27 *E. coli* enzymes to 119 different ligands. This data was also used by [7] to evaluate the performance of

Table 1. Predictions for cellular interactions mediated by cytokines

	BayesSpectral	MZ-Spectral
Percent of cells with predicted value within 5% of the original one	11.2	9.1
Percent of cells with predicted value within 10% of the original one	21.8	17.5
Percent of cells with predicted value within 20% of the original one	39.9	33.5
Percent of cells with predicted value within 50% of the original one	79.2	78.7
Percent of cells with predicted value within 100% of the original one	100.0	100.0
Total number of cells: 418		

Table 2. Predictions for binding energies of enzyme-ligands

	BayesSpectral	DeGustibus
Percent of cells with predicted value within 5% of the original one	45.6	24.8
Percent of cells with predicted value within 10% of the original one	76.6	45.7
Percent of cells with predicted value within 20% of the original one	95.0	78.1
Percent of cells with predicted value within 50% of the original one	99.4	97.3
Percent of cells with predicted value within 100% of the original one	99.9	100.0
Total number of cells: 3213		

the correlation-based algorithm *De Gustibus* [2]. Using a similar testing scheme as above, we obtained the results in table 2, implying a considerable improvement in performance of our Bayes Spectral method.

Microarray Data. There has been a few papers proposing different methods for microarray data missing values imputation. We compared our proposed approach with the KNNimpute [16], and Bayesian PCA [11]. KNNimpute is the most popular algorithm for microarray data imputation, mostly due to its simplicity and efficiency. Bayesian PCA is an algorithm using PCA in combination with Bayesian variational estimation, which was reported to produce best results when compared with a number of different imputation methods [17].

We used the yeast data from [14], removing all the rows and columns that contain missing values. The comparison of performance using the same testing scheme as above is shown in Figure 2. We chose $K = 10$ for KNNimpute, and $K = D - 1$ for BPCA, which were reported as optimal parameter values by the authors.

3.2 Inference on Data Noise

To test the capability of the method to extract information on data noise, we performed two experiments on synthetic data where we have full understanding about the noise nature.

The first experiment corresponds to the case of random noise resulted from the random effects during experiments and the stochastic nature of molecular biology samples. We model them by a standard Gaussian distribution. The second experiment corresponds to systematic device inefficiency typically seen during microarray experiments. We model this by a filter function, using thresholds to assign discrete values to data points within specified ranges. The distorted data matrices were then fed to the procedure in section 2.2. The correlation distribution of the predicted errors are shown in Figure 3. As can be seen, the shape of

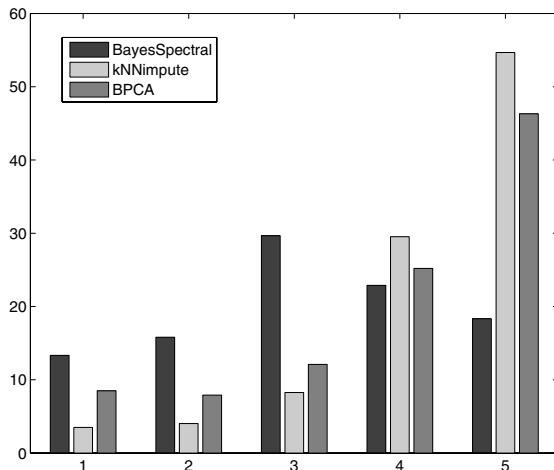


Fig. 2. *Predictions for microarray gene expression.* Comparison of the performance of our Bayes Spectral approach to KNNimpute and BPCA. Predicted values are compared to original ones and divided into 5 categories: the percentage of predicted values (1) within 5%, (2) within 10%, (3) within 20%, (4) within 50%, (5) exceed 50%, of the original values.

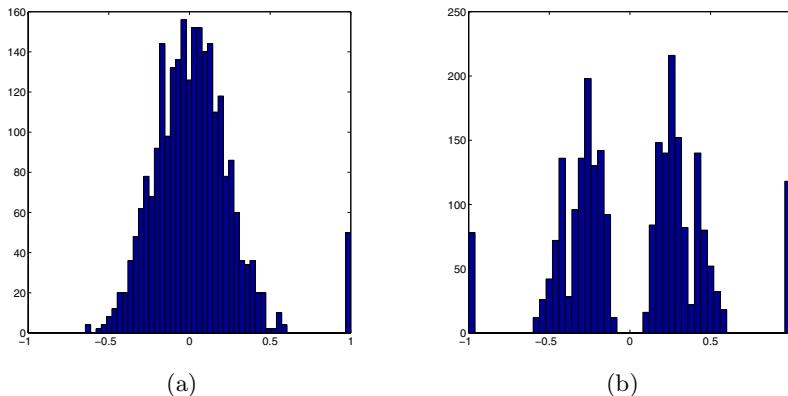


Fig. 3. *Predictions of data noise nature by BayesSpectral.* (a) Data with random Gaussian noise. (b) Data with discrete filter function.

the extracted noise's correlation distribution could be used as an effective variable to suggest us about the origin of the noise existing within our experimental data sets.

4 Conclusions

We have developed a fully-automated Bayesian spectral algorithm which proves useful in estimating missing data as well as predicting the nature of noise

containing within the investigated system. The evaluation on three different datasets of cytokine networks, enzyme-ligands, and microarray gene expressions shows significant improvement compared to other general and data-specific methods. Moreover, the investigation on synthetic data shows that the approach is very robust in handling large percentage of missing data both in terms of accurate prediction and quick convergence rate. Although a solid conclusion on the ability of our method in predicting systems noise nature has not been made, the result on synthetic data is very promising. A systematic investigation with specific consideration on different kinds of noises by experts would prove useful.

Our approach was developed with a general purpose in mind. For example it can be applied to economics or to psycho-sociological data (opinions on daily events, news, books, movies, fashion trends etc; see for instance [13]). Work in progress is to improve the prediction methodology in the areas of medical care, for instance for molecular biology data exploitation methods in cancer therapy development. Medical observations and laboratory outcomes are usually integrated by an expert physician or one prevails and the other is used for mere confirmation. Ideally, the diagnostic process should include the evaluation of molecular and clinic tests alongside medical observations by the medical practitioner. Given the massive amount of data, an AI-based support to the evaluation, integration of data from disparate data sources is urgently needed for accurate diagnosis of complex disorders which the prerequisite of appropriate and effective treatment. We really do hope that our efforts in this direction would lead to improvements in medical care.

The extraction of the maximum amount of information from high-throughput put data is of great importance to corroborate or falsify beliefs or models that are developed to describe life science or social complex systems. Noteworthy, it will lead to improvements in methods for data gathering in complex system research, data mining of large set of data, and better consideration of legal aspects of the use of proprietary data for research purposes.

Acknowledgements

Viet-Anh Nguyen is supported by the Computer Laboratory Premium Studentship, Cambridge Overseas Research Studentship, Cambridge Overseas Trust, and King's College Studentship.

References

1. Barabasi, A.L., Oltvai, Z.N.: Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.* 5, 101–113 (2004)
2. Bagnoli, F., Berrones, A., Franci, F.: De gustibus disputandum (forecasting opinions by knowledge networks). *Physica A* 332, 509–518 (2004)
3. Comas, I., Moya, A., González-Candela, F.: Phylogenetic signal and functional categories in Proteobacteria genomes. *BMC Evolutionary Biology* 7(1), S7 (2007)
4. Everson, R., Roberts, S.: Inferring the eigenvalues of covariance matrices from limited, noisy data. *IEEE Trans Signal Processing* 48, 2083–2091 (2000)

5. Frankenstein, Z., Alon, U., Cohen, I.: The immune-body cytokine network defines a social architecture of cell interactions. *Biology Direct* 1(32), 1–15 (2006)
6. Gilks, W.R., Audit, B., de Angelis, D., Tsoka, S., Ouzounis, C.A.: Percolation of annotation errors through hierarchically structured protein sequence databases. *Math. Biosci.* 193(2), 223–234 (2005)
7. Assareh, A., Moradi, M.H., Volkert, L.G.: A hybrid random subspace classifier fusion approach for protein mass spectra classification. In: Marchiori, E., Moore, J.H. (eds.) *EvoBIO 2008. LNCS*, vol. 4973, pp. 1–11. Springer, Heidelberg (2008)
8. Macchiarulo, A., Nobel, I., Thornton, J.: Ligand selectivity and competition between enzymes in silico. *Nature Biotechnology* 22(8), 1039–1045 (2004)
9. Maslov, S., Zhang, Y.-C.: Extracting Hidden Information from Knowledge Networks. *Physical Review Letters* 87(24), 1–4 (2001)
10. Minka, T.: Automatic choice of dimensionality for PCA. *Neural Information Processing Systems* 13 (2000)
11. Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K., Ishii, S.: A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics* 19(16), 2088–2096 (2003)
12. Rajan, J.J., Rayner, P.J.W.: Model order selection for the singular value decomposition and the discrete Karhunen-Loeve transform using a Bayesian approach. *IEE Vision, Image and Signal Processing* 144, 123–166 (1997)
13. Smarts is a Web-based, person-to-person betting exchange for Amazon Products, <http://www.midasoracle.org/2008/03/28/smarts/>
14. Spellman, R., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D., Futcher, B.: Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization. *Molecular Biology of the Cell* 9, 3273–3297 (1998)
15. Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B* 61(3), 611–622 (1999)
16. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.: Missing value estimation methods for DNA microarrays. *Bioinformatics* 17(6), 520–525 (2001)
17. Wong, D.S.V., Wong, F.K., Wood, G.R.: A multi-stage approach to clustering and imputation of gene expression profiles. *Bioinformatics* 23(8), 998–1005 (2007)

Personalized Search Using ODP-based User Profiles Created from User Bookmark

Tetsuya Oishi, Yoshiaki Kambara, Tsunenori Mine, Ryuzo Hasegawa,
Hiroshi Fujita, and Miyuki Koshimura

Department of Intelligent Systems, Kyushu University,
744 Motooka, Nishiku, Fukuoka 819-0395, Japan

{[oishi](mailto:oishi@is.kyushu-u.ac.jp),[kambara](mailto:kambara@is.kyushu-u.ac.jp),[hasegawa](mailto:hasegawa@is.kyushu-u.ac.jp),[fujita](mailto:fujita@is.kyushu-u.ac.jp),[koshi](mailto:koshi@is.kyushu-u.ac.jp)}@ar.is.kyushu-u.ac.jp,
mine@al.is.kyushu-u.ac.jp

Abstract. When searching for intended pages on the Internet, users often have a hard time to find the pages because the pages do not always come at the higher rank of searched results. The Personalized Search is a promising approach to solve this problem. In the Personalized Search, User Profiles (UPs in short) that represent interests of the users, are well used and often created from personal documents of the users. This paper proposes (1) a method for making UPs based on Open Directory Project (ODP) and shows (2) a Personalized Search system using the UPs made from Book Marks. Some of experimental results illustrate the validity of our method for making the UPs, and show the precision enhancement of this system.

1 Introduction

The rapid spread of the Internet has brought about a big revolution in information technologies and information environments. The development of the World Wide Web (WWW) especially makes available a lot of knowledge and useful means for accessing electronic information entities. However, it is still not easy task to find pages intended by users with a search engine because most queries issued to the search engine are ambiguous and there are lots of possibilities as information relevant to the queries. As the results, many irrelevant information come up at the higher rank of retrieved results returned by the search engine.

The Personalized Search is a promising approach to solve this problem. In the Personalized Search, User Profiles (UPs in short) that represent interests of the users, are well used and often created from their personal documents such as e-mail messages, book marks, click histories and so on. When making UPs, we analyze the personal documents, extract salient words with their statistical information and make word vectors from them. Such extracted words are sometimes classified into some categories.

In this paper, we propose a method for creating UPs based on the categories of the Open Directory Project (ODP in short)¹. Each directory provided by the

¹ Open Directory Project: <http://www.dmoz.org/>

ODP is corresponding to one category and keeps several Web pages selected by hand. Therefore we can say that each directory expresses a clear meaning and is distinguished with one another. Our method creates a UP that is expressed as a category vector of the topmost categories of the ODP, and reranks, using the UP, retrieved Web pages returned by a search engine.

We conducted several experiments to illustrate the validity of our method. We used user bookmark to create long-term UPs. Some of highlighted results showed that the created UPs are useful for improving precision and MAP scores, especially for the queries with some ambiguous.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 describes how to make UPs. Section 4 shows the overview of our personalized search system and experimental results are discussed in the section 5. Lastly we conclude the paper and discuss our future work.

2 Related Work

A lot of studies on personalizing search have been done so far. Here, we focus on personalized search with UPs.

Chirita et al. [2][3] compared their personalized search results with Google search results using user queries classified into three types: clear queries, semi-ambiguous queries and ambiguous queries. In [2], they explored methods for query extension from desktop documents and in [3], used the ODP to create a UP and explored the hierarchical categories of the ODP.

Dou et al.[4] described the characteristics of UPs changed by the Personal Document. They classified this feature into two types of UPs : long-term UPs (daily hobby, interests, and so forth) and short-term UPs (temporal examination or purposes, and so on).

In our study, we propose a personalized search system that uses multiple UPs created from user bookmarks, which are corresponding to the long-term UPs, and lets a user select one of the UPs when searching for Web pages relevant to his/her query. This method makes a difference especially for ambiguous queries.

3 User Profile Based on ODP

3.1 Open Directory Project

The Open Directory Project (ODP)² is the largest, most comprehensive human-edited directory of the Web. It is constructed and maintained by a vast, global community of volunteer editors[1]. Since all the Web pages in the ODP categories are classified by hand, the classification accuracy is fairly high. In addition, one Web page is put into only one category in principle³.

² <http://www.dmoz.org/>

³ When the Web page is related to “Region”, it might be put into more than one categories(“Region” and the rest).

Table 1. An example of UP

User	Art	Sport	Computer
A	0.4	0.1	0.5
B	0.2	0.7	0.1

3.2 How to Make UPs

First, we will show the detail of a UP. The UP proposed in this paper is defined as a category vector created from the topmost categories in the ODP⁴, and each element of the vector takes a numerical value whose range is from 0 to 1.

Table 1 shows an example of a UP whose significant categories are “Art”, “Sport” and “Computer”. We can see that user A is strongly interested in “Art” and “Computer”, but not in “Sport”. On the other hand, user B is interested in “Sport”, but not in the other two so much. Next, we will show the procedure on how to make a UP.

Analysis of Words Extracted from Categories in the ODP

1. Word Extraction

For each topmost category in the ODP, words are extracted from the name and description of its sub-categories. To do this, morphological analyzer MeCab⁵ is used. Here, our target documents are Japanese ones.

2. Weight of Word

We calculate $W(t_i, c_j)$, which is the weight of word t_i to category c_j .

First, we define entropy H_{t_i} of the word t_i by equation (1) so that the bias of t_i occurring in a category is calculated.

$$H_{t_i} = - \sum_{j=1}^{N_c} P_{t_i}(c_j) * \log(P_{t_i}(c_j)) \quad (1)$$

Where N_c is the total number of categories, $n(t_i, c_j)$ is the occurrence frequency of word t_i in category c_j , and $P_{t_i}(c_j) = n(t_i, c_j) / \sum_{j=1}^{N_c} n(t_i, c_j)$.

When t_i uniformly occurs in every category, i.e., $P_{t_i}(c_j) = 1/N_c$, H_{t_i} takes the maximum value $\log(N_c)$. If t_i occurs only a certain category, H_{t_i} takes 0. With this entropy H_{t_i} , weight w_{t_i} of word t_i is defined by equation (2).

$$w_{t_i} = \log(N_c) - H_{t_i} \quad (2)$$

As the occurrence of word t_i is biased toward less number of specific categories, the value of w_{t_i} becomes greater. $W(t_i, c_j)$ is finally defined by equation (3).

$$W(t_i, c_j) = P_{t_i}(c_j) * w_{t_i} \quad (3)$$

⁴ “Art”, “Online-shop”, “Game”, “Computer”, “Sport”, “News”, “Business”, “Recreation”, “Home”, “Science”, “Various material”, “Health”, “Social”, and “Region”.

⁵ <http://mecab.sourceforge.net/>

Table 2. (A) Example of Extracted Words and their Occurrence Frequency. (B) Entropy H_{t_i} and Weight of Word t_i , w_{t_i} . (C) Weight of word.

(A)				(B)		(C)		
word	Art	Sport	Computer	H_{t_i}	w_{t_i}	Art	Sport	Computer
t_1 soccer	2	34	1	0.48	1.11	0.06	1.02	0.03
t_2 book	15	8	13	1.54	0.05	0.02	0.01	0.02

With the $W(t_i, c_j)$, the weight of category c_j is calculated by the following equation. $W(c_j) = \sum_{t_i \in T} W(t_i, c_j)$ Where T is a set of all the words that occur in all the categories of the ODP.

As an example, the occurrence frequencies of words are shown in table 2(A). Their entropy and weight values are shown in table 2(B). In the example, word t_1 (soccer) occurs twice in category “Art”, 34 times in “Sport” and once in “Computer”. The entropy H_{t_1} of the word t_1 that is biased toward category “Sports”, takes a small value 0.48, and the weight value w_{t_1} becomes 1.11 by equation (2). On the other hand, since word t_2 (book) almost evenly occurs in each category, the value of entropy H_{t_2} approaches the maximum value $\log(3) = 1.59$, and thus weight w_{t_2} becomes a smaller value 0.05.

Extraction of words occurred in Personal Documents. We also extract word u_t that occurs in Personal Documents as the same way as word extraction from the categories in the ODP.

Creation of UP. U_{c_j} , that represents the degree of user’s interests in category c_j in a UP, is defined by equation (4).

$$U_{c_j} = \sum_i N_{u_{t_i}} * W(u_{t_i}, c_j) \quad (4)$$

Where u_{t_i} represents a word in user’s personal documents and $N_{u_{t_i}}$ the occurrence frequency of word u_{t_i} in his/her personal documents. $W(u_{t_i}, c_j)$ is calculated by applying equation (3) to u_{t_i} , that is, the word occurrence distribution of the categories of the ODP is used for this calculation. Applying the equation (4) to all the categories, the UP is finally created just as shown in the following vectors $\mathbf{UP} = [U_{c_1}, U_{c_2}, \dots, U_{c_{N_c}}]$. The UP vector is normalized so as to be $|\mathbf{UP}| = 1$. We assume that only two words of “Soccer” and “Book” are extracted from a personal document. The occurrence frequencies of the words are $\{soccer, book\} = \{4, 13\}$. Using equation (4), $\mathbf{UP} = [0.12, 0.99, 0.08]$. In this case, we can see the UP is biased toward category “Sports”.

4 Personalized Search System

A lot of studies on personalized system using UPs have been done. The UPs have been created from a various kinds of personal documents such as e-mail messages,

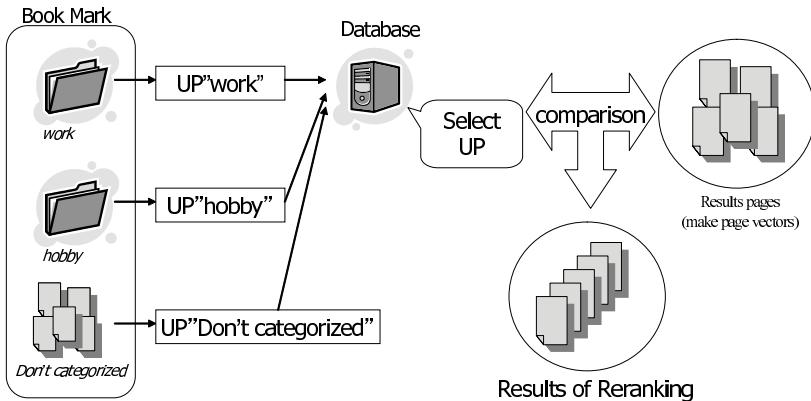


Fig. 1. Personalized Search System

Web browsing histories, retrieved Web pages, click histories, other desk top documents. In this paper, we focus on user bookmark as the personal documents.

4.1 Overview of Personalized Search System

The overview of our personalized search system is shown in Figure 1. The process flows of the system consists of two steps: Creating UPs from bookmarks, searching for Web pages relevant to a user query with a search engine, re-ranking them using the UPs and presenting the re-ranked results.

Making a User profile from bookmark. A bookmark folder keeps the Internet short cuts, which are links to the Web pages. A UP is created from each bookmark folder. A set of unclassified bookmarks is considered as being belonged to one bookmark folder, say un-classified-folder, and they creates another UP, say unclassified-UP. The target area of extracting words is the content of a Web page linked by an Internet short cut. Although it might be concerned that the number of noisy words is increased when the words are extracted from the contents of Web pages, it would not cause a big problem because similar Web pages are usually gathered in a bookmark folder by the user. The UPs are created by the same way as mentioned in section 3.2.

For example, the user has the bookmark shown in table 3. In this case, the values of "Sports" and "Art" become higher than others for his/her bookmark "Hobby" because it has Web pages related to soccer and music. In addition, his/her bookmark "Research" has significant categories on "Computer" and "Science", bookmark "Shopping" has a peak only on "Online shop" category . As mentioned earlier, the system also makes the UP of "Unclassified" bookmark, and the bookmarks has significant categories such as "Shop" and "Weather" in this example.

Searching, Re-Ranking and Displaying. When searching for certain intended pages, a user first selects a UP related to a query from a list of created UPs. For ex-

Table 3. Example of Bookmark

Bookmarks	Contents
Hobby	5 pages of soccer 2 pages of music
Research	4pages of computer and science
Shopping	3 pages of online shop
Unclassified	4 pages of shops, weather and so on.

ample, when the user wants to search for a page related to Hidetoshi Nakata, who was a soccer player, with query “Nakata”, s/he will select UP “Hobby”. In another case, when s/he wants Hidetoshi Nakata’s goods, such as a hat, T-shirts, soccer ball with his autograph, s/he will select UP “Shop”. Then, the user issues the query to a search engine. In this paper, we use the search engine Yahoo!developer⁶. The system makes a set of page vectors of the top 100 ranked of Web pages returned by the search engine. The page vectors are made by the same way as the method mentioned in section 3.2, although the words to create page vectors are extracted from the Snippet that are the description of each page returned by the search engine. The similarity between each page vector X and selected UP Y is calculated with the cosine measure. Since both X and Y are normalized, we calculate the inner product of X and Y instead of the cosine measure. The top 100 results are re-ranked according to the similarity, and the re-ranked results are displayed.

5 Experiments

5.1 Verification of User Profile

The user profile is created from the words that are extracted from the Web pages registered in the ODP as mentioned in section 3.2. Where in bookmark folders, there might coexist the Web pages that are categorized into various categories. We first investigate the influence of increasing number of categories in the bookmark folders. Figure 2 shows the graphs that compare the ideal user profile that only represents specified categories such as art, computer, sports, online and recreation, with the user profile created from the user bookmarks. To create a UP, we prepared 5 Web pages per each category. That means, 2 category UP consists of 10 Web pages and 3 category UP consists of 15 Web pages. The results show that the values of categories of added Web pages are almost greater than others except for category “recreation” of the 5 category UP. This would be caused by increasing noisy words unrelated to specified categories.

In table 4, the similarity between created UPs and the ideal user profiles keeps the value between 0.7 and 0.8. Consequently the created UP would be appropriate for personalizing a search system because the user profiles reflect the characteristics of Web pages in the bookmark folder even if the various categories coexist.

⁶ <http://developer.yahoo.co.jp/>

“art”, “computer”, “art”, “computer”, “sports”, “art”, “computer”, “sports”,
and “sports” and “online” “online”, and “recreation”

Fig. 2. The user profile created from Web pages categorized into the various categories (3 categories (left), 4 categories (middle) and 5 categories (right)). The horizontal line represents the topmost categories of the ODP and vertical line represents the value of U_{c_i} defined in equation (4).

Table 4. The comparison of similarity between user profile created from the various categories with ideal user profile

the number of categories	2	3	4	5
similarity	0.743	0.734	0.796	0.785

5.2 Preparation

We conduct several experiments to evaluate our personalized search system, comparing original search results returned by a search engine with the results re-ranked by our system. As mentioned in section 3, we use user bookmarks to create long-term UPs. In the experiments, we use 100 queries given by undergraduate students at universities. For each query, an appropriate UP is selected from a set of UPs. First, each query is issued to search engine Yahoo! Developer network⁷. Next, top-100 Web pages returned are re-ranked with the UP. Each user who issued queries judges the relevancy of retrieved Web pages by him/herself according to his/her criteria. The judge is whether the retrieved Web pages are relevant or not to the queries. Then, we compare the results returned by Yahoo search engine with the re-ranked results using the following evaluation metrics: Precision, Recall and Mean Average Precision (MAP).

⁷ <http://developer.yahoo.co.jp/>

Table 5. Results averaged for all the 100 Queries

top-10 of the retrieved results						
Yahoo			Proposed System			
precision	recall	MAP	precision	recall	MAP	Improvement rate of MAP
0.4240	0.1509	0.096	0.5380	0.2795	0.176	1.83
top-20 of the retrieved results						
Yahoo			Proposed System			
precision	recall	MAP	precision	recall	MAP	Improvement rate of MAP
0.4075	0.2795	0.153	0.5150	0.4199	0.271	1.77

Firstly, Precision and Recall are defined as follows:

$$Precision = \frac{n_x}{x}, \quad Recall = \frac{n_x}{n_c}$$

Where n_c and n_x are the number of relevant Web pages in top-100, the number of relevant Web pages retrieved in top- x , respectively.

Next, Average Precision (AP_i) for query i is defined as follows:

$$AP_i = \frac{1}{N_i} \sum_{k=1}^{M_i} \frac{k}{r_{i,k}}$$

Where N_i , M_i and $r_{i,k}$ are the number of Web pages relevant to query i , the number of relevant Web pages found and the order of the k -th found Web page relevant to query i , respectively.

Lastly, MAP is defined as follows:

$$MAP = \frac{1}{N} \times \sum_i^N AP_i \quad (N \text{ is the number of queries.})$$

5.3 Discussion

Experimental results are shown in Table 5. The improvement rates of MAP are about 1.83 times in top-10, and 1.77 times in top-20. Figure 3 compares the MAP of different UPs that are ordered by the number of queries used to. The UPs that make their MAP value improved are UP4, UP6, and UP10, that are related to “Online Shop”, “Game” and “Book”, respectively. On the other hand, the UPs that does not make the MAP improved so much are UP1, UP7, UP9. They are “Recreation”, “Science” and “News”, respectively. The less MAP values UPs originally take, the greater MAP values they get on average after re-ranking, although UP9 is exceptional due to less relevant Web pages originally retrieved by the search engine.

In addition, we classified the queries to the three types of queries that take the Yahoo’s precision ratio of less than 20%, between 20% and 70%, and greater than 70%. The three types of queries are corresponding to ambiguous queries, semi-ambiguous queries and clear queries from the point of view of Yahoo search engine. The results are shown in table 6. From the results, as the number of ambiguous queries issued increases, we get greater number of significant re-ranking effects. These results show that re-ranking with the ODP-based UPs help to improve retrieved results returned by Yahoo search engine. Finally, we investigate the queries whose re-ranked results were not improved.

Table 6. Retrieved Results classified by Query Type

Queries (# of queries)		Yahoo			Proposed System			MAP Improvement
		Precision	Recall	MAP	Precision	Recall	MAP	
clear query (21)	top-10	0.9429	0.1469	0.143	0.9191	0.1410	0.134	0.94
	top-20	0.8738	0.2675	0.254	0.9071	0.2762	0.257	1.01
semi-ambiguous query (38)	top-10	0.4737	0.1763	0.115	0.5895	0.2115	0.161	1.40
	top-20	0.4434	0.3079	0.177	0.5711	0.3982	0.278	1.57
ambiguous query (41)	top-10	0.1122	0.1294	0.054	0.2951	0.3211	0.211	3.91
	top-20	0.1354	0.2594	0.080	0.2622	0.5136	0.266	3.33

Fig. 3. MAP at top-10 (Top) and top-20 (Bottom) of retrieved results

When there are several salient peaks of the categories in page vectors of retrieved Web pages, re-ranking those pages with the ODP-based UPs help to improve the MAP score. On the other hand, if there are only one or two peaks of categories in the page vectors, re-ranking does not work well. The main reason is that the UP is made from only 14 topmost categories of the ODP⁸. ODP's category "art" has sub-categories such as "comic", "animation", "visual art",

⁸ We used the Japanese ODP.

“music”, and so on. For example, we assume that after performing the personalized search using the UP with a remarkable category “art”, we luckily get a lot of Web pages related to the category “art” in 100 results. Since there are 15 sub-categories in “Art”, such as “Comic”, “Visual Art”, “Animation”, “Movies” and so on, the results may be classified more than one sub-categories. Accordingly even if the user is interested in the sub-category “Music”, the user may obtain other Web pages that are not related to “Music”, but “Comic” or “Visual art”. One of the solutions of this problem is to handle not only topmost categories, but also their sub-categories for creating UPs. However, there exists some trade-off. As UPs are presented in more detail, the longer time of re-ranking with the UPs is required. Moreover, since the re-ranking process becomes more sensitive, the robustness of the process might be lost. This is future work.

6 Conclusion

In this paper, we proposed an ODP-based user profile (UP) and the personalized search system using the UPs. We discussed the validity of our method and showed the effectiveness of our personalized search system using the UPs created from user bookmarks. Each UP has the features of the Web pages stored in every bookmark folder. Even if the Web pages in a bookmark folder can be categorized into more than one category, the UP reflects the feature corresponding to the categories.

Experimental results showed that the ODP-based UPs created from user bookmarks help to improve precision and MAP score. In particular, as the issued queries are ambiguous, the MAP score becomes more improved. This result is the same as one reported in previous works (e.g. [2,3]).

Although the paper presented the effectiveness of the ODP-based UPs, there still exist a lot of future work. Some of them are UP extension, refinement of words extracted from the ODP categories, coping with short-term UPs, conducting experiments with large-scale data, comparison with or adoption of clustering methods and so forth.

References

1. About the Open Directory Project, <http://www.dmoz.org/about.html>
2. Chirita, P.A., Firan, C., Nejdl, W.: Summarizing local context to personalize global web search. In: Proc. of CIKM 2006 (2006)
3. Chirita, P.A., Nejdl, W., Paiu, R., Kohlschütter, C.: Using ODP metadata to personalize search. In: Proc. of SIGIR (2005)
4. Dou, Z., Song, R., Wen, J.: A Large-scale Evaluation and Analysis of Personalized Search Strategies. In: WWW 2007 (2007)
5. Qiu, F., Cho, J.: Automatic identification of user interest for personalized search. In: WWW 2006 (2006)
6. Nagata, H.: Personalized Search System using ODP-based User Profiles. In: JAWS 2007 (2007)

Domain-Driven Local Exceptional Pattern Mining for Detecting Stock Price Manipulation

Yuming Ou, Longbing Cao, Chao Luo, and Chengqi Zhang

Faculty of Information Technology, University of Technology, Sydney, Australia

{yuming, lbciao, chaoluo, chengqi}@it.uts.edu.au

Abstract. Recently, a new data mining methodology, Domain Driven Data Mining (D^3M), has been developed. On top of data-centered pattern mining, D^3M generally targets the actionable knowledge discovery under domain-specific circumstances. It strongly appreciates the involvement of domain intelligence in the whole process of data mining, and consequently leads to the deliverables that can satisfy business user needs and decision-making. Following the methodology of D^3M , this paper investigates local exceptional patterns in real-life microstructure stock data for detecting stock price manipulations. Different from existing pattern analysis mainly on interday data, we deal with tick-by-tick data. Our approach proposes new mechanisms for constructing microstructure order sequences by involving domain factors and business logics, and for measuring the interestingness of patterns from business concern perspective. Real-life data experiments on an exchange data demonstrate that the outcomes generated by following D^3M can satisfy business expectations and support business users to take actions for market surveillance.

1 Introduction

The traditional data mining is a data-driven trial-and-error process in which data is used to create and verify research innovations. Typically, it aims to build standard models to summarise training and test data well. However, the general patterns emerged from standard models is unactionable to business needs, and cannot support business users to take decision-making actions in the business world. This may be due to many reasons. One of key reasons is that the domain knowledge is underemphasised by the traditional data mining. For instance, in the area of stock market surveillance [1], the pattern that *if there is a sharp price change then an alert is generated to indicate the occurrence of price manipulation* often leads to too many false positive alerts. The reason for that is the pattern does not take the real-life factors into consideration and embed the business logics in stock markets. The simple ignorance of domain factors and pure data-centered pattern mining result in a big gap between academic deliverables and business expectations.

To deal with the above issues, Domain Driven Data Mining (D^3M) [2, 3, 4, 5] was recently proposed targeting actionable knowledge discovery for real user needs under domain-specific circumstances. It aims to narrow down the gap between academic deliverables and business expectations through catering for key issues surrounding

real-world actionable knowledge discovery. (1) D³M makes full use of both real-life data intelligence and domain intelligence, for instance, all real-life constraints [6] including domain constraints, data constraints and deliverable constraints during the whole mining process. (2) The resulting outcomes are evaluated by not only technical but also business interestingness metrics toward knowledge actionability [5]. D³M has attracted more and more attention including workshops with KDD and ICDM.

In stock markets, the detection of intraday price manipulation is of great importance to market integrity. However, it is very challenging to effectively detect price manipulation in real markets. Based on D³M, this paper carries out a study on mining actionable local exceptional patterns indicating price manipulation on intraday trading data for market surveillance. We fully employ domain knowledge through the mining process. We first define a five-dimension vector to represent trading orders based on domain knowledge. The order vector is designed specifically for the stock market domain and captures the characteristics of stock market properly. Furthermore, we develop two interestingness measures for pattern mining which also reflect the business concerns. We deploy our approach in mining real-life orderbook data, and evaluate the mined patterns in terms of business concerns, which shows the findings are deliverable to business users for market surveillance.

The remainder of this paper is organised as follows. Section 2 introduces the related work on the actionable knowledge discovery. In Section 3, the domain problem is carefully studied with the involvement of domain intelligence. Base on the analysis in Section 3, Section 4 proposes an approach to construct the domain-driven multidimensional sequences. To discover local exceptional patterns, two interestingness measures and an algorithm are designed for identifying such local exceptional patterns in Section 5. Experiments and performance evaluation are demonstrated in Section 6. We conclude this paper and present our future work in Section 7.

2 Related Work on Actionable Knowledge Discovery

Recently the actionable capability of discovered knowledge has been payed more and more attention [7, 8, 9], a typical work is on D³M. D³M aims to narrow down the gap between academia and business, and a paradigm shift from data-centered hidden pattern mining to domain-driven actionable knowledge discovery to support decision making [4]. In D³M, both data and domain knowledge make contributions to the outputs.

The concept of actionability was initially studied from the interestingness perspective [10, 11, 12, 13, 14] to filter out the redundant and ‘explicit’ patterns through the mining process or at the stage of post-processing [15]. A pattern is *actionable* if a user can get benefits from taking actions on it (e.g., *profit* [16, 17]). Particularly, subjective measures such as *unexpectedness* [13, 14], *actionability* [2, 9, 14, 15] and *novelty* [18] were studied to evaluate the actionable capability of a pattern. However these research efforts mainly focus on the development of general business interestingness measures. Due to the absence of domain-specific knowledge, the general business interestingness measures are often inefficient when they are applied to various domains. Thus, in order to enhance the actionable capability of discovered knowledge, a reasonable assumption is that the domain-specific intelligence should be involved in the whole mining process and the business interestingness measures should also be studied in

terms of specific domain problems. In this paper, we demonstrate the development of domain-specific interestingness to measure actionability.

3 Domain Problem Definition

In the stock market, one of typical illegal trading behaviour is price manipulation. Price manipulation is defined as the trading behaviour attempting to raise or lower the price of a security for the purpose of exceptionally high profit. Price manipulation can damage the market integrity and ruin the market reputation. Consequently, any market regulators in the world are keen on developing effective detection, combating and prevention tools for price manipulation.

3.1 Current Methods

To detect price manipulation, current methods mainly focus on the sharp price changes and/or large trade volumes. When a sharp price change or a large trade volume occurs in the market, an alert is triggered and then a further investigation may be carried out. However, these methods are far from working well. As the stock market is a complex system, there are too many factors which can affect the stock price and trade volume. For example, the disclosure of a really bad news for a certain company is likely to make the stock price of that company go down dramatically. On the other hand, the experienced manipulators may manipulate the stock price without sharp price changes and large trade volumes. For example, manipulators can split a big order into several small ones to trade, or place a large-scale order on the orderbook that cannot be traded but still has great impact on the market. Obviously, the current methods are incompetent for dealing with these cases. The key reason is that current methods normally deal with price or volume only, while price manipulation is a dynamic emergence of trading behaviour that needs to be catered from microstructure perspective.

3.2 Scenario Analysis into Approach Design

Our approach is based on the following foundation: (1) analysing trading behaviour from microstructure perspective, (2) involving domain knowledge and market factors, and (3) implementing data mining process by following the principle of D³M.

In the stock market, an order is an instruction made by a trader to purchase or sell a security under certain condition. Traders enter their orders into the market and trade with others for making money. Both the attributes of orders and the way of entering orders are consistent with traders' intention. For instance, a trader who is urgent to sell and does not care about the return much will enter a sell order with a lower price which is easy to trade. However, a trader who does care much about the return and does not want to sell in haste will enter a sell order with a higher price. In fact, as well known by domain experts, manipulators also use orders to achieve their purposes. They manipulate the market by placing a series of particular buy or sell orders into the market. These tricky orders create an artificial, false, or misleading market appearance that misguide public traders but affiliate the manipulators for opportunities to make extra profit.

The above scenario analysis shows that two items of important domain knowledge. First, there is a strong connection between a trader's intentions and his/her trading activities reflected through entering orders. Second, it is reasonable to investigate order sequences for scrutinizing price manipulations. These indicate that the trading patterns of those genuine traders are different from fraudulent manipulators'. This can be through analysing order sequences to identify the difference. Once the exceptional patterns of entering orders are detected, it is reasonable to believe that there are likely suspicious trading behaviours taking place.

Inspired by the above scenarios analysis, this paper proposes an approach to discover the local exceptional patterns of order sequences for detecting price manipulation. Our approach has the following key steps: (1) constructing order sequences based on domain knowledge and scenario analysis, (2) defining interestingness specific for mining trading patterns catering for the domain issues. The *market micro-structure patterns* [19] discovered by our approach can really power market surveillance system. We interpret them in detail in the following sections.

4 Domain-Driven Multidimensional Sequence Construction

4.1 Vector-Based Order Sequence Representation

Before mining patterns, it is necessary to represent and construct order sequences in the way reflecting market mechanism and trader's intention properly. In the stock market, orders have many attributes. Some attributes have categorical values like trade direction, and other attributes have continuous values such as price and volume. Though the values of these attributes vary from order to order, they are set according to the traders' own intentions. In addition, orders have their lifecycle, and may present in certain state at a single time point:

$$\{enter, trade\ partly, trade\ entirely, delete, and outstanding\}.$$

Two orders with the same values of attribute when they are entered into the market, however, may pass through different stages later on under different circumstances. That means the order's lifecycle also has a connection to the trader's purpose. From the above analysis, we can learn that both the information of order's attributes and the information of orders' lifecycle are related to the trader's intention directly or indirectly. Therefore, a reliable representation of market order should meet the requirement that it covers all the information of order's attributes and order's lifecycle.

In our approach, a five-dimension vector $O(d, p, v, t, b)$ is defined to represent the order. Among the five dimensions, dimension $d \in \{B, S\}$ reflects the trade direction of order, dimension $p \in \{H, M, L\}$ stands for the probability that the order can be traded, dimension $v \in \{S, M, L\}$ measures the size of order, dimension $t \in \{N, O, S\}$ represents how many trades the order leads to and dimension $b \in \{C, O, D\}$ reflects the balance of order when the market closes.

From the above definition and formulas, it can be learned that our five-dimensional vector contains the information not only of the order's attributes but also of the lifecycle which the order has passed through. The dimension d , p and v contain the

information of order's attributions while the dimension t and b contain the information of order's lifecycle. Consequently, it satisfies the requirement of reliable representation of order.

4.2 Constructing Multidimensional Sequences

To construct order sequences, there is a need to decide the time range of sequence first. In the stock market, orders last for not more than one day. Traders can enter their orders after the market opens and the orders which have not been traded expire at the market closing time. This means that orders have only one-day impact on the market. According to this domain-specific characteristic, we construct the order sequences with time range of one day. The second issue is how to divide orders into sequences. There are so many orders placed by traders in a trading day. It is unreasonable to put all the orders into a sequence. Recall that traders use a series of orders to implement their own intentions. Consequently, it is rational to assign all the orders placed by the same trader to a sequence.

A multidimensional sequence (for short sequence) \mathcal{Q} is defined as the sequence of orders placed by a same trader within a trading day,

$$\mathcal{Q} = \{O_1(d_1, p_1, v_1, t_1, b_1), O_2(d_2, p_2, v_2, t_2, b_2), \dots, O_i(d_i, p_i, v_i, t_i, b_i), \dots\} \quad (1)$$

in which O_i is the five-dimensional vector defined in Section 4.1.

5 Mining Local Exceptional Patterns

5.1 Targeted Data and Benchmark Data

In the area of market surveillance, the exceptional patterns are more interesting. However, there are two kinds of exceptional patterns: *global exceptional patterns* and *local exceptional patterns*. The *global exceptional patterns* are the patterns which are exceptional for the whole data, while the *local exceptional patterns* are the patterns which are exceptional only for the local data. As the stock market is a dynamic system changing very fast, a pattern is exceptional for this period but may be normal for another period. Besides, a pattern is exceptional for a long period but may be normal for a short period. Consequently, the *global exceptional patterns* do not mean much in our case. We are interested in identifying the *local exceptional patterns*.

The definitions of *targeted data* and *benchmark data* are based on a sliding time window. As shown in the Fig. 1, there is a sliding time window with size of $m + 1$ days. Among these $m + 1$ days falling into the sliding time window, the first m days are called *benchmark day 1, 2, ..., m* respectively, and the last day is called *targeted day*. Furthermore, the data drawn from the *benchmark day i* is called *benchmark data i* while the data drawn from the *targeted day* is called *targeted data*. With the movement of the sliding time window from left-hand side to right-hand side, any day can be the *targeted day* and has its corresponding *benchmark days*. Because the *benchmark days* are the close neighbours of the *targeted day*, there are correlations between the *targeted day* and its *benchmark days*. Intuitively, the closer the *benchmark day* is, the bigger the degree of correlation is. Therefore, each *benchmark day* is assigned a weight by the following formula:

$$W_i = (1 + \gamma)^{i-1} \quad (2)$$

Where $\gamma \geq 0$ reflects the volatility of market.

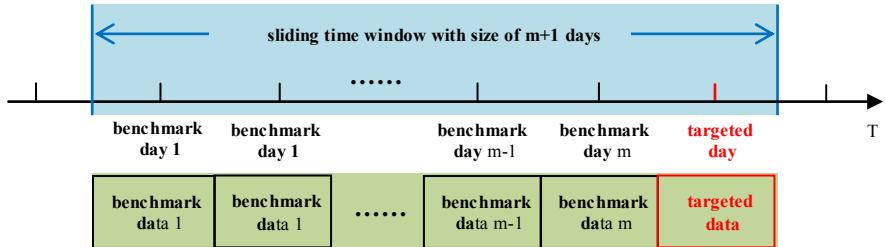


Fig. 1. Targeted data and benchmark data

5.2 Algorithms for Mining Local Exceptional Patterns

Definition 1. Intentional Interestingness (II): II quantifies the intentional interestingness of pattern as defined in the following formula:

$$II = Sup_t \times \frac{|\Omega|}{AvgL_t} \quad (3)$$

where

- Sup_t is the support of sequence Ω in the *targeted data*,
- $|\Omega|$ is the length of sequence Ω ,
- $AvgL_t$ is the weighted average length of sequences in the *targeted data*,

II is positively related to the support in the *targeted data* and the length of pattern. The idea behind this measure is very straightforward. As it is said before, traders tend to use a series of orders to implement their intentions. Therefore the order sequence with a higher support and a longer length is placed more intentionally in the *targeted day*.

Definition 2. Exceptional Interestingness (EI): EI quantifies the exceptional interestingness of pattern as defined in the following formula:

$$EI = \frac{\frac{Sup_t}{AvgL_t} \times \sum_{i=1}^m W_i}{\sum_i (\frac{SupB_i}{AvgLB_i} \times W_i)} \quad (4)$$

- where
- $SupB_i$ is the support of sequence Ω in the *benchmark data i*,
- $AvgLB_i$ is the weighted average length of sequences in the *benchmark data i*,
- W_i is the weight for the *benchmark data i*, and
- m is the number of *benchmark days*.

EI is negatively related to the supports in the *benchmark data*. The lower support the pattern has in the *benchmark data*, the more exceptional the pattern is. It reflects how exceptional the pattern is in the *targeted day* compared with in the *benchmark days*.

Consequently, a sequence is a local exceptional pattern, if it satisfies the conditions: (1) $II \geq MinII$, and (2) $EI \geq MinEI$, where $MinII$ and $MinEI$ are the thresholds given by users or domain experts.

To discover the local exceptional patterns, we design an algorithm, namely *Mining Local Exceptional Patterns*, as shown in the Fig. 2.

ALGORITHM: *Mining Local Exceptional Patterns*

INPUT: trading dataset TD , order dataset OD , m , γ , $MinII$, $MinEP$
OUTPUT: local exceptional patterns LEP

```

 $LEP = \emptyset$ ; /*local exceptional patterns*/
 $BS = \emptyset$ ; /*benchmark sequences*/
FOR each trading day  $i$  from trading day 1 to trading day  $m$ 
     $S = \text{GenSeq}(TD_i, OD_i)$ ; /*generate sequences*/
     $BS = BS + S$ ; /*add the sequences to benchmark sequences*/
ENDFOR
FOR each trading day  $i$  from trading day  $m + 1$  to the last trading day
     $S = \text{GenSeq}(TD_i, OD_i)$ ; /*generate sequences from targeted data*/
     $P = \text{MinePatterns}(S)$  /*mine patterns from the sequences*/
    FOR each pattern  $P_j$  in  $P$ 
         $II_j = \text{GetII}(P_j)$ ; /* quantify the intentional interestingness*/
         $EI_j = \text{GetEI}(P_j, BS, \gamma)$ ; /*quantify the exceptional interestingness*/
        /*add the pattern into  $LEP$ , if it meets the conditions*/
        IF  $II_j \geq MinII$  and  $EI_j \geq MinEI$ 
             $LEP = LEP + P_j$ ;
        ENDIF
    ENDFOR
    Replace the sequences generated from trading day  $i - m$  in  $BS$  with  $S$ ;
ENDFOR
OUTPUT local exceptional patterns  $LEP$ ;
```

Fig. 2. Algorithm Mining Local Exceptional Patterns

6 Experiments

Our approach has been tested on a real dataset from an Exchange. It covers 240 trading days from 2005 to 2006 for a security. There were 213,898 orders entered by traders during this period. These orders led to 228,186 trades.

Table 1 shows some samples of local exceptional patterns discovered by our approach. These patterns reflect the traders' exceptional intentions in the corresponding day. For example, in 24/05/2005, the *intentional interestingness* and *exceptional interestingness* for pattern $\{(S, M, S, O, C), (S, M, S, O, C)\}$ are 0.054 and 11.2 respectively, which means that this pattern indicates a strong intention and exception of trading activities for that day.

Table 1. Local exceptional pattern samples ($m = 10$, $\gamma = 0.01$, $MinII = 0.025$, and $MinEI = 5$); AR stands for the security's *abnormal return* compared with *market return*

Date	Local Exceptional Patterns	II	EI	$ Return \%$	$ AR \%$
05/01/2005	$\{(B,H,S,N,D), (B,H,S,N,D), (B,H,S,N,D), (B,H,S,N,O)\}$	0.026	$+\infty$	1.68	0.77
14/01/2005	$\{(B,H,S,N,D), (B,H,S,N,D), (B,H,S,O,C)\}$	0.025	7.6	2.68	1.38
18/01/2005	$\{(S,M,S,N,D), (S,M,S,N,D), (S,M,S,N,D), (S,M,S,O,C)\}$	0.026	10.8	2.25	1.85
20/01/2005	$\{(S,M,S,N,D), (S,H,S,O,C)\}$	0.038	5.4	2.98	1.56
28/01/2005	$\{(B,H,S,O,C), (B,M,S,O,C)\}$	0.030	8.2	2.63	1.97
01/02/2005	$\{(B,H,S,N,D), (B,H,S,N,D), (B,H,S,N,D)\}$	0.030	5.2	0.79	0.93
13/05/2005	$\{(S,M,S,N,D), (S,M,S,N,D), (S,M,S,N,O)\}$	0.026	5.1	0.95	2.24
24/05/2005	$\{(S,M,S,O,C), (S,M,S,O,C)\}$	0.054	11.2	6.82	6.38
16/06/2005	$\{(B,M,S,O,C), (S,M,S,N,O)\}$	0.025	6.1	2.64	2.47
17/06/2005	$\{(S,H,S,N,O)\}$	0.033	19.0	1.88	1.00
01/07/2005	$\{(B,H,S,N,D), (B,H,S,N,D), (B,H,S,O,C)\}$	0.028	54.0	2.21	1.28
13/07/2005	$\{(B,M,S,N,O)\}$	0.028	5.6	9.55	9.12
15/09/2005	$\{(B,H,S,N,O), (B,H,S,N,O)\}$	0.035	8.8	1.43	3.49
05/12/2005	$\{(S,M,S,O,C), (S,M,S,O,C)\}$	0.035	8.6	1.85	3.41

Figs 3 and 4 illustrate the performance of our approach under different thresholds of $MinII$ and $MinEI$.

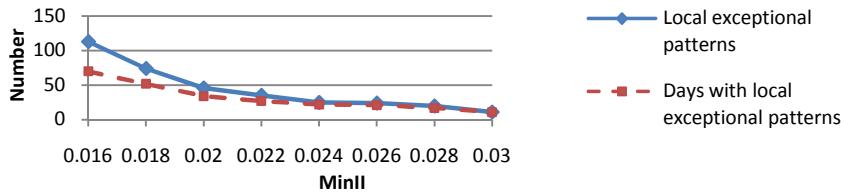


Fig. 3. Number of local exceptional patterns and number of days with local exceptional patterns for different $MinII$ when $m = 10$, $\gamma = 0.01$ and $MinEI = 5$

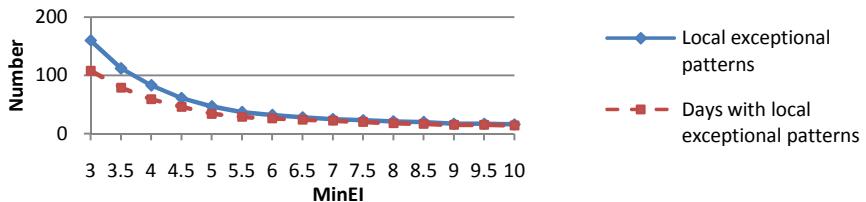


Fig. 4. Number of local exceptional patterns and number of days with local exceptional patterns for different $MinEI$ when $m = 10$, $\gamma = 0.01$ and $MinII = 0.02$

To evaluate the actionability of our findings in the business world, we calculate the absolute *return* and *abnormal return* of the security. In the stock market, *return* refers to the gain or loss for a single security over a specific period while *abnormal return*

indicates the difference between the *return* of a security and the *market return*. As shown in Table 1, the absolute *return* and *abnormal return* on 24/05/2005 are as high as 6.82%, 6.38% respectively, which are aligned with the values of *II* and *EI* for pattern $\{(S, M, S, O, C), (S, M, S, O, C)\}$. These results from both technical and business sides present business people strong indicators showing that there likely is price manipulation on that day.

7 Conclusion and Future Work

Often the outputs of traditional data mining cannot support people to make decision or take actions in the business world. There is a big gap between academia and business. However, this gap can be filled in by domain-driven data mining (D^3M) which generally targets the actionable knowledge discovery under domain-specific circumstances. In D^3M , the domain-specific characteristics are considered and domain knowledge is also encouraged to involve itself in the whole data mining process. Consequently the D^3M -based mining results have potential to satisfy the business expectations.

In this paper, we have investigated local exceptional trading patterns for detecting stock price manipulations in real-life orderbook data by utilizing D^3M . The main contributions of this paper are as follows: (1) studying exceptional trading patterns on real-life intraday stock data that is rarely studied in current literature, (2) proposing an effective approach to represent and construct trading orders, (3) developing an effective interestingness and algorithms for mining and evaluating local exceptional patterns, and (4) testing the proposed approach in real-life data by considering market scenarios and business expectations.

In the future, we plan to improve the representation of orders by including more domain-specific characteristics. Besides, we also expect to develop more interestingness measures reflecting the business concern.

Acknowledgement

The project is partially supported by Australian Research Council Discovery grants DP0773412, LP0775041 and DP0667060.

References

1. <http://www.marketsurveillance.org/>
2. Cao, L., Zhang, C.: Domain-driven Data Mining: A Practical Methodology. *Int'l J. Data WareHousing and Mining* 2(4), 191–196 (2006)
3. Cao, L., Yu, P.S., Zhang, C., Zhang, Y., Williams, G.: DDDM 2007: Domain Driven Data Mining. *ACM SIGKDD Explorations* 9(2), 84–86 (2007)
4. Cao, L.: Domain-Driven actionable knowledge discovery. *IEEE Intelligent Systems* 22(4), 78–89 (2007)
5. Cao, L., Luo, D., Zhang, C.: Knowledge actionability: satisfying technical and business interestingness. *Int. J. Business Intelligence and Data Mining* 2(4), 496–514 (2007)

6. Cao, L., Luo, C., Zhang, C.: Developing Actionable Trading Strategies for Trading Agents. In: IAT 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 72–75 (2007)
7. Ankerst, M.: Human Involvement and Interactivity of the Next Generation's Data Mining Tools. In: Workshop on Research Issues in Data Mining and Knowledge Discovery joint with DMKD 2001, Santa Barbara, CA (2001)
8. Aggarwal, C.: Towards Effective and Interpretable Data Mining by Visual Interaction. ACM SIGKDD Exploration Newsletter 3(2), 11–22 (2002)
9. Cao, L., Zhang, C.: The Evolution of KDD: Towards Domain-driven Data Mining. *Int. J. Pattern Recognition and Artificial Intelligence* 21(4), 667–692 (2007)
10. Freitas, A.A.: On Objective Measures of Rule Surprisingness. In: Zytkow, J., Quafafou, M. (eds.) PKDD 1998, vol. 1510, pp. 1–9. Springer, Heidelberg (1998)
11. Hilderman, R.J., Hamilton, H.J.: Applying Objective Interestingness Measures in Data Mining Systems. In: Zighed, D.A., Komorowski, J., Źytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 432–439. Springer, Heidelberg (2000)
12. Liu, B., Hsu, W., Chen, S., Ma, Y.: Analyzing Subjective Interestingness of Association Rules. *IEEE Intelligent Systems* 15(5), 47–55 (2000)
13. Padmanabhan, B., Tuzhilin, A.: Unexpectedness as A Measure of Interestingness in Knowledge Discovery. *Decision and Support Systems* 27, 303–318 (1999)
14. Silberschatz, A., Tuzhilin, A.: On Subjective Measures of Interestingness in Knowledge Discovery. *Knowledge Discovery and Data mining*, 275–281 (1995)
15. Yang, Q., Yin, J., Lin, C., Chen, T.: Postprocessing Decision Trees to Extract Actionable Knowledge. In: Proc. ICDM 2003, pp. 685–688. IEEE Computer Science Press, Los Alamitos (2003)
16. Ling, C., Sheng, W., Bruckhaus, T., Madavji, N.: Maximum Profit Mining and Its Application in Software Development. In: Proc. SIGKDD 2006, pp. 929–934. ACM Press, New York (2006)
17. Wang, K., Jiang, Y., Tuzhilin, A.: Mining Actionable Patterns by Role Models. In: ICDE 2006, p. 16. IEEE Computer Science Press, Los Alamitos (2006)
18. Tuzhilin, A.: Knowledge Evaluation: Other Evaluations: Usefulness, Novelty, and Integration of Interesting News Measures. In: *Handbook of Data Mining and Knowledge Discovery*, pp. 496–508 (2002)
19. Cao, L., Ou, Y.: Market Microstructure Patterns Powering Trading and Surveillance Agents. *Journal of Universal Computer Sciences* (to appear, 2008)

A Graph-Based Method for Combining Collaborative and Content-Based Filtering

Nguyen Duy Phuong, Le Quang Thang, and Tu Minh Phuong

Faculty of Information Technology,

Posts and Telecommunications Institute of Technology, Hanoi, Vietnam

phuong.ptit@yahoo.com, leqthang@gmail.com, phuong.tu@gmail.com

Abstract. Collaborative filtering and content-based filtering are two main approaches to make recommendations in recommender systems. While each approach has its own strengths and weaknesses, combining the two approaches can improve recommendation accuracy. In this paper, we present a graph-based method that allows combining content information and rating information in a natural way. The proposed method uses user ratings and content descriptions to infer user-content links, and then provides recommendations by exploiting these new links in combination with user-item links. We present experimental results showing that the proposed method performs better than a pure collaborative filtering, a pure content-based filtering, and a hybrid method.

Keywords: Collaborative filtering, content-based filtering, hybrid recommender systems, graph-based model.

1 Introduction

Recommender systems help users search for favorite products (such as movies, books, news stories etc.) by providing personalized suggestions in form of list of products that are likely to interest the users. These systems have played an important role in E-commerce and information filtering with a number of commercial systems deployed, examples include Amazon, Netflix, IMDB.

Collaborative filtering (CF) and *content-based filtering* (CBF) are the two main techniques used in recommender systems. CF systems work by first collecting user preferences for items in a given domain. The systems then use the collected data to find users with similar profiles and use their ratings to predict items that might interest a specific user [18,19]. CBF is an alternative technique that originates from the field of information retrieval. Content-based systems rely on the content descriptions of items (such as title, author, text description) to find items similar to items that interest the user. CBF has been mainly used in domains where content descriptions are available [3,14,1].

Both CF and CBF have their strengths and weaknesses. A key advantage of CF over CBF is that the former can perform in domains where it is difficult to get descriptions of items' content, for example where items to recommend are ideas, opinions etc. This makes CF the most successfully recommendation method in various

domains. On the other hand, CF relies only on user ratings to produce recommendations. In practice, most users rate very few items and the user-item rating data are typically very sparse. Therefore it is difficult to reliably compare the profiles of two users. This problem is widely known as the *data sparsity* problem and presents a major challenge to CF algorithms. Another difficulty is that CF methods cannot handle an item if no user has rated it before. This problem, known as the *first rater* problem, applies to new and obscure items. Such problems are easily solved by CBF, which can make recommendations by comparing the descriptions of item content.

In this paper, we propose to combine content information with rating information by using a unified graph representation for the two types of information. From the combined graph model representing user-item and item-content interactions, our method first determines content features that have significant impact on the behavior of each user. A network-propagation algorithm is then used to compute the association between user node and item nodes by exploring both user-content and user-item links of the graph. We apply the method in the domain of movie recommendation and show that our method gives promising results.

Related Work

The potential benefits of combining collaborative and content-based filtering have been studied in a number of works. The most simple *hybrid* approach is to implement content-based and collaborative methods separately and then combine their predictions [7]. In another approach, content information and rating information are first combined to produce data that serve as mixed input for predictors. Pazzani [16] proposed to represent each user-profile by a vector of weighted words selected from content descriptions using the Winnow algorithm. The matrix of user-profiles is then used as input for collaborative filtering instead of the user-item matrix. The Fab system [3] employs content analysis to generate user profiles from relevance feedbacks, which are then used to create personal filters. Melville and Mooney [13] used a pure content-based predictor to calculate so called pseudo-ratings. They used the predicted ratings to augment user ratings vector before applying CF techniques.

Another family of approaches creates a general unified recommendation model and treats user rating prediction as a machine learning problem, in which predictors are learned from labeled examples [5,15]. Popescul *et al.* [17] proposed a unified probabilistic latent semantic analysis that combines collaborative and content-based characteristics. The approach of [4] uses kernel functions to combine user-user and item-item similarities in a unified kernel vector space and applies support vector learning to produce predictions. Crammer *et al.* [8] approaches the problem as learning a ranking on items set by incorporating additional item features.

Due to intuitiveness of representation and availability of graph algorithms, graph-based models have been used in a number of recommendation methods. Aggarwal *et al.* [2] represented relationships among users as a directed graph in which a directed link connecting two users indicates that the behavior of the source user is highly predictive from the behavior of the target user. Huang *et al.* [11] introduced a graph-based model that includes both users and items. A weighted link between two item nodes represents the similarity between the two items, which is pre-computed based on the items' content. This model allows capturing both content-based and rating information in a unified framework. In [12], the authors exploited transitive associations in a graph-based model to tackle the data sparsity problem.

2 Graph Model and Recommendation Algorithm

We first introduce notations to use in the paper. We denote by $X = \{x_1, x_2, \dots, x_{|X|}\}$ a set of items, and by $U = \{u_1, u_2, \dots, u_{|U|}\}$ a set of users. We denote user ratings over the items by matrix $R = (r_{ij})$ of size $|U| \times |X|$, such that r_{ij} is the rating user i has given to item j . Each rating r_{ij} can take on a value from a finite set of possible ratings. Here we assume r_{ij} can be either +1 (like) or -1 (dislike). If user i has not rated j then $r_{ij} = \emptyset$. Furthermore, we use $C = \{c_1, c_2, \dots, c_{|C|}\}$ to denote a set of features that characterize the items' content. We denote the item-content associations by a $|X| \times |C|$ matrix $Y = (y_{ij})$, where $y_{ij} = 1$ if item i has feature j and $y_{ij} = 0$ otherwise. For example, for x_i being a movie, c_j can be “genre = action” and $y_{ij} = 1$ means the movie belongs to genre “action”. The goal of a recommender system is to predict ratings an active user would give to unrated items and based on this provide a list of recommendations.

The Graph Model. As shown in previous works [2,11], it is natural and convenient to solve the problem at hand by using a graph-based recommendation model. The basic idea is to build a graph model of the rating and content information, and then explore the associations among nodes to make predictions. Figure 1 shows an example graph. The top part of the graph shows item-content associations, where a node corresponds to either an item or a content feature. A link is drawn between an item node x_i and a feature node c_j if there is a non-zero association between x_i and c_j , i.e. if $y_{ij} = 1$ according to the notation above. Similarly, the bottom part of the graph represents user preferences over items. In this part, a link between a user node and an item node can have +1 or -1 weights indicating the user likes or dislikes the item.

2.1 Construction of User-Content Links

Given the item-content association matrix, a simple way to compute the similarity between two items is to compare their content features. For example, Huang *et al.* [11] compute the similarity of two items by calculating the mutual information between the items' descriptions and then draw weighted links between the item nodes to represent their similarity. However, such methods do not take into account user ratings when computing item-item similarities and thus cannot adjust item similarity for a specific user. To illustrate this, let us consider the example given in figure 1.

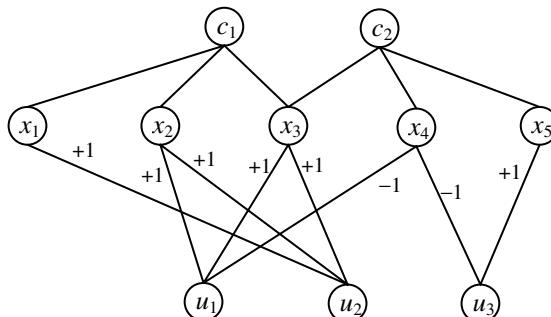


Fig. 1. An example of graph representation for rating and content information

In this example, item x_3 and x_4 share a common feature c_2 . Looking only at the item-content part of the graph, a simple similarity computation will decide that x_3 and x_4 are similar because of this common feature. But this is not true for user u_1 , who has rated x_3 as liked and x_4 as disliked. This means, from u_1 's point of view, x_3 and x_4 are not similar and c_2 has no effect on u_1 's ratings. At the same time, all the items which u_1 has rated and which contain c_1 (i.e. x_2 and x_3) get positive ratings from u_1 . Therefore, one should consider c_1 as having an important role in the opinion of u_1 .

This example shows the need to use a personalized measure of similarity when computing item-item similarities from content information. Each content feature should have its degree of importance in deciding how items are similar, and this degree of importance should be adapted for a specific user. We now present our approach that characterizes the importance of each content feature for each user. This is the first step to combine content-based and collaborative recommendations.

Given a graph introduced above, for each user u_i and each content feature c_k , we say that c_k is important for u_i if the sum of the weights of all distinct paths connecting c_k and u_i divided by the number of the paths exceeds some threshold T ($0 < T < 1$). Here, only paths of length 2 and go through item nodes are considered. For a path that connects u_i and c_k via x_j , the weight is computed as $r_{ij}^*y_{jk} = r_{ij}$, since $y_{jk} = 1$.

Let s_{ik} be the number of paths that connect u_i and c_k , and w_{ik} be the sum of the path weights. Because a path weight can be either +1 or -1, w_{ik} is equal to the number of paths with positive weights minus the number of paths with negative weights. We define the degree of importance v_{ik} of c_k with respect to u_i as:

$$v_{ik} = \begin{cases} \frac{\min(s_{ik}, \gamma)}{\gamma} \frac{w_{ik}}{s_{ik}} & \text{if } \frac{w_{ik}}{s_{ik}} > T \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In this formula, w_{ik}/s_{ik} shows the dominance of positive ratings over negative ratings that user u_i gives for items with feature c_k ; $\min(s_{ik}, \gamma)/\gamma$ is the so called *significance weighting factor* [9] that devalues the importance degree based on few paths. Following [9] we used $\gamma = 50$ in our experiments. The threshold valued T is set to 0.3, which means the number of positive paths should be about two times bigger than the number of negative ones for a feature to be considered important.

We illustrate the computation of v_{ij} through an example shown in figure 1: for user u_1 and content feature c_1 , we have $n_{11} = 2$, $w_{11} = 2$, and $v_{11} = 2/\gamma$. For user u_1 and c_2 , we have $n_{12} = 2$ and thus $w_{12} = 0$.

For each pair (u_i, c_k) that has a non-zero v_{ik} , we draw a new link with weight v_{ik} . Figure 2 shows such an extended graph for the graph from figure 1. The dotted line is the new link just added to represent the correlation between u_1 and c_1 .

2.2 Making Recommendation

We now describe the recommendation process as a graph search problem in the extended graph. We will use the example shown in figure 2 to illustrate our approach.

Suppose the system needs to recommend items for an active user. Following [11, 12], we first determine the association between this user and each of items that have not been rated by the user. The items are then sorted according to the associations, and top K items are chosen for recommendation.

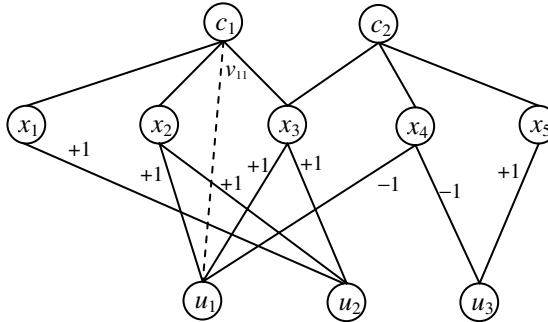


Fig. 2. Example of graph with links between user nodes and content feature nodes

In our model, the association between two nodes is determined by considering all paths connecting them. For the pair of a user node u_i and an item node x_j , we compute the association between them as the sum of weights of all distinct paths that connect u_i and x_j . In this computation we differentiate two types of paths – *paths via content nodes* and *paths via item nodes*. A path of the *first type* is one whose length is equal to 2 and goes through a content feature node. An example of such a path is $u_1-c_1-x_1$ in figure 2, which has an intuitive interpretation – u_1 likes items that contain feature c_1 and therefore likes x_1 . Such paths correspond to associations via content information. By limiting the path length to be 2, we do not allow transitive associations for paths of this type. Since the weight of an item-content link is always 1, the weight of a path of this type is equal to the weight of the respective user-content link.

The *second type* includes paths that go through item nodes and user nodes. Examples of such paths are $u_1-x_3-u_2-x_1$, $u_2-x_2-u_1-x_4-u_3-x_5$. Because we are interested in the association between a user node and an item node, the length of a path of this type must be an odd number. In addition, only paths whose lengths do not exceed a parameter M are considered. Such paths represent transitive associations that were first exploited by Huang *et al.* [12] to make recommendations.

In contrast to the work of Huang *et al.*, who applied collaborative filtering for purchasing data that contain only positive links, in our context, a user-item link can have either a positive or a negative weight. Thus when exploring paths that go through item nodes we consider three cases:

- All the intermediate links of a path are positive (here we define intermediate links of a path as all its links except the last one), for example paths $u_1-x_3-u_2-x_1$, and $u_2-x_2-u_1-x_4$ in figure 2. We consider such a path important and compute its weight as the product of the link weights.
- A path has two intermediate links that end at the same item node but have different signs. This means the two corresponding users have rated this item differently and thus we ignore such a path.
- A path has two consecutive negative links that end at the same item, for example in path $u_1-x_4-u_3-x_5$, links u_1-x_4 and x_4-u_3 both have negative weights. This means the two users have similar ratings for x_4 and this path may indicate the similarity between the two users. However, our experiments show that including such paths in computation makes results unstable and hence we ignore such paths in our implementation.

Association computation. An important step in the algorithm above is computing the association via different paths. This can be solved by a variety of algorithms known as *network propagation* algorithms, an example of which is the Google PageRank algorithm. Huang *et al.* [12] applied several network propagation algorithms to CF. Here, to compute user-item associations, we use the following algorithm which is a modification of the algorithm by Weston *et al.* [22].

Let u_a be the active user, i.e. the user for which the system needs to make recommendations. Let N denote the set of nodes that can form paths from u_a to items nodes. For paths of the first type, $N = C \cup X \cup u_a$. For paths of the second type, $N = X \cup U$. For ease of algorithm description we use n_i to denote a node $i \in N$ regardless this is a user node, content node or an item node. We denote by e_{ij} the weight of the link between node n_i and n_j , and by e_{aj} the link weight between node u_a and n_j . The matrix (e_{ij}) is formed from matrices R , Y introduced in section 3 and matrix (v_{ij}) from equation (1). Furthermore, let $a_i(t)$ denote the association degree between u_a and node $n_i \in N$ when considering paths of length t . The algorithm for computing association between u_a and item nodes using only paths of the same type is shown in figure 3.

```

1. Initialize  $a_i(0) = 0$  for all  $n_i \in N$ ,  $a_a(0) = 1$ 
2. for  $t = 1, 2, \dots, m$  or until convergence do
3.   for each node  $n_i \in N$  do
4.      $a_i(t) \leftarrow e_{ai}$ 
5.     for each node  $n_j \in N$  do
6.       if  $e_{ij} > 0$  or  $t = m$  then
7.          $a_i(t) \leftarrow a_i(t) + \alpha e_{ji} a_j(t-1)$ 
8.       end for
9.     end for
10.   end for
11.   return  $a_i(m)$ .  $a_i(m)$  is the association between  $u_a$  and  $n_i$  via
      paths of length  $m$ 
```

Fig. 3. Algorithm for computing association degrees

Here, $\alpha \in [0,1]$ is a parameter that down-weights longer paths. Our experiments use $\alpha=1$ for paths of the first type and $\alpha=0.5$ for paths of the second type.

The algorithm needs to be run separately for $N = C \cup X \cup u_a$, and $N = X \cup U$ to compute associations via content nodes and via item nodes respectively. Parameter m is equal to 2 (path length = 2) for the former case and is an odd number less than or equal to M for the later case. If no limit is set, the algorithm runs until convergence, that is when $a_i(t)$ become stable. In our implementation, we use $M = 10$.

In the initialization stage, the algorithm activates the active user node by setting its activation level to 1 and the activation levels of the remaining nodes to 0. In each iteration t , the activation level is pumped from the active user node to the remaining nodes of N via paths of length t . Step 6 ensures that a path is counted only if it does not contain a negative link(s) unless this is the last link. Clearly, this step is needed only for paths of the second type.

The most time consuming part of this algorithm is from line 3 to line 9 which requires $O(N^2)$ computations over all e_{ji} . Fortunately, matrix (e_{ij}) is very sparse with most elements equal to zero. This allows us to use sparse-matrix representation for (e_{ij}) , which reduces the complexity to $O(|N|S)$, where S is averaged number of non-zero elements for each row of matrix (e_{ij}) .

Assume the algorithm returns a_i^c and a_i^r for $N = C \cup X$, and $N = X \cup U$ respectively, we compute the overall association a_i^o between user u_a and item x_i as:

$$a_i^o = \beta a_i^c + (1 - \beta) a_i^r \quad (2)$$

Where $\beta \in [0, 1]$ is a parameter that controls the contribution of each type of association. $\beta = 1$ means only association via content is considered while $\beta = 0$ corresponds to pure collaborative filtering.

The unrated items are then sorted based on their associations and top K items with strongest associations are recommended for the active user.

3 Experimental evaluation

Experimental setup. We evaluated the proposed algorithm on the MovieLens data set (<http://www.grouplens.org>). The data set contains 100000 ratings from 943 users for 1682 movies. Ratings are in five-point scale (1, 2, 3, 4, 5) and each user has rated at least 20 movies. We transformed the two highest scores (4 and 5) into +1 (like) and the rest into -1 (dislike). 80 percent of the users were randomly selected to form the training set and the rest were used as the test users. Users in the test set were used to measure recommendation accuracy. From each test user, 25 percent of the ratings were withheld. The rest of ratings were used as input for recommendations.

We used movie genre provided with the MovieLens data set and retrieved other information about the movies from IMDB (<http://www.imdb.com>) to form content features. In our experiments we used only genre and director as content information. In general, other content features can be exploited.

Following experimental procedures reported in the literature [10, 12] we used *precision*, *recall*, and *F-measure* to measure the effectiveness of recommendation methods. The metrics are defined as follows:

$$\text{precision} = \frac{\# \text{ of recommended items that get actual positive ratings}}{\# \text{ of all recommended items}} \quad (3)$$

$$\text{recall} = \frac{\# \text{ of recommended items that get actual positive ratings}}{\# \text{ of all items that get actual positive ratings}} \quad (4)$$

$$F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (5)$$

We compared the proposed method (denoted by *CombinedGraph*) to the following methods:

- *User-Based* k-nearest neighbor using Pearson correlation. This method uses the Pearson correlation as the measure of similarity between two users, and makes recommendations based on the ratings of users that are highly similar with the active user.
- *Content-Based* using graph search. This method counts the number of paths of length 3 that go from the active user node via item and content nodes to an un-rated item node and recommends items with the largest number of paths.
- *3-Hop* collaborative filtering using graph search. This method makes recommendations based only on the association via item nodes as described in the previous section with the maximal path length $M = 3$.
- *Simple Hybrid* method that combines Content-Based and 3-Hop by merging top recommended items returned by each method.

Results. We varied parameter β to control the contribution of the content component and the collaborative component of the method. The best results were obtained for β between 0.7 and 0.8, which emphasizes the contribution of the collaborative filtering component. In what follows we report only results when using $\beta = 0.8$.

The recalls, precisions, and F-measure values for top 10, 20, and 50 items are summarized in table 1. The results show that our method performs better than the other methods on all three metrics. On average, among 50 items recommended by the CombinedGraph, eight will receive positive ratings from the users.

The table also shows that simple 3-Hop significantly outperforms both User-Based and Content-Based methods. This result supports recent findings that user-based k-NN approach gives poor results in terms of precision and recalls while performs well in term of the *mean absolute error* (MAE) metric - defined as the average absolute difference between predicted ratings and actual ratings [10].

Table 1. Recalls, precisions, and F-measure values of experimented methods

Algorithm	Metrics	Number of recommended items		
		10	20	50
User-Based	Recall	0.007	0.021	0.069
	Precision	0.015	0.025	0.034
	F-measure	0.009	0.023	0.045
Content-Based	Recall	0.009	0.017	0.037
	Precision	0.022	0.020	0.018
	F-measure	0.013	0.018	0.024
3-Hop	Recall	0.155	0.222	0.377
	Precision	0.284	0.225	0.164
	F-measure	0.200	0.223	0.228
Simple Hybrid	Recall	0.117	0.162	0.279
	Precision	0.186	0.148	0.118
	F-measure	0.144	0.155	0.166
CombinedGraph	Recall	0.165	0.234	0.381
	Precision	0.292	0.240	0.175
	F-measure	0.211	0.237	0.240

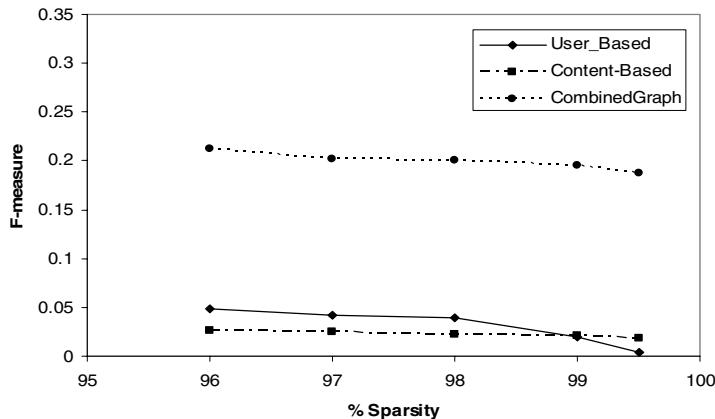


Fig. 4. F-measure values at different sparsity levels

One challenge for CF algorithms is that the recommendation accuracy suffers when the user-item matrix is sparse. Our method can alleviate this problem by exploiting association via content feature nodes even when most paths via item nodes are not present. Thus the sparsity of rating data has a smaller effect on CombinedGraph than on a pure collaborative algorithm. To verify this hypothesis, we conducted the following experiments. We used the first 400 users to form the training set and the next 100 user to form the test set. For each test user, 25% ratings were withheld for prediction. We randomly deleted elements from the user-item matrix to increase the sparsity and measured the accuracy of recommendations at different sparsity levels. The F-measure values for top 50 recommendations are shown in figure 4. As can be seen, CombinedGraph gives more stable results than User-Based and 3-Hop when the sparsity increases. This confirms our hypothesis that CombinedGraph is less sensitive to data sparsity than pure CF.

4 Conclusion

We have presented a natural and effective way to combine content-based and collaborative filtering for achieving more accurate recommendations. Our method uses a graph-based model to represent both content and rating information. This representation allows exploiting user ratings to select important content features that connect users with items of interest. The graph model also provides a convenient way to compute the association between users and items using available network propagation algorithms. We have shown how our method performs better than the user-based k-NN collaborative filtering method, a content-based method and a hybrid recommendation method.

Acknowledgments. This research was supported by Ministry of Science and Technology of Vietnam under a grant for fundamental research.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Trans. Knowl. and Data Eng.* 17(6) (2005)
2. Aggarwal, C.C., Wolf, J.L., Wu, K.L., Yu, P.S.: Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. In: Proc. of ACM SIGKDD 1999 (1999)
3. Balabanovic, M., Shoham, Y.: FAB: Content-based, collaborative recommendation. *Communication of the ACM* 40(3), 66–72 (1997)
4. Balisico, J., Hofman, T.: Unifying collaborative and content-based filtering. In: Proceedings. of Int. Conf. on Machine learning (ICML 2004) (2004)
5. Basu, C., Hirsh, H., Cohen, W.: Recommendation as classification: Using social and content-based information in recommendation. In: Proc. Nat. Conf. on AI, pp. 714–720 (1998)
6. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proc. of 14th Conf. on Uncertainty in AI, pp. 43–52 (1998)
7. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining contentbased and collaborative filters in an online newspaper. In: Proc. of ACM SIGIR Workshop on Recommender Systems (1999)
8. Crammer, K., Singer, Y.: Pranking with ranking. In: Advances in Neural Information Processing Systems, vol. 14, pp. 641–647 (2002)
9. Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: SIGIR 1999: Proc. of the 22nd Inter. Conf. on Research and Development in Information Retrieval (SIGIR), pp. 230–237 (1999)
10. Herlocker, J., Konstan, K., Terveen, L.G., Riedl, J.: Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. on Inform. Syst.* 22(1), 5–53 (2004)
11. Huang, Z., Chung, W., Ong, T.-H., Chen, H.: A graph-based recommender system for digital library. In: Proc. of 2nd ACM/IEEE-CS Joint Conf. on Digital Libraries, pp. 65–73 (2002)
12. Huang, Z., Chen, H., Zeng, D.: Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering. *ACM Trans. Inf. Syst.* 22(1), 116–142 (2004)
13. Melville, P., Mooney, R.J., Nagarajan, R.: Content-boosted collaborative filtering for improved recommendations. In: Proc. of 18th Nat. Conf. on AI, pp. 187–192 (2002)
14. Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: Proc. of the 5th ACM Conference on Digital Libraries, pp. 195–204 (2000)
15. Phuong, N.D., Phuong, T.M.: Collaborative filtering by multitask learning. In: Proc. IEEE Int. Conf. on Research, Innovation and Vision for the Future, pp. 227–232 (2008)
16. Pazzani, M.J.: A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review* 13(5-6), 393–408 (1999)
17. Popescul, A., Ungar, L.H., Pennock, D.M., Lawrence, S.: Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. In: Proc. 17th Conf. Uncertainty in Artificial Intelligence (2001)
18. Resnick, P., Varian, H.R.: Recommender systems. Special issue of Communications of the ACM, 56–58 (1997)
19. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating word of mouth. In: Human Factors in Computing Systems ACM CHI, pp. 210–217 (1995)
20. Wang, J., de Vries, A.P., Reinders, M.J.T.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proc. of SIGIR 2006, Seattle, USA (2006)

21. Yu, K., Schwaighofer, A., Tresp, V., Ma, W.-Y., Zhang, H.: Collaborative ensemble learning: Combining collaborative and content-based information filtering via hierarchical bayes. In: Proc. of the 19th Conference on Uncertainty in Artificial Intelligence, UAI (2003)
22. Weston, J., Elisseeff, A., Zhou, D., Leslie, C.S., Noble, W.S.: Protein ranking: From local to global structure in the protein similarity network. Proc. of National Academy of Science 101(17), 6559–6563 (2004)

Hierarchical Differential Evolution for Parameter Estimation in Chemical Kinetics

Yuan Shi¹ and Xing Zhong²

¹ School of software, Sun Yat-sen University, Guangzhou 510006, P.R. China

² Department of chemistry and chemical engineering, Nanjing University

Nanjing 210093, P.R. China

sycaszsu@gmail.com

Abstract. Parameter estimation, a key step in establishing the kinetic models, can be considered as a numerical optimization problem. Many optimization techniques including evolutionary algorithms have been applied to it, yet their efficiency needs further improvement. This paper proposes a hierarchical differential evolution (HDE) in which individuals are organized in a hierarchy and mutation base is selected based on the hierarchical structure. Additionally, the scaling factor of HDE is adjusted according to both the hierarchy and the search process, elaborately balancing the exploration and exploitation. To demonstrate the performance of HDE, experiments are carried out on kinetic models of two chemical reactions: pyrolysis and dehydrogenation of benzene as well as supercritical water oxidation. The results show that the proposed algorithm is an efficient and robust technique for kinetic parameter estimation.

Keywords: Differential evolution, hierarchy, parameter estimation, kinetic model.

1 Introduction

Chemical kinetics is the study of the characteristics of chemical reactions. Models for chemical kinetics are established on the basis of the predicting reaction mechanisms which are consistent with experimental data. Based on those models, we can deduce the rate constants, active energy and many other reaction parameters which are the foundation for designing reasonable reactors. Kinetic models are usually systems of nonlinear ordinary differential equations with several adjustable parameters [1]. A key step in establishing the models is the parameter estimation whose objective is to minimize the error between the experimental data and data calculated from the model. Finding the optimal parameters is crucial since exact parameter estimation can help to figure out the influence of each factor and ultimately play an important role in guiding the industrial manufacture.

Parameter estimation in chemical kinetics is actually a numerical optimization problem. Several conventional methods have been used as parameter estimation techniques, such as the graphical method [2] and the gradient-based non-linear optimization method [3]. The graphical method can only tackle those problems that can be converted to linear regression problems, while the gradient-based nonlinear optimization

method is easy to trap into local optima. Recent study on optimization technique has developed an attractive class of algorithms, namely evolutionary algorithms (EAs). They have been received increasing attention due to their powerful capability for global search. Also, they have the distinctive advantages such as implicit parallelism and information requirement only for the objective functions. Some popular EAs are genetic algorithm (GA) [4], ant colony optimization (ACO) [5], particle swarm optimization (PSO) [6], differential evolution (DE) [7] and so forth. EAs have been successfully applied to estimate the parameters for chemical kinetic models [8], [9] and showed their superiority to some conventional methods. However, there exist a number of complicated parameter estimation problems with high non-linearity; hence more efficient techniques are increasingly favored.

DE is a simple yet efficient population-based EA over continuous domain. The crucial idea behind DE is to use the differences between two individuals to disturb a third individual named mutation base. Several different variations of DE have been proposed [10] and two widely-used ones are denoted as DE/rand/1/bin and DE/best/1/bin. DE/rand/1/bin uses a random individual as the mutation base, which can maintain the population diversity but probably lower the search efficiency. While DE/best/1/bin, utilizing the best individual as the mutation base, gives the best individual a very high influence on the mutated population. This results in relatively high search efficiency but is likely to get stuck into local optima. Literature [10] also introduced another variation called DE/lbest/1/bin, which uses the best individual in a defined neighborhood as the mutation base, so that the population diversity won't lose quickly. It should be mentioned that the neighborhood strategy has been already investigated in the study of PSO. We can see that considerable efforts have been made to incorporate such strategy into PSO [11] [12] and many neighborhood structures have been developed. Literature [13] designed a hierarchical neighborhood structure for PSO and reported superior results.

In this paper, we propose a hierarchical differential evolution (HDE) which is inspired by the work in literature [13]. The population is arranged in a hierarchy and mutation base is selected based on the hierarchical structure. Individuals move up or down in the hierarchy, depending on their qualities. In addition, one of the parameter of the algorithm, the scaling factor, is set based on the hierarchy and adjusted adaptively during the search process. In the case study, HDE is applied to kinetic parameter estimation for two chemical reactions: pyrolysis and dehydrogenation of benzene as well as supercritical water oxidation. The results are compared with those obtained by DE and PSO, demonstrating the efficiency and robustness of HDE.

2 Differential Evolution

The population in DE is composed of PS individuals $x_{i,G}$ ($i = 1, 2, \dots, PS$) , where G denotes the current generation. Each individual $x_{i,G}$ is represented by a D dimensional vector: $x_{i,G} = (x_{i1,G}, x_{i2,G}, \dots, x_{iD,G})$. DE evolves the population by iteratively performing operations including mutation, crossover and selection.

(1) Mutation

For each individual $x_{i,G}$ ($i = 1, 2, \dots, PS$), the mutation operation first randomly selects three other individuals, denoted by $x_{r_1,G}, x_{r_2,G}$ and $x_{r_3,G}$, then a mutated individual is generated according to the following equation:

$$v_{i,G} = x_{r_1,G} + F \cdot (x_{r_2,G} - x_{r_3,G}) \quad (1)$$

where $i, r_1, r_2, r_3 \in [1, PS]$ and $i \neq r_1 \neq r_2 \neq r_3$. $x_{r_1,G}$ is called the mutation base, while F is a positive parameter named scaling factor which determines the amplification of the added differential variation of $(x_{r_2,G} - x_{r_3,G})$.

(2) Crossover

Crossover operation recombines $x_{i,G}$ and $v_{i,G}$ to yield a trail individual $u_{i,G+1}$. The operation is carried out following the equation below:

$$u_{ij,G+1} = \begin{cases} v_{ij,G}, & \text{if } r(j) < CR \text{ or } j = rn(i) \\ x_{ij,G}, & \text{otherwise} \end{cases} \quad (2)$$

with $j = 1, 2, \dots, D$. $r(j) \in [0, 1]$ is the j th evaluation of a uniform random number generator and $CR \in [0, 1]$ denotes the crossover rate. $rn(i) \in \{1, \dots, D\}$ is a randomly chosen index which ensures that $u_{i,G+1}$ gets at least one element from $v_{i,G}$.

(3) Selection

The selection operation allows the better one between $x_{i,G}$ and $u_{i,G+1}$ to survive according to the following rule, in which f stands for the objective function.

$$x_{i,G+1} = \begin{cases} u_{i,G+1}, & \text{if } f(u_{i,G+1}) < f(x_{i,G}) \\ x_{i,G}, & \text{otherwise} \end{cases} \quad (3)$$

3 Hierarchical Differential Evolution

3.1 Hierarchical Structure

In the proposed algorithm, individuals are arranged in a hierarchy which is represented by a (nearly) regular tree. All inner nodes in the hierarchy have the same number of child nodes, only the inner nodes on the deepest level of the tree might have a smaller number of child nodes. The hierarchical structure is defined by three parameters:

- (1) total number of nodes m .
- (2) height h – the number of levels of the tree. The level is counted from 0 to $h-1$.
- (3) branching degree d – the maximum number of child nodes of the inner nodes.

Fig. 1 shows a hierarchy with $m = 12$, $h = 3$ and $d = 3$.

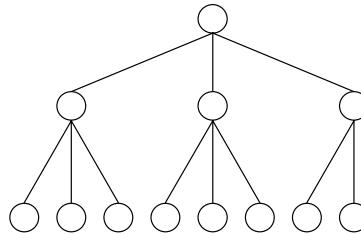


Fig. 1. A hierarchy defined by $m = 12$, $h = 3$ and $d = 3$

Each node of the hierarchy stands for an individual in the population. The positions of the individuals are updated every iteration, moving the better individuals gradually to the upper levels.

3.2 Structure of HDE

In the mutation operation of HDE, for each individual, mutation base is the individual on its parent node, only the individual on the root node uses itself as the mutation base. For each individual i ($i = 1, 2, \dots, NP$), the mutated individual is formed as follows:

$$v_{i,G} = x_{p(i),G} + F \cdot (x_{r_2,G} - x_{r_3,G}) \quad (4)$$

where $p(i)$ is the index of the individual on the parent node of the i th individual. r_2 , r_3 and G are the same as those in Section 2.

In addition, the scaling factor F is set according to the level on which the i th individual placed. Mathematically, F is determined according to the equation below:

$$F_k = F_0 + \alpha \cdot k \quad (5)$$

where F_k represents the value of the scaling factor on the k th level of the hierarchy, α is a real number between 0 and 1. Since the individuals in the upper levels tend to have better qualities, the strategy places smaller disturbance on better individuals to enhance the local search while adds larger disturbance on worse individuals to improve the global search. Furthermore, F_0 is adjusted adaptively during the search following the rule below:

$$F_0 = F_{0,\max} - F_{0,\min} \cdot \text{iter} / \text{MAXITER} \quad (6)$$

where $F_{0,\max}$ and $F_{0,\min}$ are the maximum and minimum values of F_0 , respectively. iter denotes the current iteration and MAXITER is the predefined maximum number of iterations. The adjustment of the scaling factor according to both the hierarchical structure and the search process can elaborately balance the global exploration and local exploitation of the algorithm. Our previous numerical experiments have indicated a desired combination of the above three parameters, that is: $F_{0,\max} = 0.6$, $F_{0,\min} = 0.4$ and $\alpha = 0.05$. This combination is also used in Section 4.

To give a better individual in the population higher influence, the positions of the individuals are updated after the selection operation each iteration. The position updating procedure is performed in the following way: for each individual i ($i = 1, 2, \dots, NP$), its objective function value is compared with that of individuals on its child nodes. If the best one among these individuals is better than the individual i , their positions swap.

Based on the above description, the flowchart of HDE is given in Fig. 2.

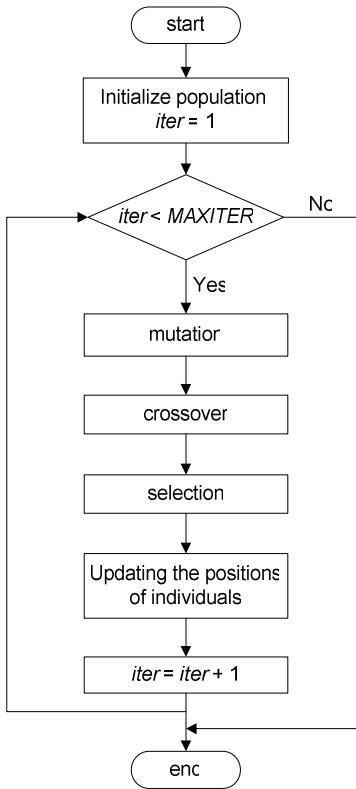


Fig. 2. Flowchart of HDE

4 Case Study

In this section, HDE is applied to two chemical kinetic models: pyrolysis and dehydrogenation of benzene as well as supercritical water oxidation. The performance of HDE is compared with that of DE (DE/rand/1/bin) and PSO. For either case, every algorithm is executed in 30 independent runs and the best, average and the worst value of the results are recorded. In both cases, the branching degree of the hierarchy in HDE is 3, the crossover rate CR in HDE and DE are set to 0.9 while F in DE is fixed as 0.5. The parameters in PSO are configured as follows: $w = 0.5$, $c_1 = c_2 = 2$.

4.1 Case 1: Pyrolysis and Dehydrogenation of Benzene

Dehydrogenation of benzene in vapor-phase at industrial level is mainly used to prepare Biphenyl that is an important industrial material mainly served as heat transfer agent, anti mildew agent and raw material of synthetic resins [14]. Derivatives of Biphenyl are widely used as monomers of various polymers, liquid crystal materials [15]. Paraterphenyl can also be obtained as by-products by this reaction. Under different conditions, the ratio of the biphenyl and paraterphenyl can vary over a certain range. They are obtained in the following reactions:



The proposed kinetic model is [16]:

$$\frac{dx_1}{dt} = -r_1 - r_2 \quad (9)$$

$$\frac{dx_2}{dt} = \frac{r_1}{2} - r_2 \quad (10)$$

$$r_1 = k_1[x_1^2 - x_2(2 - 2x_1 - x_2)/(3K_1)] \quad (11)$$

$$r_2 = k_2[x_1x_2 - (1 - x_1 - 2x_2)(2 - 2x_1 - x_2)/(9K_2)] \quad (12)$$

where x_1 and x_2 are the residual volume of benzene and biphenyl, respectively. k_1 and k_2 are the reaction rate constants that are to be estimated. $K_1 = 0.242$ and $K_2 = 0.428$ are the equilibrium constants.

Table 1. Experimental data of phrolysis and dehydrogenation of benzene

$t \times 10^4$	5.63	11.32	16.07	22.62	34	39.7	45.2	169.7
x_1	0.828	0.704	0.622	0.565	0.499	0.482	0.47	0.443
x_2	0.0737	0.113	0.1322	0.14	0.1468	0.1477	0.1477	0.1476

Table 1 shows the experimental sample data provided by the literature [16]. The initial condition is $t = 0$, $x_1 = 1$ and $x_2 = 0$.

The objective function is defined as:

$$f = \sum_{j=1}^n \sum_{i=1}^2 (x_{i,j} - x'_{i,j})^2 \quad (13)$$

where n is the sample size. $x_{i,j}$ is the experimental value of x_i in the j th sample while $x'_{i,j}$ is the value obtained by the kinetic model.

The population size of the three algorithms is 20 and the maximum number of iterations is 25. The optimization results are listed in Table 2, which shows that the best objective function value achieved by the three algorithms are the same while the average and worst values obtained by HDE are better than those gained by DE and PSO. The kinetic parameters found by HDE are: $x_1 = 358.342$ and $x_2 = 404.274$.

Table 2. Results obtained by HDE, DE and PSO in Case 1

Algorithm	HDE	DE	PSO
Best	0.0008372	0.0008372	0.0008372
Average	0.0008372	0.0008383	0.0008379
Worst	0.0008372	0.0008434	0.0008390

Fig. 3 illustrates the objective function value of the best individual in the population versus iteration, demonstrating that HDE converges much faster towards the global than DE and PSO.

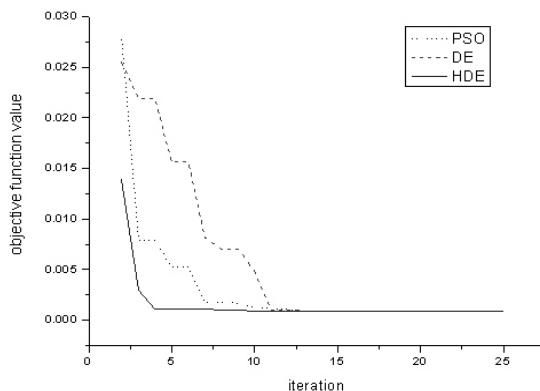


Fig. 3. Objective function value of the best individual versus iteration in Case 1

4.2 Case 2: Supercritical Water Oxidation

Supercritical water oxidation (SCWO) is an eco-environmental method which can be very effective in the disposal of organic waste. Using supercritical water as medium to decompose organic material by completely destroying the structure of the material, this method have many advantages such as high reaction rate, high removing rate of the waste (normally, it can reach the level of 99.9%) [17]. Literature [18] showed the removing rate expression for dichloropheno (2-cp), a common industrial organic wastewater, during SCWO:

$$r = A \exp\left(-\frac{E_a}{RT}\right) [2CP]^a [\text{O}_2]^b [\text{H}_2\text{O}]^c \quad (14)$$

where r is the removing rate of 2-cp, A is the pre-exponential factor, E_a denotes the apparent activation energy, R is the molar gas constant, a, b, c is the order of reaction of 2-cp, O_2 , H_2O , respectively.

Combining the rate law of (14) with the definition of conversion and the design equation for a constant-volume, we can get:

$$\frac{dX}{dt} = A \exp\left(-\frac{E_a}{RT}\right) [2CP]_o^{a-1} (1-X)^a [O_2]_o^b [H_2O]^c \quad (15)$$

Here, X represents the conversion rate of 2-cp.

Considering that when in the presence of large volume of O_2 and H_2O (in excess), the concentration of the O_2 and H_2O can be taken as constant and equal to their initial concentrations, namely $[O_2]_r = [O_2]_0$, $[H_2O]_r = [H_2O]_0$. Reorganizing and integrating the equation (15) with the initial condition ($X = 0, \tau = 0$) lead to:

$$\ln(1-X) = -A \exp\left(-\frac{E_a}{RT}\right) [O_2]_0^b [H_2O]_0^c \tau \quad \text{if } a = 1 \quad (16)$$

$$\left[(1-X)^{1-a} - 1 \right] = (a-1) A \exp\left(-\frac{E_a}{RT}\right) [2CP]_0^{a-1} [O_2]_0^b [H_2O]_0^c \tau \quad \text{if } a \neq 1 \quad (17)$$

Our objective is to estimate the five parameters (a, b, c, A, E_a) according to the experimental data provided in Literature [18]. This optimization problem is the one with high dimension, high non-linearity besides possess many local optima. The objective function is defined as:

$$f = \sum_{i=1}^n (X_i - X'_i)^2 \quad (18)$$

where n is the sample size, X_i is the conversion rate of 2-cp in the i th sample data while X'_i is the conversion rate calculated by the kinetic model.

For all the three algorithms, the population size is 50 and the maximum number of iteration is 200. The optimization results are listed in Table 3, which reveals that HDE outperforms DE and PSO in terms of the average and worst values. The parameters found by HDE are: $a = 0.808$, $b = 0.444$, $c = 0.324$, $A = 63.547$ and $E_a = 45625.75$.

Table 3. Results obtained by HDE, DE and PSO in Case 2

Algorithm	HDE	DE	PSO
Best	0.217685	0.217685	0.217685
Average	0.217685	0.217725	0.217819
Worst	0.217685	0.218037	0.218733

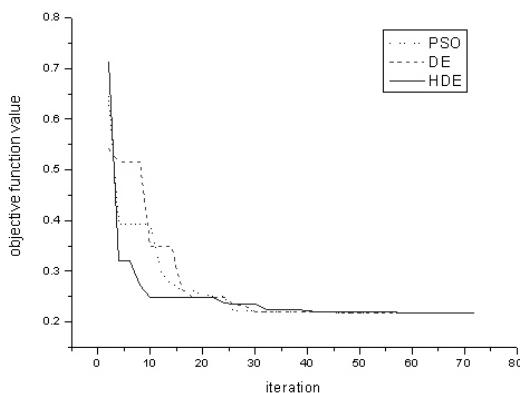


Fig. 4. Objective function value of the best individual versus iteration in Case 2

The objective function value of the best individual in the population versus iteration is shown in Fig. 4, which illustrates the high efficiency of HDE.

5 Conclusion

Parameter estimation in chemical kinetics plays a prominent role in establishing the kinetic models for chemical reactions. To develop a highly efficient and effective technique for this numerical optimization problem, we incorporate a hierarchical structure into the differential evolution. In the proposed algorithm, individuals are arranged hierarchically according to their qualities. The mutation scheme is modified by setting the mutation base on the hierarchy while the scaling factor is adjusted hierarchically and adaptively. To study the performance of HDE, we apply it to two chemical kinetic models. The optimization results demonstrate that HDE is a promising technique to estimate parameter in chemical kinetics.

Our future work will focus on applying HDE to more complicated parameter estimation problems in chemical or biological kinetics.

References

- Adam, B.S., James, W.T., Paul, I.B., William, H.G.: Global Dynamic Optimization for Parameter Estimation in Chemical Kinetics. *J. Phys. Chem.* 110, 971–976 (2006)
- Bailey, J.E., Ollis, D.E.: Biochemical Engineering Fundamentals. McGraw-Hill, New York (1986)
- Denn, M.M.: Optimization by Variational Methods. McGraw-Hill, New York (1969)
- Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
- Dorigo, M.: Optimization, Learning and Natural Algorithms, Ph.D thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy (1992)
- Kennedy, K., Eberhart, R.C.: Particle Swarm Optimization. In: Proc. IEEE Int. Conf. Neural Netw., pp. 1942–1948 (1995)

7. Storn, R., Price, K.V.: Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Global Opt.* 11(4), 341–359 (1997)
8. Yan, X.F., Chen, D.Z., Hu, S.X., Ding, J.W.: Estimation of Kinetic Parameters Using Chaos Genetic Algorithms. *Journal of Chemical industry and Engineering* 53(8), 810–814 (2002)
9. Zhang, B., Chen, D.Z., Rao, J.: Estimation of Kinetic Parameters by Using Eugenic Evolution Programming. *Journal of Chemical Engineering of Chinese Universities* 18(5), 638–642 (2004)
10. Price, K., Storn, R., Lampinen, J.: *Differential Evolution - A Practical Approach to Global Optimization*. Springer, Heidelberg (2005)
11. Kennedy, J., Mendes, R.: Population Structure and Particle Swarm Performance. In: Proc. Congr. Evolutionary Computation, pp. 1671–1676 (2002)
12. Suganthan, P.N.: Particle Swarm Optimizer with Neighborhood Operator. In: Proc. Congr. Evolutionary Computation, pp. 1958–1962 (1999)
13. Janson, S., Middendorf, M.: A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant. *IEEE Transaction on Systems, Man, and Cybernetics—Part B: Cybernetics* 35(6) (2005)
14. Standen, A.: *Kirk-Othmer Encyclopedia of Chemical Technology*, 2nd edn., vol. 7, p. 191. Interscience Publishers, New York (1972)
15. Yokata, T., Sakaguchi, S., Ishii, Y.: Aerobic Oxidation of Benzene to Biphenyl Using a Pd(II)/Molybdovanadophosphoric Acid Catalytic System. *Adv. Synth. Catal.* 344(8) (2002)
16. Zhu, Z.N., Dai, Y.C.: *Chemical Process Data Treatment and Experiment Design*, pp. 190–191. Hydrocarbon Processing Press, Beijing (1989)
17. Pray, H.A., Schweinert, C.E., Minnich, B.H.: Solubility of Hydrogenoxygen Nitrogen and Helium in Water at Elevated Temperature. *Ind. Eng. Chem.* 44, 1146–1151 (1952)
18. Li, R.K., Savage, P.E., Szmukler, D.: 2-Chlorophenol Oxidation in SuperCritical Water: Global Kinetics and Reaction Products. *Journal of AIChE* 39(1), 178–187 (1993)

Differential Evolution Based on Improved Learning Strategy

Yuan Shi, Zhen-zhong Lan, and Xiang-hu Feng

Software School, Sun Yat-sen University, Guangzhou, P.R. China
sycaszsu@gmail.com

Abstract. From a learning perspective, the mutation scheme in differential evolution (DE) can be regarded as a learning strategy. When mutating, three random individuals are selected and placed in a random order. This strategy, however, probably suffers some drawbacks which can slow down the convergence rate. To improve the efficiency of classic DE, this paper proposes a differential evolution based on improved learning strategy (ILSDE). The proposed learning strategy, inspired by the learning theory of Confucius, places the three individuals in a more reasonable order. Experimenting with 23 test functions, we demonstrate that ILSDE performs better than classic DE.

Keywords: Differential evolution, mutation scheme, learning strategy.

1 Introduction

For human beings, leaning is a fundamental process through which we gain knowledge and explore the world outside. By learning, an individual is able to get information from others and thus broaden personal experience. Therefore, learning provides a way for a group of people to cooperate and share their ideas and experience. In other words, learning provides a cooperative way to solve complicated problems. This is one of the essential characteristics of a class of algorithms, namely evolutionary algorithms (EAs).

EAs, inspired by biological phenomenon, are a branch of stochastic search algorithms. During last several decades, a variety of EAs have been developed and applied to various problems in science and engineering. Some popular ones are genetic algorithm (GA) [1], ant colony optimization (ACO) [2], particle swarm optimization (PSO) [3], differential evolution (DE) [4] and so forth. In GA, an individual learns from another one by the crossover operation and the direction of learning is guaranteed by the selection operation. In ACO, through pheromone deposit and pheromone decay, good information is enhanced while bad information is weakened. Therefore, ants are attracted to superior areas and further improve the solutions. In PSO, each particle learns from two positions: the global best position found so far and the personal best position found so far. Thus particles quickly fly towards promising areas just like bird flocking. DE proposes another distinctive learning strategy, in which an individual learns information from three other individuals. The strategy first randomly selects one individual as the mutation base; then adds the weighted differences between two other randomly selected individuals to the mutation base.

The mutation scheme of classic DE is meaningful and shares some similarity with the learning theory of Confucius who is one of the famous ideologists of ancient China. He said, "When I walk along with two others, they may serve me as my teachers. I will select their good qualities and follow them, know their bad qualities and avoid them." [5] According to his theory, human beings learn in a similar yet more intelligent and effective way compared with that in classic DE. The major difference is that human beings are sophisticated enough to distinguish good and bad information. Therefore the efficiency and effectiveness of learning in classic DE can be improved if we take some ideas from the Confucius' learning theory.

In this paper, we design an improved learning strategy based on the learning theory of Confucius and apply it to the mutation scheme in DE. In the proposed mutation scheme, the information of the good individual and the differences between the good and bad ones are utilized in a specific manner. In other words, the three randomly selected individuals are placed in a specific yet more reasonable order. Numerical experiments on 23 test functions are carried out to test the performance of differential evolution based on the improved learning strategy (ILSDE). Experimental results demonstrate the superiority of ILSDE.

2 Problem Formulation

In this paper, the following function optimization problem is addressed:

$$\text{minimize } f(x) \quad \text{s.t. } x \in \Omega$$

where $x = (x_1, x_2, \dots, x_N)$ is the continuous variable vector in domain $\Omega \subset R^N$, and $f(x) : \Omega \rightarrow R$ is a continues real-valued function. The domain Ω is defined by specifying upper $u = (u_1, u_2, \dots, u_N)$ and lower $l = (l_1, l_2, \dots, l_N)$ bounds, respectively.

3 Differential Evolution

Differential Evolution (DE) is a population-based EA over continuous domain. The population is composed of PS individual $x_{i,G}, i = 1, 2, \dots, PS$ where G denotes the current generation. Each individual $x_{i,G}$ is represented by a D dimensional vector:

$$x_{i,G} = (x_{i1,G}, x_{i2,G}, \dots, x_{iD,G})$$

Like other EAs, DE guides the population towards the global optimum through repeated cycles of evolutionary operations, including mutation, crossover and selection. Detailed description of the three operations is given as follows.

(1) Mutation

For each individual $x_{i,G}, i = 1, 2, \dots, PS$, the mutation operation is applied by first randomly selecting three other individuals, denoted by $x_{r_1,G}, x_{r_2,G}$ and $x_{r_3,G}$, then a mutated individual $v_{i,G}$ is generated according to the following equation:

$$v_{i,G} = x_{r_1,G} + F \cdot (x_{r_2,G} - x_{r_3,G}) \quad (1)$$

where $i, r_1, r_2, r_3 \in [1, PS]$ and $i \neq r_1 \neq r_2 \neq r_3$, while $F \in [0, 2]$, is a positive real factor named scaling factor

Literature [4] discussed some other variants of DE in which more than three individuals are involved in the mutation operation.

(2) Crossover

Crossover operation recombines $x_{i,G}$ and $v_{i,G}$ to yield a trail individual $u_{i,G+1}$. The operation is carried out following the equation below:

$$u_{ij,G+1} = \begin{cases} v_{ij,G}, & \text{if } r(j) < CR \text{ or } j = rn(i) \\ x_{ij,G}, & \text{otherwise} \end{cases} \quad (2)$$

with $j = 1, 2, \dots, D$. $r(j) \in [0, 1]$ is the j th evaluation of a uniform random number generator and $rn(i) \in \{1, \dots, D\}$ is a randomly chosen index which ensures that $u_{i,G+1}$ gets at least one element from $v_{i,G}$.

(3) Selection

A one-to-one competition is played between $x_{i,G}$ and $u_{i,G+1}$ after crossover operation. The better one will be promoted to the next generation according to the following equation:

$$x_{i,G+1} = \begin{cases} u_{i,G+1}, & \text{if } f(u_{i,G+1}) < f(x_{i,G}) \\ x_{i,G}, & \text{otherwise} \end{cases} \quad (3)$$

4 DE Based on Improved Learning Strategy

The mutation scheme of DE can be regarded as a learning strategy. It randomly determines the mutation base and the other two individuals to generate the weighted differences in classic DE. This strategy, however, probably suffers from two drawbacks. First, if the quality of the randomly selected mutation base is relatively bad, the generated mutated individual is likely to be unfavorable. Second, since the two individuals for calculating the weighted difference are randomly placed, the direction of the difference may be too random, making the mutation a bit inefficient. Those two drawbacks can slow down the convergence rate of the algorithm considerably.

How to improve the efficiency of the mutation scheme? In other words, how to make the learning strategy more efficient? The Confusions' learning theory puts forward a feasible and promising solution. According to his words, "I will select their good qualities and follow them, know their bad qualities and avoid them", we should design an improved learning strategy in which the mutated individual learns the good qualities but avoids the bad qualities of the random selected individuals.

Generally, suppose K ($K \geq 3$) individuals are randomly chosen to perform the mutation operation. To “follow the good qualities”, the best one among the K individuals, referred as $lbest$, is set as the mutation base. To “avoid the bad qualities”, the direction of the weighted difference should be generated not only towards the good individual, but also apart from the bad one. Thus, the weighted difference of the second best individual ($slbest$) and the worst one ($lworst$) is used. Mathematically, this improved learning strategy can be represented by the formula:

$$v_{i,G} = lbest + F \cdot (slbest - lworst) \quad (4)$$

where the parameter F and $v_{i,G}$ are the same as those in (1).

5 Numerical Experiments

5.1 Test Functions

To test the performance of ILSDE, 23 commonly-used test functions [6] of three categories are chosen. Table 1 lists the test functions and their key properties. These functions can be divided into three categories. $f_1 - f_7$ are unimodal functions, which are relatively easy to optimize. They can be used to test the converge rate of an optimization algorithm. $f_8 - f_{13}$ are multimodal functions where the number of local minima increases exponentially with the problem dimension. $f_{14} - f_{23}$ are multimodal functions having a few local minima, which are always used to test the ability of an optimization algorithm in escaping from deceptive optima and locating the desired near-global solution. Details of some functions can be found in literature [6].

Table 1. Test Functions

Benchmark Function	Ω	f_{min}
$f_1 = \sum_{i=1}^n x_i^2$	$[-100,100]^{30}$	0
$f_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10,10]^{30}$	0
$f_3 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100,100]^{30}$	0
$f_4 = \max_i(x_i , 1 \leq i \leq n)$	$[-100,100]^{30}$	0
$f_5 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]^{30}$	0
$f_6 = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$	$[-100,100]^{30}$	0
$f_7 = \sum_{i=1}^n i x_i^4 + \text{random}[0,1)$	$[-1.28,1.28]^{30}$	0
$f_8 = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	$[-500,500]^{30}$	-12569.5
$f_9 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12,5.12]^{30}$	0

$f_{10} = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i)$ + 20 + e	$[-32, 32]^{30}$	0
$f_{11} = 1/4000 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^{30}$	0
$f_{12} = \frac{1}{10} \{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	$[-50, 50]^{30}$	0
$f_{13} = \frac{\pi}{n} \{ 10 \sin^2(3\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + \sin^2(3\pi y_{i+1})] + (y_n - 1)^2 [1 + \sin^2(2\pi y_n)] \} + \sum_{i=1}^n u(y_i, 5, 100, 4)$	$[-50, 50]^{30}$	0
$f_{14} = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right]^{-1}$	$[-65.536, 65.536]^2$	1
$f_{15} = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	$[-5, 5]^4$	0.0003
$f_{16} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]^2$	-1.03
$f_{17} = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	$[-5, 10] \times [0, 15]$	0.398
$f_{18} = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	$[-2, 2]^2$	3
$f_{19} = -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^4 a_{ij} (x_j - p_{ij})^2]$	$[0, 1]^4$	-3.86
$f_{20} = -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2]$	$[0, 1]^6$	-3.32
$f_{21} = -\sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	$[0, 10]^4$	-10.1532
$f_{22} = -\sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	$[0, 10]^4$	-10.4029
$f_{23} = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	$[0, 10]^4$	-10.5364

5.2 Experimental Setup

For both algorithms, the parameters are set the same for all test functions. That is: population size $PS = 100$, crossover rate $CR = 0.9$, and the scaling factor $F = 0.5$. These parameters follow the suggestions from Rainer Storn and Kenneth price [4], Jakob Vesterstrom [7]., The initial population is generated uniformly at random in the search domain of the functions.

5.3 Comparisons between DE and ILSDE

The performance of the ILSDE is evaluated based on the optimization results on the 23 test functions. The comparisons between ILSDE and DE are described in Table 2, in which mean and variance denote the mean value and standard deviation of the optima in 30 runs, respectively. The results reveal that ILSDE can achieve higher accuracy in the obtained optimum both in unimodal functions and multimodal functions.

Table 2. Comparison between DE and ILSDE

FN	Computational effort	Mean Best (Variance)	
		ILSDE	DE
f_1	100000	1.19×10^{-19} (1.34×10^{-19})	1.81×10^{-8} (7.54×10^{-8})
f_2	100000	1.67×10^{-9} (7.95×10^{-9})	1.52×10^{-4} (3.94×10^{-4})
f_3	200000	1.26×10^{-8} (1.27×10^{-8})	9.00×10^{-2} (3.34×10^{-2})
f_4	200000	8.51×10^{-6} (3.37×10^{-6})	1.85×10^{-1} (2.46×10^{-1})
f_5	500000	1.08×10^{-29} (2.31×10^{-29})	1.10×10^{-12} (1.79×10^{-11})
f_6	50000	0 (0)	0 (0)
f_7	50000	2.60×10^{-2} (5.63×10^{-3})	5.21×10^{-2} (1.26×10^{-2})
f_8	50000	-6281.2 (3.90×10^2)	-5354.0 (2.22×10^2)
f_9	50000	187.37 (9.21)	199.8 (12.05)
f_{10}	50000	1.66×10^{-4} (4.57×10^{-5})	3.21×10^{-1} (1.26×10^{-1})
f_{11}	50000	7.42×10^{-4} (4.18×10^{-3})	2.39×10^{-1} (1.42×10^{-1})
f_{12}	50000	3.67×10^{-8} (1.89×10^{-8})	2.24×10^{-3} (1.34×10^{-3})
f_{13}	50000	4.78×10^{-7} (2.98×10^{-7})	4.98×10^{-2} (2.05×10^{-2})
f_{14}	10000	0.998 (9.93×10^{-17})	0.998 (3.07×10^{-16})
f_{15}	50000	3.99×10^{-4} (2.74×10^{-4})	4.29×10^{-4} (3.11×10^{-4})
f_{16}	5000	-1.03 (9.45×10^{-9})	-1.03 (5.73×10^{-7})
f_{17}	5000	0.398 (5.82×10^{-9})	0.398 (2.17×10^{-7})
f_{18}	5000	3.00 (8.36×10^{-13})	3.00 (2.84×10^{-9})
f_{19}	5000	-3.86	-3.86

f_{20}	5000	(1.78×10 ⁻¹²) -3.25 (5.54×10 ⁻²) -10.15 (8.25×10 ⁻¹¹) -10.40 (2.08×10 ⁻¹³) -10.53 (9.21×10 ⁻¹²)	(3.48×10 ⁻⁸) -3.21 (2.97×10 ⁻²) -10.15 (3.69×10 ⁻⁵) -10.40 (1.48×10 ⁻⁷) -10.53 (2.26×10 ⁻⁶)
f_{21}	10000		
f_{22}	10000		
f_{23}	10000		

5.4 Detailed Analysis

(1) Unimodal Function

Without loss of generality, f_1, f_5 are chosen to represent the unimodal function and given more detailed analysis.

f_1 is relatively simple to optimize. It is commonly used to test the convergence rate of an algorithm. While f_5 always appears to be one of the most difficult functions since its global minimum is inside a long, narrow, parabolic shaped flat valley, the variables are strongly dependent, and the gradients generally do not point towards the optimum. As indicated in Table 2 and Fig 1, ILSDE outperforms DE in terms of convergence rate.

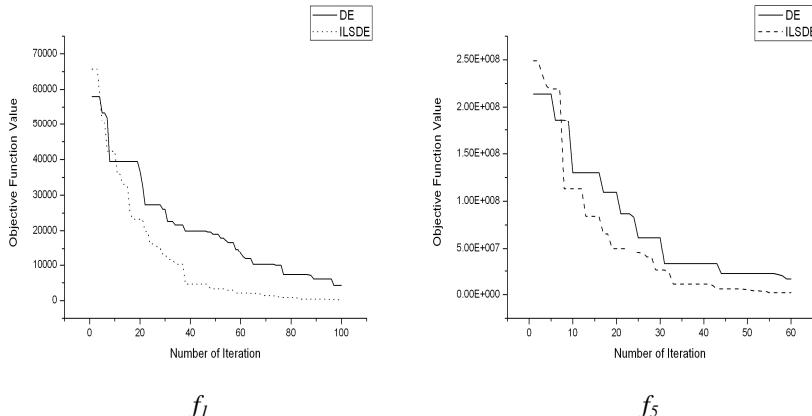


Fig. 1. Comparisons of converge rate between DE and ILSDE on f_1 and f_5

(2) Multimodal Function

f_9, f_{10} and f_{16}, f_{17} are selected to represent multimodal functions with high dimensions and low dimensions, respectively.

f_9 and f_{10} is highly multimodal. Most evolutionary algorithms with relatively small population tend to trap into local optima. Actually, both DE and ILSDE trap into local minima when CR is 0.9, but when CR is set to 0.1, both of the algorithms are able to obtain much better results in f_9 . As can be seen in Fig 2 and Table 2, ILSDE can perform better in escaping from deceptive optima.

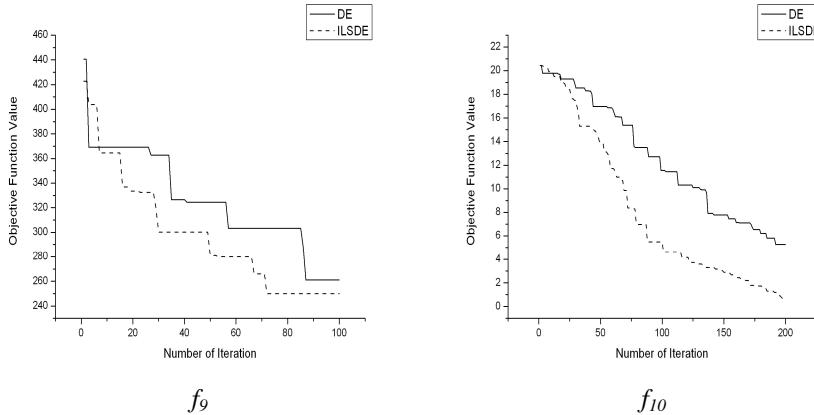


Fig. 2. comparisons of convergence rate between DE and ILSDE on f_9 and f_{10}

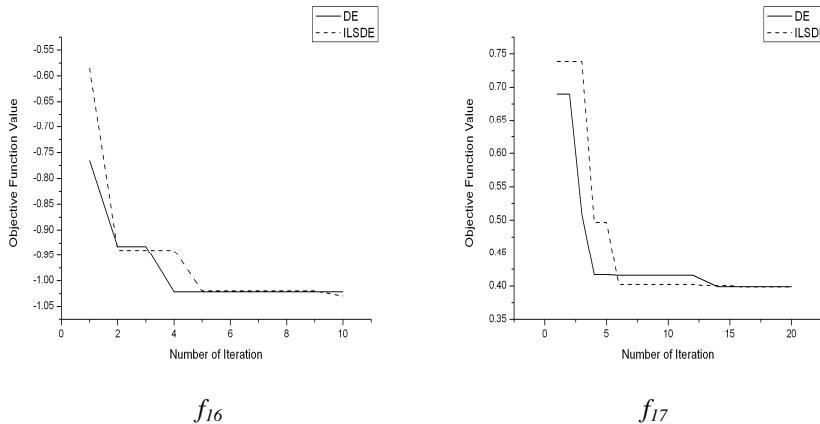


Fig. 3. Comparisons of convergence rate between DE and ILSDE on f_{16} and f_{17}

Both f_{16} and f_{17} are two-dimensional functions with a few local optima. From Fig 3 and Table 2, no significant difference can be observed between DE and ILSDE in this category of functions.

In the previous experiments, the value of K is 3. It's necessary and interesting to have a study on the influences when setting different values for K . Here, we carry out another set of experiments on higher K ($K=5, 10$). Fig. 4 illustrates their performances on f_1 , f_5 , f_9 and f_{10} , showing that they converge faster than ILSDE when $K=3$. On the other hand, however, as we have observed from some test cases, faster convergence rate may lead to easier trapping into local minima. Therefore, further analysis and study should be made in our future work.

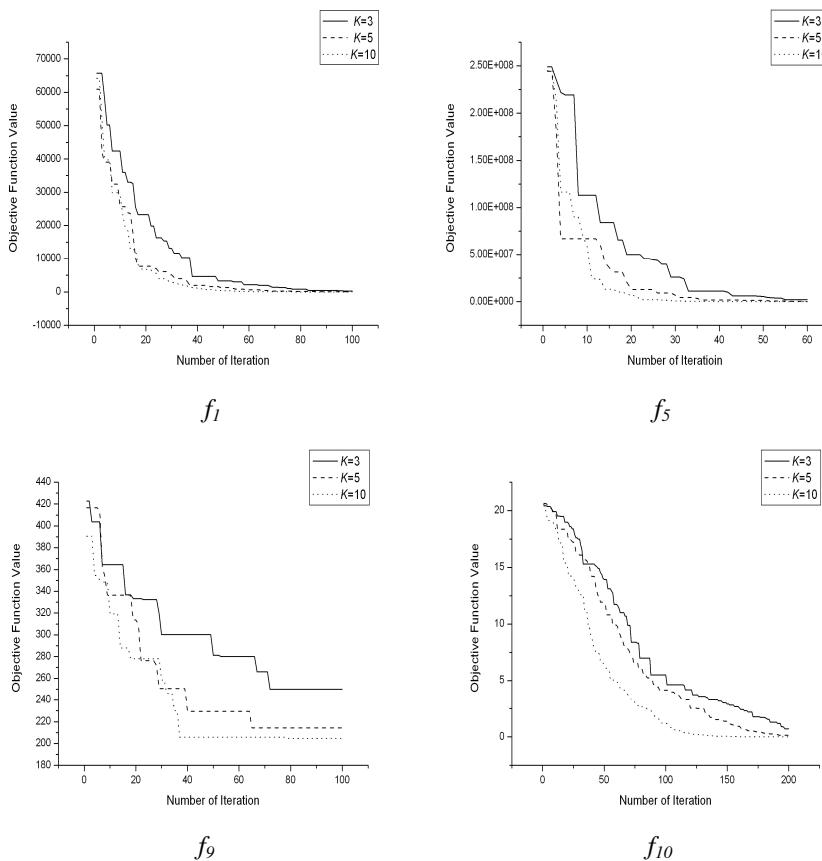


Fig. 4. Comparisons of convergence rate of ILSDE using different K on f_1, f_5, f_9 and f_{10}

6 Conclusion

In this paper, the mutation scheme in DE is studied as a learning strategy. The strategy in classic DE has some drawbacks since it randomly places the selected individuals to generate mutated individual. Taking some ideas from the learning theory of Confucius, we designed an improved learning strategy involving more intelligence. Extensive experiments show that ILSDE converges faster and obtains better results than classic DE in most test cases.

Our future work will be emphasized on finding more effective learning strategy. Also, we will apply the proposed approach to solve some real-world problems.

References

1. Holland, J.H.: *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor (1975)
2. Dorigo, M.: *Optimization, Learning and Natural Algorithms*. Ph.D thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy (1992)

3. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. IEEE Int. Conf. Neural Netw., pp. 1942–1948 (1995)
4. Storn, R., Price, K.V.: Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Opt.* 11(4), 341–359 (1997)
5. Confucius, D.C.L.: *Confucius: The Analects*. Chinese University Press (1992)
6. Yao, X., Liu, Y., Lin, G.: Evolutionary Programming Made Faster. *IEEE Trans. on Evolutionary Computation* 3, 82–102 (1999)
7. Vesterstrom, J., Thomsen, B.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems *Evolutionary Computation*. In: CEC 2004. Congress, vol. 2(19-23), pp. 1980–1987 (2004)

SalienceGraph: Visualizing Salience Dynamics of Written Discourse by Using Reference Probability and PLSA

Shun Shiramatsu, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno

Graduate School of Informatics, Kyoto University
`{siramatu,komatani,ogata,okuno}@kuis.kyoto-u.ac.jp`

Abstract. Since public involvement in the decision-making process for community development needs a lot of efforts and time, support tools for speeding up the consensus building process among stakeholders are required. This paper presents a new method for finding, tracking and visualizing participants' concerns (topics) from the record of a public debate. For finding topics, we use the salience of a term, which is computed as its *reference probability* based on referential coherence in the Centering Theory. Our system first annotates a debate record or minute into Global Document Annotation (GDA) format automatically, and then computes the salience of each term from the GDA-annotated text sentence by sentence. Then, by using the Probabilistic Latent Semantic Analytsis (PLSA), our system reduces the dimensions of the vector of salience values of terms into a set of major latent topics. For tracking topics, we use the salience dynamics, which is computed as the temporal change of joint attention to the major latent topics with additional user-supplied terms. The resulting graph is called SalienceGraph. For visualizing SalienceGraph, we use 3D visualizer with GUI designed by “overview first, zoom and filter, then details on demand” principle. SalienceGraph provides more accurate trajectory of topics than conventional TF-IDF.

Keywords: discourse analysis, visualization, discourse salience, PLSA.

1 Introduction

1.1 Background and Aim

Public involvement (PI), a citizen participation process in the decision-making of public policy, is characterized as an interactive communication process among stakeholders [1]. PI processes such as public debate, town meeting, and workshop have been getting popular in Japan recently. The decision-making through PI processes, however, needs a lot of efforts and time. For instance, readers of the transcription of long debate may feel difficulty in overviewing the contextual flow and finding the target section. Such overviewing work may be alleviated, if tools based on natural languaage processing are available for automatic identification of participants' concerns (topics) and for tracking various concerns sentence by

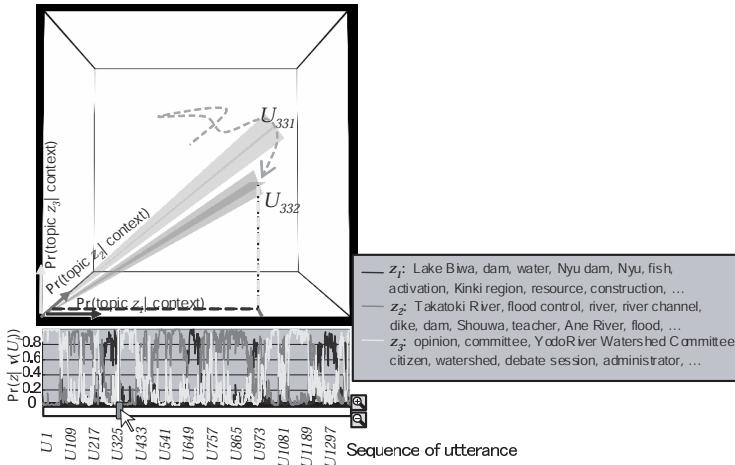


Fig. 1. SalienceGraph: Visualizing the salience dynamics of three latent topics in a long discourse in 3D and 2D

sentence. In addition, such tools may speed up the consensus building process among stakeholders.

Our aim is to develop a new method for visualizing the dynamics of topics in a discourse (e.g., the record of a public debate) in order to help users find and track participants' topics. Such visualization would give users an overview of the contextual flow, as shown in Figure 1. We call this visualization scheme "SalienceGraph". Figure 1 depicts the dynamics of salience of topics or terms, i.e., how the degree of joint attention to the topics or terms varies temporally. It contains three latent topics specified by z_1, \dots, z_3 and their reference probabilities are shown in 2D and 3D. 3D visualizer displays how the salience vector changes temporally.

The GUI for the visualizer is designed according to Shneiderman's *Visual Information-Seeking Mantra*, that is "overview first, zoom and filter, then details on demand" [2]. After getting the overview of the topics, users may either browse the salience dynamics of a particular latent topic, or inspect the discourse at a particular point by consulting the record.

1.2 Requirements and Our Approach

To achieve our aim, we define a metric for discourse salience; i.e., the degree of joint attention to terms and topics at each utterance unit. The method for calculating the metric should meet several requirements.

[Requirement 1] Quantification: Discourse salience should be defined as a quantitative metric rather than an ordinal one because quantitative scales are generally more suitable to engineering applications than ordinal ones.

[Requirement 2] Objectivity: Discourse salience should be objectively measured using corpus-based statistics.

[Requirement 3] Integration of salience factors: Discourse salience depends on term frequency, grammatical function, recency effect, etc. A metric for discourse salience should be designed by integrating these factors.

[Requirement 4] Dynamic transition: Discourse salience dynamically changes with each utterance unit. A metric for discourse salience should reflect these dynamic transitions.

[Requirement 5] Methodology for optimization: The method for calculating discourse salience should be optimizable on the basis of a corpus. The optimization requires a criterion for evaluating the calculation method.

To meet these requirements, we designed a calculation method with the following approaches.

[Approach 1] Probabilistic definition: We quantified a probabilistic metric for discourse salience. This quantification is based on the assumption that a salient entity tends to be referred to in the subsequent utterance unit.

[Approach 2] Corpus-based statistics: We used a statistical approach based on a corpus to design an objective calculation method.

[Approach 3] Metrics definition on feature space: To integrate various salience factors, we defined the metric on a feature space consisting of the salience factors.

[Approach 4] Incorporate recency effect: To deal with the dynamic transition of discourse salience, we incorporated the recency effect of term occurrences in the preceding discourse.

[Approach 5] Evaluation criterion: To enable the calculation to be optimized, we designed a criterion for evaluating the accuracy of the calculation method.

The linguistic features for calculating salience (in Approach 3) are extracted from the text annotated with Global Document Annotation (GDA) [3]. GDA is an XML vocabulary for linguistic metadata. Before calculating salience, we should automatically annotate the transcription of discourse with GDA tags representing the analysis result by the CaboCha [4], a Japanese dependency analyzer.

The joint attentional state among discourse participants at a moment can be represented as a high-dimensional vector comprising the salience values for each term. We developed a scheme for visualizing the salience dynamics on the basis of the transition of the salience values. Such a scheme should meet the following requirement.

[Requirement 6] Decrease number of salience values: There are too many values to visualize the salience dynamics because they can be calculated for every term in the target discourse. Decreasing the number of salience values is needed to visualize the salience dynamics.

To meet this requirement, we use *Probabilistic Latent Semantic Analysis* (PLSA).

[Approach 6] Dimensional compression: To decrease the number of salience values, we used dimensional compression with PLSA. To help a user to grasp the meaning of PLSA latent topics, terms representing the topics are extracted on the basis of *Pointwise Mutual Information* (PMI). The user can identify particular terms to view their salience dynamics from the terms representing the latent topics.

We next describe our salience metric and SalienceGraph, our scheme for visualizing salience dynamics.

2 Reference Probability: Metric for Discourse Salience

This section describes our method for calculating discourse salience.

2.1 Quantification with Probabilistic Approach

We designed a metric for discourse salience by using a probabilistic approach on the basis of an assumption consistent with the Centering Theory [5]. This theory contains the rule of referential coherence; that is, a salient entity tends to be referenced in the subsequent utterance unit. We previously defined a metric for discourse salience as the *reference probability* [6]. We use it here as well. Let w be a word, U_i be the current utterance unit, U_{i+1} be the subsequent one, and $\text{pre}(U_i) = [U_1, \dots, U_i]$ be the preceding discourse. Let $w' \xrightarrow{\text{coref}} w$ in U_{i+1} denote that a word in U_{i+1} has a coreference relation with w . The discourse salience of w at U_i is defined as the reference probability:

$$\begin{aligned} (\text{Salience of } w \text{ at } U_i) &= \Pr(\exists w' \xrightarrow{\text{coref}} w \text{ in } U_{i+1} | \text{pre}(U_i)) \\ &= \Pr(w | \text{pre}(U_i)). \end{aligned}$$

$\Pr(w | \text{pre}(U_i))$ is simplified notation for the reference probability. This probabilistic definition gives a quantification of discourse salience (Requirement 1).

2.2 Design of Objective Calculation by Integrating Salience Factors

To design an objective calculation method, we need to use corpus-based statistics for calculating $\Pr(w | \text{pre}(U_i))$ (Requirement 2). To integrate various salience factors, we need to define $\Pr(w | \text{pre}(U_i))$ on a feature space consisting of the salience factors in $\text{pre}(U_i)$ (Requirement 3). The features are extracted from $\langle w', \text{pre}(U_i) \rangle$ for $\forall w'$ in $\text{pre}(U_i)$ which co-refers to the referent of target w . For instance, the feature space consists of such salience factors as the following ones.

- **Recency effect:** utterance distance from $w' \xrightarrow{\text{coref}} w$ to U_{i+1}
- **Frequency:** frequency of $w' \xrightarrow{\text{coref}} w$ in $\text{pre}(U_i)$
- **Grammatical function:** function word governing $w' \xrightarrow{\text{coref}} w$
- **Part-of-speech:** part-of-speech of $w' \xrightarrow{\text{coref}} w$

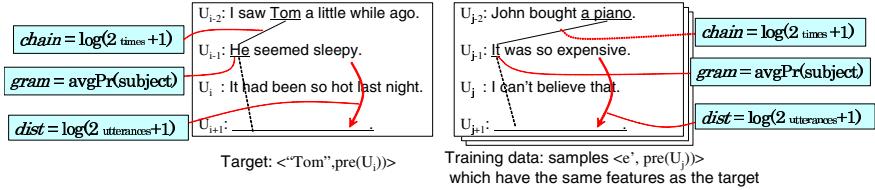


Fig. 2. Basic idea for calculating reference probability

The features are extracted from the text which is annotated with Global Document Annotation (GDA) [3] tags which represent the analysis result of the CaboCha [4], a Japanese dependency parser.

The salience factors are integrated into $\Pr(w|\text{pre}(U_i))$, the metric for discourse salience, through corpus-based learning. Ideally, it can be calculated using a corpus:

$$\Pr(w|\text{pre}(U_i)) = \frac{\#\{(w_c, U_j); \text{feat}(w_c, \text{pre}(U_j)) = \text{feat}(w, \text{pre}(U_i)) \wedge \exists w' \xrightarrow{\text{coref}} w_c \text{ in } U_{j+1}\}}{\#\{(w_c, U_j); \text{feat}(w_c, \text{pre}(U_j)) = \text{feat}(w, \text{pre}(U_i))\}},$$

where $\langle w, U_i \rangle$ denotes the target example, $\langle w_c, U_j \rangle$ denotes a sample in the corpus, and $\text{feat}(w_c, \text{pre}(U_j)) = \text{feat}(w, \text{pre}(U_i))$ denotes that the feature vector of w_c in $\text{pre}(U_j)$ is equal to that of the target w in $\text{pre}(U_i)$. This calculation in the ideal situation is illustrated in Fig. 2. The samples in the corpus are, however, not always dense enough to approximate $\Pr(w|\text{pre}(U_i))$ using the above ratio. A learning regression model from the corpus is generally needed to cope with the data sparseness. We use logistic regression to integrate the salience factors and calculate $\Pr(w|\text{pre}(U_i))$.

To apply logistic regression, the discrete features such as grammatical function and part-of-speech should be assigned to real numbers. Let $\text{feat}(w)$ denote a particular discrete feature of a word w . When $\text{feat}(w) = x$, we assign the real number $\text{avgPr}(x)$ to a particular discrete feature x .

$$\text{avgPr}(x) = \frac{\#\{(w, U_i); \text{feat}(w) = x \wedge \exists w' \xrightarrow{\text{coref}} w \text{ in } U_{i+1}\}}{\#\{(w, U_i); \text{feat}(w) = x\}}$$

The values assigned to discrete features are weighted on the basis of the recency effect as described next.

2.3 Window Function Representing Decay Curve of Recency Effect

The *recency effect* is the cognitive phenomenon in which recent occurrences are more likely to be recalled than old occurrences in a sequence of occurrences [7]. It is a particularly important factor when dealing with the dynamics of salience (Requirement 4). To deal with the recency effect of every $w' \xrightarrow{\text{coref}} w$ in $\text{pre}(U_i)$, we should weight the features of w' on the basis of the recency, i.e., the utterance distance from w' to U_{i+1} . The decay curve of the recency effect is represented as a window function that shifts with each utterance unit as shown in Fig. 3.

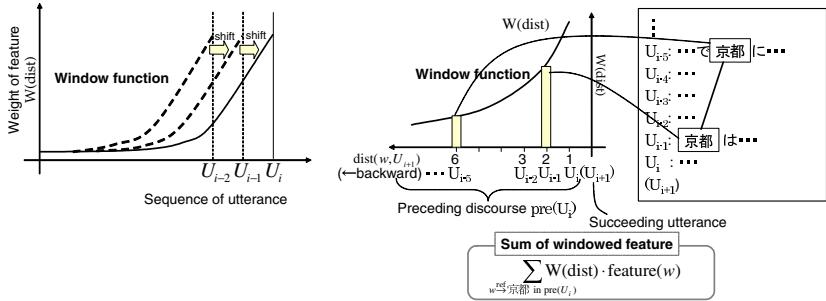


Fig. 3. Window function that represents decay curve of recency effect

The other features are weighted by the window function, as shown in Fig. 3. The shape of the window function is optimized using an evaluation criterion described later in this section. The features weighted by the window function are integrated by logistic regression as described next.

2.4 Applying Logistic Regression

Logistic regression requires a training phase, i.e., an estimation of the regression weights. Given that $\{\text{feat}_1(\langle w, U_i \rangle), \dots, \text{feat}_n(\langle w, U_i \rangle)\}$ is the set of features weighted by the window function, the regression weights satisfy a logit formula:

$$\log \frac{\Pr(w|\text{pre}(U_i))}{1 - \Pr(w|\text{pre}(U_i))} = b_0 + \sum_{k=1}^n b_k \cdot \text{feat}_k(\langle w, U_i \rangle).$$

The regression weights are calculated with the maximum-likelihood method by using a training corpus. The dummy variable used to train the regression model is defined as: $\text{isRef}(w, U_{i+1}) = \begin{cases} 1 & (\exists w' \xrightarrow{\text{coref}} w \text{ in } U_{i+1}) \\ 0 & (\text{otherwise}) \end{cases}$.

That is, we can calculate $\Pr(w|\text{pre}(U_i))$ by using a logistic regression model that explains $\text{isRef}(w, U_{i+1})$. If a trained regression model is used, $\Pr(w|\text{pre}(U_i))$ can be calculated using

$$\Pr(w|\text{pre}(U_i)) = \frac{1}{1 + \exp(-(b_0 + \sum_{k=1}^n b_k \cdot \text{feat}_k(\langle w, U_i \rangle)))}.$$

2.5 Methodology for Optimization

To optimize the method for calculating $\Pr(w|\text{pre}(U_i))$, we have to design a criterion for evaluating the accuracy of the calculation (Requirement 5). We designed the evaluation criterion on the basis of the assumption that a salient entity tends to be referred to in the subsequent utterance unit. We assume that the accuracy of a particular scheme for calculating salience can be represented by its ability to predict the subsequent references. This assumption is consistent with the rule of referential coherence in centering theory.

Given that $\text{sal}_m(w|\text{pre}(U_i))$ is a value calculated using method m , the evaluation measure is defined on the basis of the test-set corpus:

$$\text{evalSal}(m) = \text{cor}\left(\left[\text{sal}_m(w|\text{pre}(U_i))\right]_{\langle w, U_i \rangle}, \left[\text{isRef}(w, U_{i+1})\right]_{\langle w, U_i \rangle}\right),$$

where $\text{isRef}(w, U_{i+1})$ is the dummy variable defined above (i.e., 1 if $\exists w \xrightarrow{\text{ref}} e$ in U_{i+1} , otherwise 0), and $\text{cor}(x, y)$ denotes Pearson's correlation coefficient between x and y . Note again that $\text{evalSal}(m)$ is calculated not by using a training corpus but by using a test-set corpus. The window function described above is optimized so as to maximize $\text{evalSal}(m)$.

2.6 Visualization with Dimensional Compression Based on PLSA

There are too many salience values to visualize the salience dynamics because they can be calculated for every term in the target discourse. When the discourse contains N terms w_1, \dots, w_N , the attentional state at the moment of U_i is represented as a high-dimensional vector:

$$\mathbf{v}(U_i) = [\Pr(w_1|\text{pre}(U_i)), \dots, \Pr(w_N|\text{pre}(U_i))]^T.$$

To visualize the salience dynamics, we need to compress the dimensions. To do this, we compress the high-dimensional vector into a three-dimensional vector by using PLSA. The parameters $\Pr(z_k)$, $\Pr(w_j|z_k)$, and $\Pr(\mathbf{v}(U_i)|z_k)$ are determined using the EM algorithm so as to maximize log-likelihood L :

$$\begin{aligned} L &= \sum_{i,j} \Pr(w_j|\text{pre}(U_i)) \log \Pr(w_j, \mathbf{v}(U_i)) \\ &= \sum_{i,j} \Pr(w_j|\text{pre}(U_i)) \log \sum_k \Pr(z_k) \Pr(w_j|z_k) \Pr(\mathbf{v}(U_i)|z_k). \end{aligned}$$

To help a user grasp the meaning of latent topics z , we need to extract terms representing z . We use the following weighted *Pointwise Mutual Information* (PMI) to extract the terms w representing each z .

$$\Pr(w|z)\text{PMI}(w, z) = \Pr(w|z) \log \frac{\Pr(w, z)}{\Pr(w)\Pr(z)}$$

PMI is usually used as a degree of co-occurrence. To avoid infrequent terms that are not suitable for visualization, PMI is weighted with $\Pr(w|z)$.

3 Corpus-Based Experiment

We prepared the three corpora in Japanese, which are annotated with GDA tags.

CSJ: Four interview dialogues from the Corpus of Spontaneous Japanese (CSJ) [8], which contain

- 1,780 utterance units (IPUs; inter-pause units),
- 6.92 morphemes per utterance unit, and
- 1,180 anaphora relations annotated manually.

- Mainichi:** 3,000 newspaper articles in Japanese from the Mainichi Shinbun for 1994 (GSK2004-A [9]), which contain
- 63,221 utterance units (predicate clauses), 37,340 sentences,
 - 10.79 morphemes per utterance unit, and
 - 86,541 anaphora relations annotated manually.

YodoRiver: The minutes of a public debate in spoken Japanese [10] containing 1394 sentences.

The *CSJ* and *Mainichi* corpora were used to evaluate and to optimize our method for calculating the reference probability. The *YodoRiver* corpus was used to visualize the salience dynamics.

3.1 Optimization of Window Function

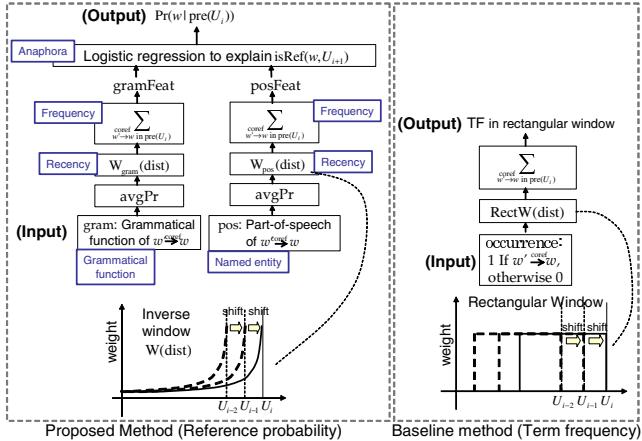
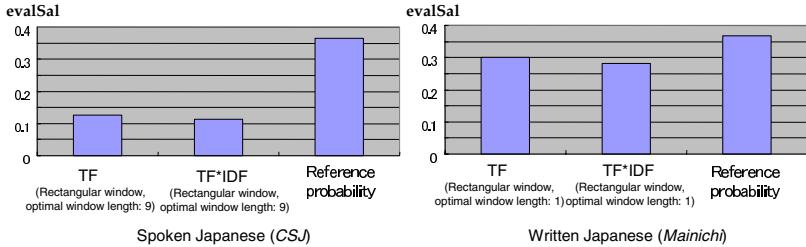
We need to clarify the optimal shape of the window function used to represent the decay curve of the recency effect in discourse. There are four candidates.

- **Rectangular window:** $W(\text{dist}) = \begin{cases} 1 & \text{dist} \leq k \\ 0 & \text{otherwise} \end{cases}$ (k is a variable parameter)
- **Gaussian window:** $W(\text{dist}) = \exp\left(-\frac{\text{dist}^2}{\sigma^2}\right)$ (σ is a variable parameter)
- **Exponential window:** $W(\text{dist}) = \exp\left(-\frac{\text{dist}}{T}\right)$ (T is a variable parameter)
- **Inverse window:** $W(\text{dist}) = \frac{1}{\text{dist}^d}$ (d is a variable parameter)

To do this, we experimentally measured $\text{evalSal}(m)$, the scale for evaluating calculation method m . In this experiment, we regarded the sum of the windowed features as the salience value calculated using the window function. The experimental results for *CSJ* and *Mainichi* showed that the inverse window is the most suitable for dealing with the salience dynamics. For *CSJ*, parameter d is optimized: $d = 1.14$ for grammatical function and 1.24 for part-of-speech. For *Mainichi*, $d = 3.01$ for grammatical function and 3.80 for part-of-speech. That is, the decay curve of the recency effect for *Mainichi* is more precipitous than that for *CSJ* because the newspaper articles in *Mainichi* are shorter than the interview dialogues in *CSJ*.

3.2 Evaluation of Regression-Based Calculation

We compared the evaluation scale, $\text{evalSal}(m)$, of our method for calculating reference probability (left side of Fig. 4) to that of the baseline method for the term frequency (TF) measured in an optimal rectangular window (right side of Fig. 4). The results are shown in Figure 5. The evaluation measures with the naive TF were 0.1048 for *CSJ* and 0.3013 for *Mainichi*. The evaluation measures with our proposed method (0.3652 for *CSJ* and 0.3680 for *Mainichi*) were greater. This means that our method can more effectively predict whether target entity is referred to in the subsequent U_{i+1} than the naive TF. The increase in effectiveness

**Fig. 4.** Proposed method and baseline method**Fig. 5.** Comparison of the proposed method with the baseline one (*CSJ* and *Mainichi*)

with our method was more significant for *CSJ* than for *Mainichi*. This indicates that handling spoken language needs more integration of the features (especially the recency effect) than handling written language.

3.3 Visualization of Long Discourse

As an example of visualization, we used the *YodoRiver* corpus, lengthy minutes of a public debate about a dam. They contain 1394 sentences. We regarded each sentence as an utterance unit. To calculate the reference probabilities for the minutes, we used a logistic regression model acquired from *CSJ*. Fig. 6 illustrates the result of the visualization.

The terms representing the latent topics (z_1, z_2 and z_3), as determined using $\Pr(w|z)\text{PMI}(w, z)$, are shown in Table 1. We interpret the results as meaning that z_1 corresponds to citizen's opinions about regional development, z_2 corresponds to their opinions about flood control and water damage, and z_3 roughly corresponds to the utterances by the chair.

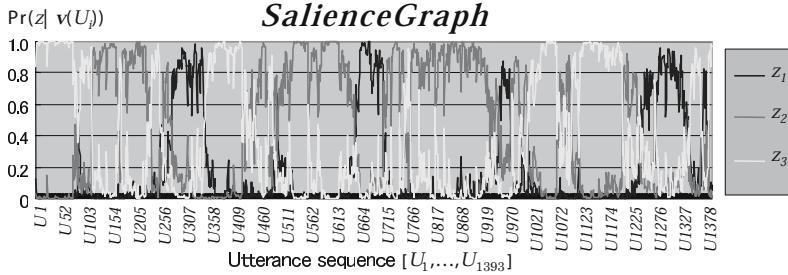


Fig. 6. SalienceGraph for lengthy minutes of a public debate (*YodoRiver* corpus)

Table 1. Terms representing each latent topic

Topic z	Term w with high $\Pr(w z)\text{PMI}(w,z)$ (translated from Japanese)	Interpretation
z_1	Lake Biwa, dam, water, Nyu dam, Nyu, fish, activation, Kinki region, resource, construction, regional development bureau, water level, periphery, Heisei, story, nation, people, flow, business, development, purpose, Kinki Regional Development, dried-up river, water resource, ...	Opinions about regional development
z_2	Takatoki River, flood control, river, river channel, dike, dam, Shouwa, teacher, Ane River, flood, region, children, water damage, water, problem, environment, revamping, water utilization, Shiga Prefecture, house, artificial ditch, damage, Shiga, Yaita, Torahime, Biwa-Machi, ...	Opinions about flood control and water damage
z_3	opinion, committee, YodoRiver Watershed Committee, citizen, watershed, debate session, administrator, river, management, Yodo River, profile, everyone, presenter, public, plan, audience, reflection, various quarters, debate, utterance, maintenance, academic expert, ...	Utterances by the chair

4 Application: User Interface for Browsing Discourse

SalienceGraph, our proposed scheme for visualizing salience dynamics, can be applied to a user interface for browsing long discourse, as illustrated in Fig. 7.

- (1) A user can automatically overview the contextual flow of discourse on the basis of the latent topics and the terms representing each topic.
- (2) The user filters out z_3 and appends the term “flood control” according to her interest.
- (3) She zooms in around U_{284} , a likely mixture point among z_1 , z_2 , and “flood”.
- (4) She reads around U_{284} by double-clicking on it.

This procedure meets Shneiderman’s ‘Visual Information-Seeking Mantra: “overview first, zoom and filter, then details on demand.”’ This user interface should be effective in browsing long discourse and analyzing the discussion. We

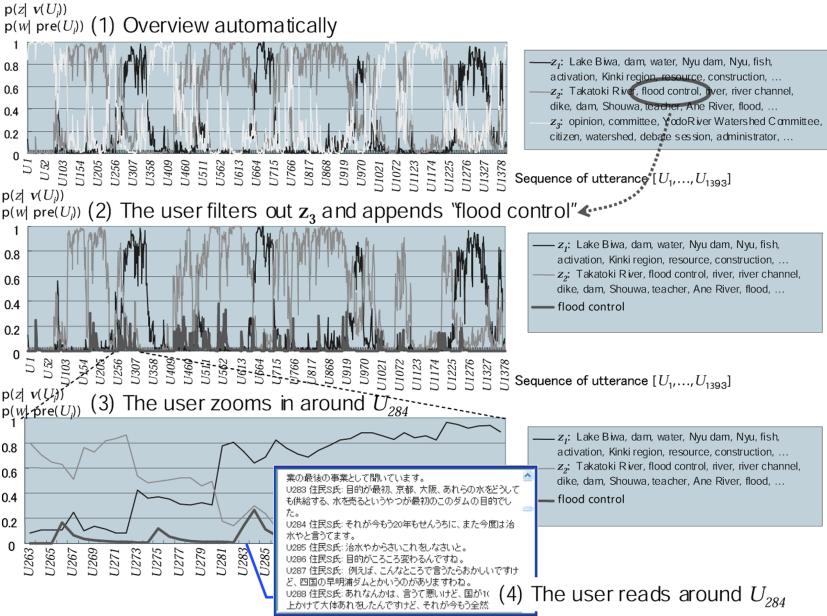


Fig. 7. Example of browsing conference minutes using SalienceGraph

plan to develop a discourse browsing system based on SalienceGraph and apply it to the analysis of public debate (e.g., for the study on public involvement).

5 Related Work

5.1 PLSA-Based Language Model

PLSA is often used for language modeling. In the conventional studies of language modeling, the Bag-of-Words vectors, which consist of term frequencies, are used for the training data representing the contextual history h . Our method is different from the conventional methods in terms of the training data for PLSA. We use vectors consisting of reference probabilities, which represent the attentional states at each utterance unit, as the training data. This is because the reference probabilities have a higher ability to predict the subsequent references than frequency-based metrics, as mentioned above.

Latent dirichlet allocation (LDA) [11] and *dirichlet mixture* (DM) [12], improvements to PLSA, are also used for language modeling. Mochihashi et al. proposed two schemes for integrating these models and the particle filter: mean shift model-LDA (MSM-LDA) and mean shift model-DM (MSM-DM) [13]. These schemes are potentially applicable to SalienceGraph. We will investigate them and consider to their use.

5.2 Visualization of Chronological Topic Dynamics

In our scheme for visualizing salience dynamics, SalienceGraph, the constituent unit of a contextual flow is an utterance or a sentence. On the other hand, there have been many studies of visualizing chronological topic dynamics in which the constituent unit is a document. For instance, the word burst [14] is such a schemes for chronological visualization. For chronological visualization, frequency-based metrics are suitable because the constituent unit of the contextual flow, a document, is large enough for frequency-based metrics. The constituent unit in Salience graph, an utterance, is too small for frequency-based metrics, so we define the reference probability as the salience metric.

6 Conclusion

We have developed SalienceGraph, a scheme for visualizing the salience dynamics of discourse. It helps readers to browse, for example, the minutes of a long debate because it can be used to identify a particular point in the minutes. First, we designed a method for calculating a salience metric, the reference probability. We then carried out experiments to evaluate it using interview dialogues and newspaper articles in Japanese. The results showed that it can better predict subsequent references than the naive TF. They also showed that it is particularly suitable for dealing with spoken language. By integrating this method and PLSA, we developed the SalienceGraph scheme. Furthermore, we proposed applying to a discourse browsing system that satisfies Shneiderman's Visual Information-Seeking Mantra. We plan to develop such a system and use it to analyze public debate for the purpose, for example, of studying public involvement.

References

1. Jeong, H., Hatori, T., Kobayashi, K.: Discourse analysis of public debates: A corpus-based approach. In: Proceedings of 2007 IEEE International Conference on Systems, Man and Cybernetics (SMC 2007), pp. 1782–1793 (2007)
2. Shneiderman, B.: Designing the User Interface: Strategies for Effective Human-Computer Interaction, 3rd edn. Pearson Addison Wesley (1998)
3. Hasida, K.: Global Document Annotation (GDA), <http://i-content.org/GDA/>
4. Kudo, T., Matsumoto, Y.: Japanese dependency analysis using cascaded chunking. In: Proceeding of the 6th conference on Natural language learning (CoNLL-2002, COLING 2002 Post-Conference Workshops), pp. 1–7 (2002)
5. Grosz, B., Joshi, A., Weinstein, S.: Centering: A Framework for Modeling the Local Coherence of Discourse. Computational Linguistics 21(2), 203–226 (1995)
6. Hasida, K., Shiramatsu, S., Komatani, K., Ogata, T., Okuno, H.G.: Meaning games. In: Satoh, K., Inokuchi, A., Nagao, K., Kawamura, T. (eds.) JSAI 2007. LNCS (LNAI), vol. 4914, pp. 228–241. Springer, Heidelberg (2008)
7. Murdock, B.: The Serial Position Effect in Free Recall. Journal of Experimental Psychology 64, 482–488 (1962)
8. Maekawa, K.: Corpus of Spontaneous Japanese: Its Design and Evaluation. In: Proceedings of the ISCA & SSPR 2003, pp. 7–12 (2003)

9. GSK (Gengo Shigen Kyokai): Linguistic resource catalogue (in Japanese),
<http://www.gsk.or.jp/catalog.html>
10. YodoRiver-Watershed-Committee: Minute of the Debate Session among Citizens and Committee Members (Nyu Dam) (2005) (in Japanese),
<http://www.yodoriver.org/kaigi/biwa/17.html#ikenkoukan>
11. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3, 993–1022 (2003)
12. Sadamitsu, K., Mishina, T., Yamamoto, M.: Topic-based language models using dirichlet mixtures. *Systems and Computers in Japan* 38(12), 76–85 (2007)
13. Mochihashi, D., Matsumoto, Y.: Context as filtering. In: *Advances in Neural Information Processing Systems (NIPS 2005)*, vol. 18, pp. 907–914 (2006)
14. Kleinberg, J.: Bursty and hierarchical structure in streams. In: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and DataMining*, pp. 1–25 (2002)

Learning Discriminative Sequence Models from Partially Labelled Data for Activity Recognition

Tran The Truyen¹, Hung H. Bui^{2,*}, Dinh Q. Phung¹, and Svetha Venkatesh¹

¹ Department of Computing, Curtin University of Technology
GPO Box U1987 Perth, Western Australia 6845, Australia
thetruyen.tran@postgrad.curtin.edu.au,
{d.phung,s.venkatesh}@curtin.edu.au

² Artificial Intelligence Center, SRI International
333 Ravenswood Avenue, Menlo Park, CA 94025-3493, USA
bui@ai.sri.com

Abstract. Recognising daily activity patterns of people from low-level sensory data is an important problem. Traditional approaches typically rely on generative models such as the hidden Markov models and training on fully labelled data. While activity data can be readily acquired from pervasive sensors, e.g. in smart environments, providing manual labels to support fully supervised learning is often expensive. In this paper, we propose a new approach based on partially-supervised training of discriminative sequence models such as the conditional random field (CRF) and the maximum entropy Markov model (MEMM). We show that the approach can reduce labelling effort, and at the same time, provides us with the flexibility and accuracy of the discriminative framework. Our experimental results in the video surveillance domain illustrate that these models can perform better than their generative counterpart (i.e. the partially hidden Markov model), even when a substantial amount of labels are unavailable.

Keywords: activity recognition, discriminative models, partially labelled data, indoor video surveillance, conditional random fields, maximum entropy Markov models.

1 Introduction

An important task in human activity recognition from low-level noisy sensory data is segmenting the data streams and labeling them with meaningful sub-activities. The labels can then be used to facilitate data indexing and organisation, to recognise higher levels of semantics, and to provide useful context for intelligent assistive agents. To handle the uncertainty inherent in the data, current approaches to activity recognition typically employ probabilistic models

* Hung Bui is supported by the Defense Advanced Research Projects Agency (DARPA), through the Department of Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010.

such as the hidden Markov models (HMMs) and variants [1,2,7]. These models are essentially generative, i.e. they model the relation between the activity sequence \mathbf{x} and the observable data stream \mathbf{o} via the joint distribution $P(\mathbf{x}, \mathbf{o})$. However, it is often difficult to capture complex dependencies in the observation sequence \mathbf{o} , as typically, simplifying assumptions need to be made so that the conditional distribution $P(\mathbf{o}|\mathbf{x})$ is tractable. This limits the choice of features that one can use to encode multiple data streams. In addition, as we are often interested in finding the most probable activity sequence $\mathbf{x}^* = \arg \max_{\mathbf{x}} P(\mathbf{x}|\mathbf{o})$, it is more natural to model $P(\mathbf{x}|\mathbf{o})$ directly.

Thus the discriminative model $P(\mathbf{x}|\mathbf{o})$ is more suitable to specify how an activity \mathbf{x} would evolve *given* that we already observe a sequence of observations \mathbf{o} . With appropriate use of contextual information, the discriminative models can represent arbitrary, dynamic long-range interdependencies which are highly desirable for segmentation tasks.

Moreover, whilst capturing unlabeled sensor data for training is cheap, obtaining labels in a fully supervised setting often requires expert knowledge and is time consuming. In many cases we are certain about some particular labels, for example, in surveillance data, when a person enters a room or steps on a pressure mat. Other labels (e.g. other activities that occur inside the room) are left unknown. Therefore, it is more desirable to employ the partially-supervised approach in that some labels are missing in the training data. Specifically, we consider two recent discriminative models, namely, the undirected Conditional Random Fields (CRFs) [3], (Figure 1(b)) and the directed Maximum Entropy Markov Models (MEMMs) [5] (Figure 1(a)). As the original models require full labels, we provide a treatment of incomplete data for the CRFs and the MEMMs. The treatment mainly contributes to the main novelty of this paper despite the fact that there have been recent attempts to apply discriminative models for activity recognition, [4,9,10,6,11]. Note that the work in [6] also investigates hidden variables in modelling activity, this is for discovering latent aspects rather than for reducing labelling effort.

We provide experimental results in the video surveillance domain where we compare the performance of the proposed models and the equivalent partially hidden Markov models (PHMMs) [8] (Figure 1(c)) in learning and segmenting human indoor movement patterns. Out of three data sets studied, a common behaviour is that the HMM is outperformed by the discriminative counterparts even when a large portion of labels are missing. Providing contextual features for the models increases the performance significantly.

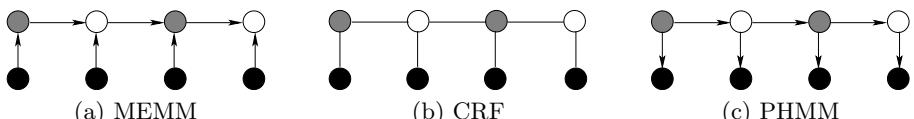


Fig. 1. (a,b): The partially labelled discriminative models, and (c): partially hidden Markov models. Filled circles and bars are data observations, empty circles are hidden labels, shaded circles are the visible labels.

The remainder of the paper is organised as follows. Section 2 provides background on CRFs and MEMMs. Section 3 describes learning discriminative models under missing labels. The paper then describes implementation and experiments and presents results in Section 4. The final section summarises major findings and further work.

2 Background

This section briefly reviews the MEMMs and the CRFs for sequence modelling. Given a data sequence \mathbf{o} of length T , the MEMMs define the conditional distribution of the activity labelling \mathbf{x} as follows

$$P(\mathbf{x}|\mathbf{o}) = P(\mathbf{x}_1|\mathbf{o}) \prod_{t=2}^T P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{o}), \text{ where,} \quad (1)$$

$$P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{o}) = \frac{1}{Z(\mathbf{o}, \mathbf{x}_{t-1})} \exp(\mathbf{w}^\top \mathbf{f}(\mathbf{o}, \mathbf{x}_{t-1}, \mathbf{x}_t)), \quad (2)$$

where $Z(\mathbf{o}, \mathbf{x}_{t-1}) = \sum_{\mathbf{x}_t} \exp(\mathbf{w}^\top \mathbf{f}(\mathbf{o}, \mathbf{x}_{t-1}, \mathbf{x}_t))$. The functions $\mathbf{f}(\mathbf{o}, \mathbf{x}_{t-1}, \mathbf{x}_t)$ are the features that capture the statistics of the observational data and the activities and their transition at time t . The parameters \mathbf{w} are the weights associated with the features and are estimated through training.

Thus, a MEMM is a directed Markov chain conditioned on the observational data \mathbf{o} . In supervised training, all activity labels $\{\mathbf{x}_t\}_{t=1}^T$ are given, so only local classifiers $P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{o})$ are learnt. During inference time, however, since no history labels are given for those local classifiers, Viterbi decoding must be used for simultaneous labelling. Since learning is conditioned on the previous labels, if the previous labels allows only limited transition to the current labels, a phenomenon known as *label-bias* will occur.

The CRFs, on the other hand, do not suffer from this drawback as they model the activity sequence entirely

$$P(\mathbf{x}|\mathbf{o}) = \frac{1}{Z(\mathbf{o})} \prod_{t=2}^T \exp(\mathbf{w}^\top \mathbf{f}(\mathbf{o}, \mathbf{x}_{t-1}, \mathbf{x}_t)), \quad (3)$$

where $Z(\mathbf{o}) = \sum_{\mathbf{x}} \prod_{t=2}^T \exp(\mathbf{w}^\top \mathbf{f}(\mathbf{o}, \mathbf{x}_{t-1}, \mathbf{x}_t))$. Since the computation of $Z(\mathbf{o})$ has the standard sum-product form, we can use dynamic programming at the cost of $\mathcal{O}(T)$ time. Thus, a CRF is a undirected Markov chain conditioned on the observational data \mathbf{o} .

Fully supervised learning in the CRFs and MEMMs typically maximises the conditional log-likelihood¹ $\mathcal{L}(\mathbf{w}) = \log P(\mathbf{x}|\mathbf{o})$.

¹ For multiple iid data instances, we should write $\mathcal{L}(\mathbf{w}) = \sum_x \tilde{P}(\mathbf{o}) \log P(\mathbf{x}|\mathbf{o})$ where $\tilde{P}(\mathbf{o})$ is the empirical distribution of training data, but we drop this notation for clarity.

3 Learning Discriminative Models from Partially Labelled Data

In our partially labelled discriminative models, the label sequence \mathbf{x} consists of a visible component \mathbf{v} (e.g. labels that are provided manually, or are acquired automatically by reliable sensors) and a hidden part \mathbf{h} (labels that are left unspecified or those we are unsure), that is $\mathbf{x} = (\mathbf{v}, \mathbf{h})$. The joint distribution of all visible variables \mathbf{v} is therefore given as

$$P(\mathbf{v}|\mathbf{o}) = \sum_{\mathbf{h}} P(\mathbf{v}, \mathbf{h}|\mathbf{o}) = \sum_{\mathbf{h}} P(\mathbf{x}|\mathbf{o}) . \quad (4)$$

To learn the model parameters that are best explained by the data, we maximise the penalised log-likelihood

$$\Lambda(\mathbf{w}) = \mathcal{L}(\mathbf{w}) - \frac{1}{2\sigma^2} \|\mathbf{w}\|^2 ,$$

where $\mathcal{L}(\mathbf{w}) = \log P(\mathbf{v}|\mathbf{o})$. The regularisation term is needed to avoid over-fitting when only limited data is available for training. For simplicity, the parameter σ is shared among all dimensions and is selected experimentally.

As with incomplete data, an alternative to maximise the log-likelihood is using the EM algorithm whose Expectation (E-step) at step j is to calculate the quantity

$$Q(\mathbf{w}^j, \mathbf{w}) = \sum_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}, \mathbf{o}; \mathbf{w}^j) \log P(\mathbf{h}, \mathbf{v}|\mathbf{o}) , \quad (5)$$

and the Maximisation (M-step) maximises the concave lower bound of the log-likelihood $Q(\mathbf{w}^j, \mathbf{w}) - \frac{1}{2\sigma^2} \|\mathbf{w}\|^2$ with respect to \mathbf{w} . Unlike Bayesian networks, the log-linear models do not yield closed form solutions in the M-step. However, as the function $Q(\mathbf{w}^j, \mathbf{w})$ is concave, it is still advantageous to optimise with efficient Newton-like algorithms.

3.1 Learning MEMMs

Directed models like the MEMMs are important in activity modeling because they naturally encode the state transitions given the observations. As we are free to encode arbitrary information exacted from the whole sequence \mathbf{o} to the local distribution, we use a sliding window Ω_t of size s centred at the current time t to capture the local context of the observation. The joint incomplete distribution is therefore

$$P(\mathbf{v}|\mathbf{o}) = \sum_{\mathbf{h}} P(\mathbf{x}_1|\Omega_1) \prod_{t=2}^T P(\mathbf{x}_t|\Omega_t, \mathbf{x}_{t-1}) . \quad (6)$$

Since this is a standard sum-product problem, dynamic programming can be used to solve in $\mathcal{O}(T)$ time.

In learning of MEMMs using EM, the E-step is to calculate

$$Q(\mathbf{w}^j, \mathbf{w}) = \sum_t \sum_{\mathbf{h}_{t-1}} P(\mathbf{h}_{t-1} | \mathbf{v}, \Omega_t^j) \sum_{\mathbf{h}_t} P(\mathbf{h}_t | \mathbf{h}_{t-1}, \Omega_t^j) \log P(\mathbf{h}_t | \mathbf{h}_{t-1}, \Omega_t) . \quad (7)$$

and the M-step is to solve the zeroing gradient equation

$$\nabla Q(\mathbf{w}^j, \mathbf{w}) = \sum_t \sum_{\mathbf{h}_{t-1}} P(\mathbf{h}_{t-1} | \mathbf{v}, \Omega_t^j) \left\{ \sum_{\mathbf{h}_t} P(\mathbf{h}_t | \mathbf{h}_{t-1}, \Omega_t^j) \mathbf{f}(\mathbf{h}_{t-1}, \mathbf{h}_t, \Omega_t) - \sum_{\mathbf{x}_t} P(\mathbf{x}_t | \mathbf{h}_{t-1}, \Omega_t) \mathbf{f}(\mathbf{h}_{t-1}, \mathbf{x}_t, \Omega_t) \right\} .$$

Computation of the EM reduces to that of marginals and state transition probabilities, which can be carried out efficiently in the Markov chain framework using dynamic programming.

3.2 Learning CRFs

From Eq. 3, we have

$$P(\mathbf{v} | \mathbf{o}) = \frac{1}{Z(\mathbf{o})} \sum_{\mathbf{h}} \exp(\mathbf{w}^\top \mathbf{f}(\mathbf{o}, \mathbf{x}_{t-1}, \mathbf{x}_t)) . \quad (8)$$

In this case, the complexity of computing $P(\mathbf{v} | \mathbf{o})$ is the same as that of computing the partition function $Z(\mathbf{o})$ up to a constant.

For the partially labelled CRFs, the gradient of incomplete likelihood reads

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}_k} = \sum_t \left(\sum_{\mathbf{h}_{t-1}, \mathbf{h}_t} P(\mathbf{h}_{t-1}, \mathbf{h}_t | \mathbf{v}, \mathbf{o}) f_k(\mathbf{h}_{t-1}, \mathbf{h}_t, \mathbf{v}, \mathbf{o}) - \sum_{\mathbf{x}_{t-1}, \mathbf{x}_t} P(\mathbf{x}_{t-1}, \mathbf{x}_t | \mathbf{o}) f_k(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{o}) \right) . \quad (9)$$

Zeroing the gradient does not yield an analytical solution, so typically iterative numerical methods such as conjugate gradient and Newton methods are needed. The gradient of the lower bound in the EM framework of (5) is similar to (9), except that the pairwise marginals $P(\mathbf{h}_{t-1}, \mathbf{h}_t | \mathbf{v}, \mathbf{o})$ are now replaced by the marginals of the previous EM iteration $P(\mathbf{h}_{t-1}, \mathbf{h}_t | \mathbf{v}, \mathbf{o}^j)$. The pairwise marginals $P(\mathbf{x}_{t-1}, \mathbf{x}_t | \mathbf{o})$ can be computed easily using a forward pass and a backward pass in the standard message passing scheme on the chain.

3.3 Comparison with the PHMMs

The main difference between the models described in this section (Figure 1(a,b)) and the PHMMs [8] (Figure 1(c)) is the conditional distribution $P(\mathbf{x} | \mathbf{o})$ in discriminative models compared to the joint distribution $P(\mathbf{x}, \mathbf{o})$ in the PHMMs.

The data distribution of $P(\mathbf{o})$ and how \mathbf{o} is generated are not of concern in the discriminative models. In the PHMMs, on the contrary, the observation point \mathbf{o}_t is presumably generated by the parent label node \mathbf{x}_t , so care must be taken to ensure proper conditional independence among $\{\mathbf{o}_t\}_{t=1}^T$. This difference has an implication that, while the discriminative models may be good to encode the output labels directly with arbitrary information extracted from the whole observation sequence \mathbf{o} , the PHMMs better represent \mathbf{o} when little information is associated with \mathbf{x} . For example, when \mathbf{x} is totally missing, $P(\mathbf{o}) = \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o})$ is still modeled in the PHMMs and provides useful information.

4 Experiments and Results

Our task is to infer the activity patterns of a person (the actor) in a video surveillance scene. The observation data is provided by static cameras while the labels, which are activities such as ‘*go-from-A-to-B*’ during the time interval $[t_a, t_b]$ (see Table 1), are recognised by the trained models.

4.1 Setup and Data

The surveillance environment is a $4 \times 6m^2$ dining room and kitchen (Figure 2). Two static cameras are installed to capture the video of the actor making some meals. There are six landmarks which the person can visit during the meals: door, TV chair, fridge, stove, cupboard, and dining chair.

We study three scenarios corresponding to the person making a short meal (denoted by SHORT_MEAL), having a snack (HAVE_SNACK), and making a normal meal (NORMAL_MEAL). Each scenario comprises of a number of primitive activities as listed in Table 1. The association between scenarios and their primitive activities are: SHORT_MEAL = {1,2,3,4,11}, HAVE_SNACK = {2,5,6,7,8}, and NORMAL_MEAL = {1,2,4,9,10,11,12}. The SHORT_MEAL data set has 12 training and 22 testing video sequences; and each of the HAVE_SNACK

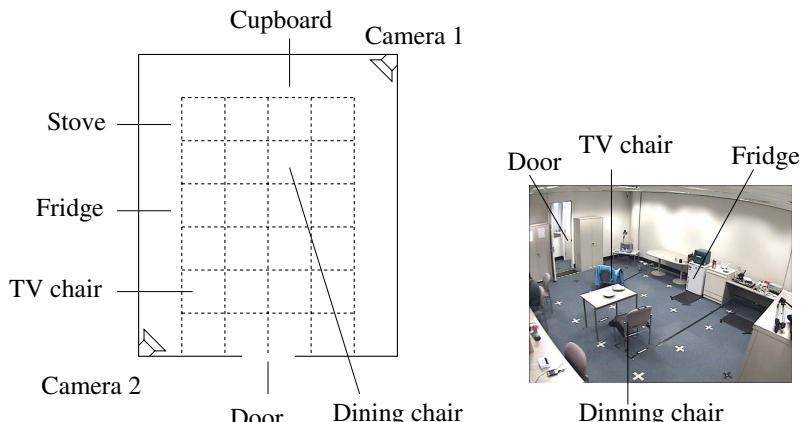


Fig. 2. The environment and scene viewed from one of the two cameras

Table 1. The primitive activities (the labels)

Activity	Landmarks	Activity	Landmarks
1	Door→Cupboard	7	Fridge→TV chair
2	Cupboard→Fridge	8	TV chair→Door
3	Fridge→Dining chair	9	Fridge→Stove
4	Dining chair→Door	10	Stove→Dining chair
5	Door→TV chair	11	Fridge→Door
6	TV chair→Cupboard	12	Dining chair→Fridge

and NORMAL_MEAL data sets consists of 15 training and 11 testing video sequences. For each raw video sequence captured, we use a background subtraction algorithm to extract a corresponding discrete sequence of coordinates of the person based on the person’s bounding box. The training sequences are partially labeled, indicated by the portion of missing labels ρ . The testing sequences provide the ground-truth for the algorithms. The sequence length ranges from $T = 20 - 60$ and the number of labels per sequence is allowed to vary as $T*(1-\rho)$ where $\rho \in [0, 100\%]$.

We apply standard evaluation metrics such as precision P , recall R , and the F1 score given as $F1 = 2 * P * R / (P + R)$ on a per-token basis.

4.2 Feature Design and Contextual Extraction

Features are crucial components of the model as they tie raw observation data with semantic outputs (i.e. the labels). The features need to be discriminative enough to be useful, and at the same time, they should be as simple and intuitive as possible to reduce manual labour. The current raw data extracted from the video contains only (X, Y) coordinates. From each coordinate sequences, at each time slice t , we extract a vector of five elements from the observation sequence $g(\mathbf{o}, t) = (X_t, Y_t, u_{X_t}, u_{Y_t}, s_t = \sqrt{u_{X_t}^2 + u_{Y_t}^2})$, which correspond to the (X, Y) coordinates, the X & Y velocities, and the speed, respectively. Since the extracted coordinates are fairly noisy, we use the average velocity measurement within a time interval of small width w , i.e. $u_{X_t} = (X_{t+w/2} - X_{t-w/2})/w$. Typically, these observation-based features are real numbers and are normalised so that they have a similar scale.

We decompose the feature set $\{f_k(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{o})\}$ into two subsets: the *state-observation* features $f_{l,m,\epsilon}(\mathbf{o}, \mathbf{x}_t) := \mathbb{I}[\mathbf{x}_t = l]\mathbf{h}_m(\mathbf{o}, t, \epsilon)$ and the *state-transition* features $f_{l_1,l_2}(\mathbf{x}_{t-1}, \mathbf{x}_t) := \mathbb{I}[\mathbf{x}_{t-1} = l_1]\mathbb{I}[\mathbf{x}_t = l_2]$, where $m = 1..5$ and $\mathbf{h}_m(\mathbf{o}, t, \epsilon) = g_m(\mathbf{o}, t + \epsilon)$ with $\epsilon = -s_1, ..0, ..s_2$ for some positive integers s_1, s_2 . The state-observation features in thus incorporate neighbouring observation points within a sliding window of width $s = s_1 + s_2 + 1$.

To have a rough idea of how the observation context influences the performance of the models, we try different window sizes s (see Equation (1)). The experiments show that incorporating the context of observation sequences does help to improve the performance significantly (see Figure 3). We did not try exhaustive searches

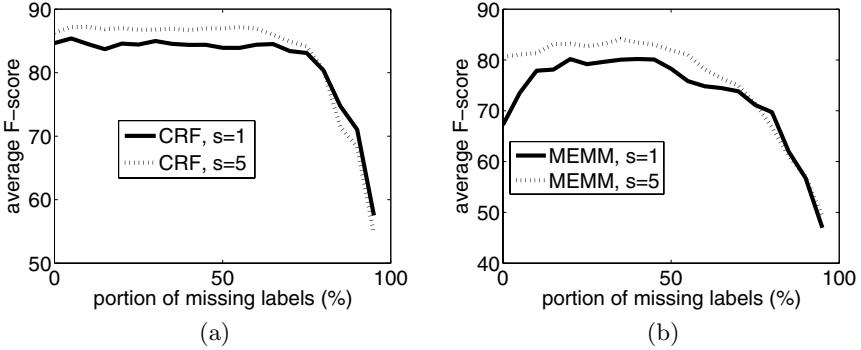


Fig. 3. The role of context (SHORT_MEAL), s : the window size to extract observation data. (a) CRFs, (b) MEMMs. In all figures, the x-axis: the portion of missing labels (%) and the y-axis: the averaged F-score (%) over all states and over 10 repetitions.

for the best context size, nor did we implement any feature selection mechanisms. As the number of features scales linearly with the context size as $K = 5s|Y| + |Y|^2$, where s can be any integer between 1 and T , where T is the sequence length, clearly a feature selection algorithm is needed when we want to capture long range correlation. For the practical purposes of this paper, we choose $s = 5$ for both CRFs and MEMMs. Thus in our experiments, CRFs and MEMMs share the same feature set, making the comparison between the two models consistent.

4.3 Performance of Models

To evaluate the performance of discriminative models against the equivalent generative counterparts, we implement the PHMMs (Figure 1(c)). The features extracted from the sensor data for the PHMMs include the discretised position and velocity. These features are different from those used in discriminative models in that discriminative features can be continuous. Thus the feature set used by PHMMs is different from those shared by CRFs and MEMMs. Although the difference may raise the concern about the compatibility of these models, it is precisely where discriminative models are more flexible as they have no difficulty selecting features.

To train discriminative models, we implement the non-linear conjugate gradient (CG) of Polak-Ribière and the limited memory quasi-Newton L-BFGS. After several pilot runs, we select the L-BFGS to optimise the objective function in (5) directly. In the case of MEMMs, the regularised EM algorithm is chosen together with the CG. The algorithms stop when the rate of convergence is less than 10^{-5} . The regularisation constants are empirically selected as $\sigma = 5$ in the case of CRFs, and $\sigma = 20$ in the case of MEMMs.

For the PHMMs, it is observed that the initial parameter initialisation is critical to learn the correct model. Random initialisations often result in very poor performance. This is unlike the discriminative counterparts in which all initial parameters can be trivially set to zeros (equally important).

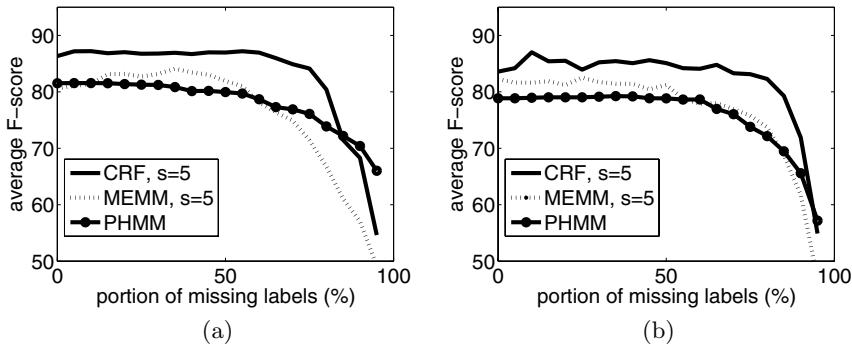


Fig. 4. Average performance of models (a: SHORT_MEAL, b: NORMAL_MEAL). x-axis: portion of missing labels (%) and y-axis: the averaged F-score (%) over all states and 10 repetitions.

Overall in our experiments (Figure 4) the generative PHMMs are outperformed by the discriminative counterparts in all cases given sufficient labels. This clearly matches the theoretical differences between these models in that when there are enough labels, richer information can be extracted in the discriminative framework, i.e. modeling $P(\mathbf{x}|\mathbf{o})$ is more suitable. On the other hand, when only a few labels are available, the unlabeled data is important so it makes sense to model and optimise $P(\mathbf{o}, \mathbf{x})$ as in the generative framework. On all data sets, the CRFs outperform the other models. These behaviours are consistent with the results reported in [3] in the fully observed setting. MEMMs are known to suffer from the label-bias problem [3], thus their performance does not match that of CRFs, although MEMMs are better than HMMs given enough training labels. In the HAVE_SNACK data set, the performance of MEMMs is surprisingly good.

A striking fact about the globally normalised CRFs is that the performance persists until most labels are missing. This is clearly a big time and effort saving for the labeling task.

5 Conclusions and Further Work

In this work, we have presented a partially-supervised framework for activity recognition on low-level noisy data from sensors using discriminative models. We illustrated the appropriateness of the discriminative models for segmentation of surveillance video into sub-activities. As more flexible information can be encoded using feature functions, the discriminative models can perform significantly better than the equivalent generative HMMs even when a large portion of the labels are missing. CRFs appear to be a promising model as the experiments show that they consistently outperform other models in all three data sets. Although less expressive than CRFs, MEMMs are still an important class of models as they enjoy the flexibility of the discriminative framework and enable online recognition as in directed graphical models.

Our study shows that primitive and intuitive contextual features work well in the area of video surveillance. However, to obtain the optimal context and to make use of the all information embedded in the whole observation sequence, a feature selection mechanism remains to be designed in conjunction with the models and training algorithms presented in this paper.

References

1. Brand, M., Oliver, N., Pentland, A.: Coupled hidden Markov models for complex action recognition. In: Proc. CVPR, pp. 994–999 (1997)
2. Bui, H.H., Venkatesh, S., West, G.: Policy recognition in the abstract hidden Markov model. Journal of Articial Intelligence Research 17, 451–499 (2002)
3. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proc. ICML, pp. 282–289 (2001)
4. Liao, L., Fox, D., Kautz, H.: Location-Based Activity Recognition using Relational Markov Networks. In: Proc. IJCAI, pp. 773–778 (2005)
5. McCallum, A., Freitag, D., Pereira, F.: Maximum Entropy Markov models for information extraction and segmentation. In: Proc. ICML, pp. 591–598 (2000)
6. Morency, L.P., Quattoni, A., Darrell, T.: Latent-Dynamic Discriminative Models for Continuous Gesture Recognition. In: Proc. CVPR, pp. 1–8 (2007)
7. Oliver, N., Garg, A., Horvitz, E.: Layered representations for learning and inferring office activity from multiple sensory channels. CVIU 96, 163–180 (2004)
8. Scheffer, T., Wrobel, S.: Active learning of partially labelled Markov models. In: Active Learning, Database Sampling, Experimental Design: Views on Instance Selection, Workshop at ECML 2001/PKDD 2001 (2001)
9. Sminchisescu, C., Kanaujia, A., Metaxas, D.: Conditional models for contextual human motion recognition. CVIU 104, 210–220 (2006)
10. Vail, D.L., Veloso, M.M., Lafferty, J.D.: Conditional random fields for activity recognition. In: Proc. AAMAS (2007)
11. Wu, T., Lian, C., Hsu, J.Y.: Joint Recognition of Multiple Concurrent Activities using Factorial Conditional Random Fields. In: AAAI Workshop on Plan, Activity, and Intent Recognition (2007)
12. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden Markov models. In: Proc. CVPR (1992), pp. 379–385 (1992)

Feature Selection for Clustering on High Dimensional Data^{*}

Hong Zeng and Yiu-ming Cheung

Department of Computer Science, Hong Kong Baptist University,
Hong Kong SAR, China
`{hzeng, ymc}@comp.hkbu.edu.hk`

Abstract. This paper addresses the problem of feature selection for the high dimensional data clustering. This is a difficult problem because the ground truth class labels that can guide the selection are unavailable in clustering. Besides, the data may have a large number of features and the irrelevant ones can ruin the clustering. In this paper, we propose a novel feature weighting scheme for a kernel based clustering criterion, in which the weight for each feature is a measure of its contribution to the clustering task. Accordingly, we give a well-defined objective function, which can be explicitly solved in an iterative way. Experimental results show the effectiveness of the proposed method.

1 Introduction

In many pattern recognition and data mining problems, e.g. the computer vision, text processing and the more recent gene data analysis, etc., the input raw data sets often have a huge number of possible explanatory variables, but there are much fewer samples available. The abundance of variables makes the classification among patterns much harder and less accurate. Under such circumstances, selecting the most discriminative or representative features of a sample inevitably becomes an important issue.

In the literature, most feature selection algorithms have been developed for supervised learning, rather than the unsupervised learning. It is believed that the unsupervised feature selection is more difficult due to the absence of class labels that can guide the search for the relevant information. Until very recently, several algorithms have been proposed to address this issue for clustering. In general, they can be categorized as *wrapper* and *filter* methods according to the evaluation criterion in searching for relevant features. For *wrapper* approaches [5,7], the quality of every candidate feature subset is assessed by investigating the performance of a specific clustering algorithm on this subset, and each candidate subset is obtained by conducting combinatorial search through the space of all feature subsets. These algorithms have shown the success on low dimensional data. Nevertheless, the size of candidate subset space is exponential increased over the number of features. As a result, their computation are laborious, particularly on the high dimensional data. In contrast, the *filter* approaches [6,12,8,4] are more efficient in dealing with the high dimensional data. Such an algorithm first evaluates the

* This work was supported by the Faculty Research Grant of HKBU under Project: FRG/07-08/II-54, and the Research Grant Council of Hong Kong SAR under Project: HKBU 210306.

features by their intrinsic properties (e.g., feature variance, similarity among features, capability of locality preserving, etc.), and then removes a number of less informative features before the clustering. In the literature, the Laplacian score [6] is considered as the state-of-art *filter* method [12]. It selects the features that can preserve the manifold locality described by the weighted nearest neighbor graph, which has a close relationship to the spectral clustering [10,11]. This method has been successfully applied to the real-world high dimensional datasets that possess the manifold characteristics. Nevertheless, it assumes that there should be much less irrelevant features in the data so as to obtain a graph characterizing the authentic similarities among data. In the presence of a large number of irrelevant features, the performance of Laplacian score may be degraded severely.

In this paper, we propose an effective feature selection approach to clustering. The proposed method assigns each feature a real-valued weight to indicate its relevance for the clustering problem, and eventually the issue of feature selection, together with the clustering, is formulated as an optimization problem. Accordingly, we give a kernel based clustering objective function, which can be optimized using an iterative algorithm. In each step of an iteration, the sub-optimization problem is convex and can be easily solved by a well-established optimization technique.

The remainder of the paper is organized as follows. Section 2 introduces the optimization formulation for the feature selection problem, whose solution is given in Section 3. Section 4 presents the extensive experiments on real-world high dimensional datasets. The concluding remarks are given in Section 5.

2 The Feature Selection in Clustering as an Optimization Problem

Before giving the feature selection scheme, we first introduce the clustering objective function we proposed in this paper.

2.1 The Clustering Objective Function

Let $X = \{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^{d \times n}$ denote the data set consisting of n samples over d -dimensional space, $S_{ij}(0 \leq S_{ij} \leq \infty)$ denote the similarity between the points \mathbf{x}_i and \mathbf{x}_j , and the similarity matrix $\mathbf{S} = [S_{ij}]_{n \times n}$ is assumed to be symmetric. An intuitive clustering objective is to seek the partition such that the summation of similarities between points in the same cluster is maximized, while that in different clusters is minimized. Such a criterion can be realized to maximize the following cost function:

$$\mathcal{Q}(\mathbb{C}) = \sum_{l=1}^k \left[\sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathbb{C}_l} \left(\frac{1}{|\mathbb{C}_l|} - \frac{1}{n} \right) S_{ij} + \sum_{\mathbf{x}_i, \mathbf{x}_j \notin \mathbb{C}_l} \left(-\frac{1}{n} \right) S_{ij} \right] \quad (1)$$

where \mathbb{C} is a possible partition, k is the number of clusters which is assumed known, \mathbb{C}_l is the set of points contained in the l -th cluster ($1 \leq l \leq k$), and the number of points in the l -th cluster is denoted by $|\mathbb{C}_l|$ ($|\mathbb{C}_l| \leq n$). The coefficients in front of the similarity items are to defy the effect of different sizes of clusters. Equation (1) can be expressed in a more compact form with matrix operation:

$$\mathcal{Q}(\mathbf{G}) = \sum_{i,j=1}^n S_{ij} \tilde{G}_{ij} = \text{trace}(\mathbf{S}\tilde{\mathbf{G}}) \quad (2)$$

where $\tilde{\mathbf{G}} = \mathbf{\Pi}_n \mathbf{G} \mathbf{\Pi}_n$, $\mathbf{\Pi}_n \in \mathbb{R}^{n \times n}$ is the centering matrix defined as $\mathbf{\Pi}_n = \mathbf{I}_n - \frac{1}{n} \mathbf{e}_n \mathbf{e}_n^T$, and \mathbf{e}_n is a vector of all ones of size n . $\mathbf{G} \in \mathbb{R}^{n \times n}$ is a matrix whose entry is defined as: $G_{ij} = \frac{1}{|\mathcal{C}_l|}$, if $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C}_l$, and zero otherwise. If we define the hard cluster indicator matrix $\mathbf{L} \in \mathbb{R}^{n \times k}$ as: $L_{il} = |\mathcal{C}_l|^{-\frac{1}{2}}$, if $\mathbf{x}_i \in \mathcal{C}_l$ and zero otherwise. It is easy to verify that the cluster indicator matrix satisfies: $\mathbf{L}\mathbf{L}^T = \mathbf{G}$, $\mathbf{L}^T\mathbf{L} = \mathbf{I}_k$, where $\mathbf{I}_k \in \mathbb{R}^{k \times k}$ is the unit matrix. Then Equation (2) can be expressed as:

$$\mathcal{Q}(\mathbf{L}) = \text{trace}(\mathbf{S}\mathbf{\Pi}_n \mathbf{L}\mathbf{L}^T \mathbf{\Pi}_n) = \text{trace}(\mathbf{L}^T \mathbf{\Pi}_n \mathbf{S}\mathbf{\Pi}_n \mathbf{L}). \quad (3)$$

By the spectral relaxation [1], we allow L_{ij} to take a continuous value, subject to the constraint $\mathbf{L}^T\mathbf{L} = \mathbf{I}_k$ so as to turn it into a tractable continuous optimization problem. Hence, the clustering criterion can be formulated as:

$$\max_{\mathbf{L}^T\mathbf{L}=\mathbf{I}_k} \text{trace}(\mathbf{L}^T \mathbf{\Pi}_n \mathbf{S}\mathbf{\Pi}_n \mathbf{L}). \quad (4)$$

2.2 The Weighting Scheme to Select Features

The matrix \mathbf{S} is not necessarily fixed. In fact, we select the relevant features to enhance the similarity matrix for maximizing the criterion in Equation (4), i.e.

$$\max_{\mathbf{S}, \mathbf{L}} \mathcal{Q}(\mathbf{L}, \mathbf{S}) = \text{trace}(\mathbf{L}^T \mathbf{\Pi}_n \mathbf{S}\mathbf{\Pi}_n \mathbf{L}) \text{ s.t. } \mathbf{L}^T\mathbf{L} = \mathbf{I}_k. \quad (5)$$

Suppose the RBF kernel function is adopted as the similarity, i.e., $S_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\sum_{p=1}^d (\mathbf{x}_i^{(p)} - \mathbf{x}_j^{(p)})^2}{t})$, where $\mathbf{x}_i^{(p)}$ is the value of the p -th feature for the point \mathbf{x}_i . Let $w_p \in \{0, 1\}$ ($1 \leq p \leq d$) be the relevance indicator value associated with the p -th feature, i.e., $w_p = 1$ if the p -th feature is selected to form the relevant feature subset and 0 otherwise. Therefore, a natural feature selection scheme is to use a modified similarity matrix, denoted as \mathbf{S}^w , whose (i, j) -th element is defined as:

$$S_{ij}^w = K(\mathbf{w} \circ \mathbf{x}_i, \mathbf{w} \circ \mathbf{x}_j) = e^{-\frac{\sum_{p=1}^d w_p (\mathbf{x}_i^{(p)} - \mathbf{x}_j^{(p)})^2}{t}}, \quad (6)$$

where \circ denotes the element-wise multiplication. Since maximization of $\mathcal{Q}(\mathbf{L}, \mathbf{S}^w)$ for all possible feature subsets is infeasible for high dimensional data, we relax the indicator w_p to real-valued nonnegative weight (i.e. $w_p \geq 0$), and a large value of w_p will indicate that the p -th feature is more important to the similarity formation. It is observed that substituting the modified similarity matrix \mathbf{S}^w into (5) will lead to a nonlinear optimization problem with respect to \mathbf{w} , which is very difficult to solve, due to the nonlinearity introduced by the RBF kernel function. In order to overcome this difficulty, we propose a simple but effective weighting scheme:

$$S_{ij}^w = \sum_{p=1}^d w_p K(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(p)}) = \sum_{p=1}^d w_p e^{-\frac{(\mathbf{x}_i^{(p)} - \mathbf{x}_j^{(p)})^2}{t}} = \sum_{p=1}^d w_p K_p(\mathbf{x}_i, \mathbf{x}_j), \quad (7)$$

where we define $K_p(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(p)})$ as the (i, j) -th element of the kernel matrix \mathbf{K}_p that is constructed by using only the p -th feature of the data points. Furthermore, we normalize \mathbf{K}_p to $\mathbf{K}_p \leftarrow \mathbf{D}_p^{-\frac{1}{2}} \mathbf{K}_p \mathbf{D}_p^{-\frac{1}{2}}$, where \mathbf{D}_p is a diagonal matrix with the row sum of \mathbf{K}_p in the diagonal, and the operation “ $\mathbf{A} \leftarrow \mathbf{B}$ ” means that the value of \mathbf{B} is assigned to \mathbf{A} . The normalized similarity can be interpreted as the probability of $\mathbf{x}_i^{(p)}$ (or $\mathbf{x}_j^{(p)}$) being close to $\mathbf{x}_j^{(p)}$ (or $\mathbf{x}_i^{(p)}$). For fixed \mathbf{w} , the aggregated and normalized \mathbf{K}_p form a combination $\mathbf{S}^w = \sum_{p=1}^d w_p \mathbf{K}_p$. If $w_p = 0$, this implies that \mathbf{S}^w will not depend on the original p -th feature. Obviously, \mathbf{S}^w is still symmetric and has nonnegative elements. Subsequently, maximizing the criterion in (5) finally becomes the following optimization problem:

$$\begin{aligned} \max_{\mathbf{w}, \mathbf{L}} \mathcal{Q}(\mathbf{L}, \mathbf{w}) &= \max_{\mathbf{w}, \mathbf{L}} \text{trace} \left[\mathbf{L}^T \boldsymbol{\Pi}_n \left(\sum_p w_p \mathbf{K}_p \right) \boldsymbol{\Pi}_n \mathbf{L} \right] \\ \text{s.t. } w_p &\geq 0, \|\mathbf{w}\|^2 = 1, \quad \mathbf{L}^T \mathbf{L} = \mathbf{I}_k, \end{aligned} \quad (8)$$

where the constraint $\|\mathbf{w}\|^2 = 1$ prevents the maximization from increasing without bound.

3 The Solution to the Optimization Problem

The objective function in Equation (8) is not convex. However, if one of the two components (\mathbf{w} and \mathbf{L}) is fixed, the objective function will be convex in terms of the other component. Subsequently, the optimization problem becomes easy to solve, which will enable us to solve the problem by updating \mathbf{w} and \mathbf{L} iteratively to find a (local) optimal solution for (8).

3.1 Calculation of \mathbf{L} for a Given \mathbf{w}

Given a weight vector \mathbf{w} , the maximization problem specified in Equation (8) reduces to the following trace maximization problem:

$$\max_{\mathbf{L}} \text{trace}(\mathbf{L}^T \tilde{\mathbf{K}} \mathbf{L}) \quad \text{s.t. } \mathbf{L}^T \mathbf{L} = \mathbf{I}_k. \quad (9)$$

where $\tilde{\mathbf{K}}$ is defined as $\tilde{\mathbf{K}} = \boldsymbol{\Pi}_n (\sum_p w_p \mathbf{K}_p) \boldsymbol{\Pi}_n$. According to the Ky Fan theorem [2], an optimal solution for \mathbf{L} is given by the k eigenvectors of $\tilde{\mathbf{K}}$ corresponding to the k largest eigenvalues, where k is the number of clusters.

3.2 Calculation of \mathbf{w} as Given \mathbf{L}

Given a cluster indicator matrix \mathbf{L} , the maximization problem specified in Equation (8) reduces to:

$$\max_{\mathbf{w}} \sum_p w_p \text{trace}(\mathbf{L}^T \boldsymbol{\Pi}_n \mathbf{K}_p \boldsymbol{\Pi}_n \mathbf{L}) \quad \text{s.t. } w_p \geq 0, \|\mathbf{w}\|^2 = 1. \quad (10)$$

Let $z_p = \text{trace}(\mathbf{L}^T \mathbf{\Pi}_n \mathbf{K}_p \mathbf{\Pi}_n \mathbf{L})$. Intuitively, it can be interpreted as the contribution of the p -th feature to the clustering. Then (10) can be simplified as:

$$\max_{\mathbf{w}} \mathbf{w}^T \mathbf{z}, \quad s.t. \quad w_p \geq 0, \|\mathbf{w}\|^2 = 1. \quad (11)$$

Applying the Lagrangian method to the optimization problem (11), we can obtain its analytical solution $\mathbf{w} = (\mathbf{z})^+ / \|(\mathbf{z})^+\|_2$, where $(z_p)^+ = \max(z_p, 0)$. Fortunately, z_p is always nonnegative because $z_p = \text{trace}(\mathbf{L}^T \mathbf{\Pi}_n \mathbf{K}_p \mathbf{\Pi}_n \mathbf{L}) = \sum_{l=1}^k (\mathbf{\Pi}_n \mathbf{L}_l)^T \mathbf{K}_p (\mathbf{\Pi}_n \mathbf{L}_l) \geq 0$, which follows from the positive semi-definite property of \mathbf{K}_p . Therefore, given \mathbf{L} , we can obtain a global maximizer for (10), i.e., $\mathbf{w} = \mathbf{z} / \|\mathbf{z}\|_2$. Namely, the weight for each feature is proportional to its contribution to the overall clustering quality.

3.3 The Main Algorithm

Based on the discussion described above, we propose to develop an iterative algorithm. The pseudo-code of the main algorithm is given in Algorithm 1. The final discrete

```

input :  $\mathbf{X}, k, \epsilon$ 
output:  $\mathbf{L}, \mathbf{w}$ 

1 Construct  $\mathbf{K}_p (p = 1, \dots, d)$  with RBF kernel function using only the  $p$ -th row of  $\mathbf{X}$ , and
normalize each kernel matrix as:  $\mathbf{K}_p \leftarrow \mathbf{D}_p^{-\frac{1}{2}} \mathbf{K}_p \mathbf{D}_p^{-\frac{1}{2}}$ , where  $\mathbf{D}_p$  is a diagonal matrix with
the row sum of  $\mathbf{K}_p$  in the diagonal;
2 Set the initial weight vector  $\mathbf{w}$  to  $\mathbf{e}_d / \|\mathbf{e}_d\|_2$ ;
3 while the relative change of the objective function value  $\geq \epsilon$  do
4     Update  $\mathbf{L}$  as in Section 3.1;
5     Update  $\mathbf{w}$  as in Section 3.2;
6     Record the objective function value in (8);
7 end
8 return  $\mathbf{L}, \mathbf{w}$ ;

```

Algorithm 1. The main algorithm for the integrated feature selection in clustering

clustering result can be obtained by applying k -means on the rows of the relaxed cluster indicator matrix \mathbf{L} as in [10]. The convergence of the proposed algorithm is guaranteed, as shown in Theorem 1.

Theorem 1. *The proposed algorithm always converges.*

Proof. Given an arbitrary weight vector \mathbf{w}^* that satisfies the required constraints, we can obtain that:

$$\begin{aligned}
\max_{\mathbf{L}} \mathcal{Q}(\mathbf{L}, \mathbf{w}^*) &= \max_{\mathbf{L}} \text{trace} \left[\mathbf{L}^T \mathbf{\Pi}_n \left(\sum_p w_p^* \mathbf{K}_p \right) \mathbf{\Pi}_n \mathbf{L} \right] = \lambda_1 + \dots + \lambda_k \\
&\leq \lambda_1 + \dots + \lambda_n = \text{trace} \left[\mathbf{\Pi}_n \left(\sum_p w_p^* \mathbf{K}_p \right) \mathbf{\Pi}_n \right] \\
&= \sum_p w_p^* \text{trace}(\mathbf{\Pi}_n \mathbf{K}_p \mathbf{\Pi}_n) \leq \max_{w_p^* \geq 0, \|\mathbf{w}^*\|^2=1} \mathbf{w}^{*T} \nu = \|\nu\|_2,
\end{aligned} \quad (12)$$

where $\nu_p = \text{trace}(\boldsymbol{\Pi}_n \mathbf{K}_p \boldsymbol{\Pi}_n)$ is fixed. According to (12), the maximum of (8) is always upper bounded by a fixed finite value. Since Step 4 and 5 of the main algorithm optimize the same objective function in (8), its value is non-decreasing, the algorithm will always converge.

In the implementation, we set the threshold ϵ at 0.0005 for checking the convergence. We observe from our experiments that the proposed algorithm converges in less than 10 iterations, and typically within 3 to 4 iterations.

4 Experiments

We conducted several experiments to demonstrate the effectiveness of the proposed algorithm. Five benchmark data sets were used in our experiments, and their characteristics are summarized in Table 1. On each data set, we investigated whether our integrated feature selection and clustering algorithm (denoted as *integrated*) could improve the conventional spectral clustering algorithm, normalized cut (Ncut)¹ [11], which is equivalent to optimize the same clustering criterion in (4) (given the row-sum normalized S matrix is fixed), but does not consider the feature selection issue (denoted as *nofs+Ncut*). Furthermore, we compared our feature selection method with the state-of-art unsupervised one, the Laplacian score [6], whose starting point is to seek the feature that can preserve the locality. Specifically, it essentially ranks the features according to the following criterion in the ascending order [6]: $\sum_{ij} w_{ij} (\mathbf{x}_i^{(p)} - \mathbf{x}_j^{(p)})^2$, where $w_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ is the local similarity between \mathbf{x}_i and \mathbf{x}_j , obtained with all the features. Since the Laplacian score is a ranking-based *filter* approach, it does not perform the clustering. For a fair comparison, the clustering result obtained by our integrated method was not utilized when comparing with the Laplacian score. Instead, we ranked the features according to their weights, the performance of the normalized cut clustering with the top-ranked features was then used to evaluate our weighting scheme (denoted as *wrank+Ncut*) and the Laplacian score (denoted as *laprank+Ncut*).

Table 1. Summary of the benchmark data sets

Data Set	#DIM (p)	#INST (n)	#CL (k)
PIEC27_5p	1280	105	5
LEUKEMIA	999	38	2
LUNG	1000	197	4
MULTITISSUE	1000	103	4
ST.LEUKEMIA	985	248	6

As we have the label information of all five benchmark datasets, the clustering results were evaluated by comparing the obtained label of each data points with the ground truth. We used two standard measurements: the accuracy (ACC) and the normalized mutual information (NMI), higher values for both measurements will indicate good clustering performance.

¹ The source code in MATLAB is downloaded from: <http://www.cis.upenn.edu/~jshi/software/>



Fig. 1. Sample images from PIE face database under varying illumination conditions

Throughout the experiments, the parameter t in RBF kernel function was simply set at $0.0025 * \max(B^{(p)})$ for our method, where $\max(B^{(p)})$ is the maximum squared pairwise Euclidean distance between the elements in the p -th feature. The similarity used by the Ncut algorithm was also built with RBF kernel function, and the parameter t in RBF kernel function was set at $0.0025 * \max(B)$, where $\max(B)$ is the maximum squared pairwise Euclidean distance between the points.

4.1 Faces with Varying Illumination Conditions

A subset of the CMU PIE face database was used in this experiment. It contains 68 human subjects of the frontal poses (C27) but under different illumination conditions, with each subject having 21 faces. We used the cropped images² of 32×32 pixels. Samples extracted from the database are represented in Figure 1, in which it is observed that partitioning the face images of the same person in an unsupervised manner may be difficult because different persons appear similar under varying illumination conditions from the viewpoint of the image intensity. It therefore suggests the illumination insensitive features, e.g., the image gradient, may help the discrimination among identities [3]. In this experiment, we simply used the wavelet transform, which is able to compute the gradient, to generate the features for an image. However, a large amount of the wavelet coefficients are irrelevant for the task of separating between facial identities and it is therefore the goal of our algorithm to find those relevant coefficients as well as a more accurate clustering.

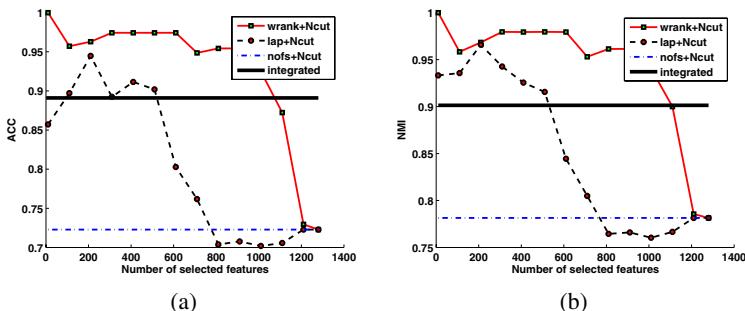


Fig. 2. Clustering results on PIE C27 data. (a): Comparison using the ACC index. (b): Comparison using the NMI index.

² The data is obtained from <http://www.cs.uiuc.edu/homes/dengcai2/Data/data.html>

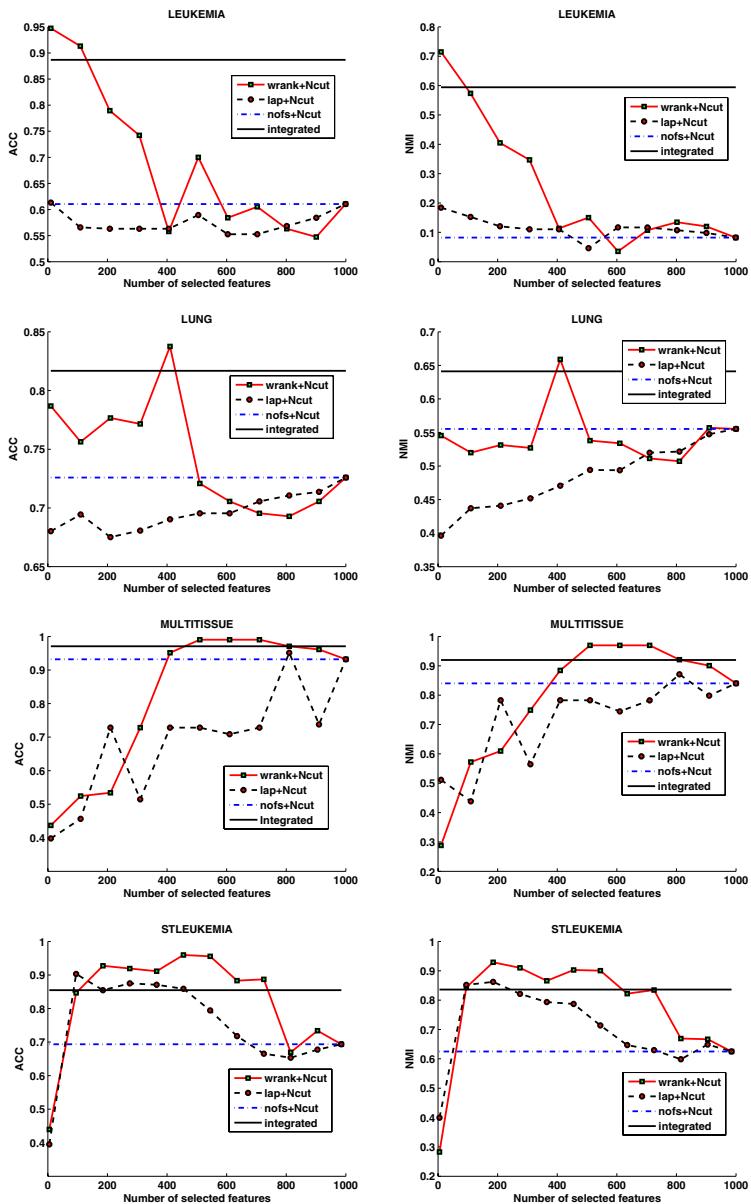


Fig. 3. Clustering results on the gene expression datasets. The first figure in each row demonstrates the comparison using the ACC index, the second one denotes the comparison using the NMI index.

Specifically, we first scaled the pixel value into $[0, 1]$, then the level-1 and level-2 Haar wavelet decompositions were performed over an image. After respectively normalizing the wavelet coefficients from each level by the average value in each corresponding level, we cascaded them to form a 1280-dimension vector representation for an image. Data from 5 classes were randomly selected out of the 68 objects, and this process was repeated 20 times, and the average performance was reported for all algorithms on the same data. Experimental results are summarized in Figure 2. It is observed that the proposed integrated algorithm significantly improves the conventional Ncut clustering with no feature selection. Besides, our weighting scheme largely outperforms the Laplacian score in the presence of many irrelevant wavelet coefficients.

4.2 Gene Expression Data

In this experiment, we studied the feature selection for the clustering on four public gene expression datasets: LEUKEMIA, LUNG, MULTITISSUE, STLEUKEMIA³. The characteristics of these datasets are summarized in Table 1. For LEUKEMIA dataset, expression values were first thresholded with a floor of 1 and a ceiling of 16000, followed by a base 10 logarithmic transformation. Then each gene was standardized to zero mean and unit variance across samples. For the MULTITISSUE and STLEUKEMIA datasets, each gene was standardized. For LUNG dataset, the already preprocessed expression profiles was used.

The average performance of all the algorithms was reported over 10 runs. The results are summarized in Figure 3. It can be seen that the integrated feature selection and clustering algorithm consistently outperforms the Ncut clustering with all features on all the datasets. In Figure 3, the Laplacian score does not always improve the clustering with no feature selection, and even falls much behind (see results for LUNG, MULTITISSUE), i.e., it is not robust for the gene expression data with large amount of noisy and irrelevant features. On the contrary, our weighting method demonstrates robustness against such features. A possible explanation for this superiority is that, for the gene expression data containing many irrelevant features, it may not be sensible to select features that try to preserve the locality, while selecting features that directly help the class discovery may be more appropriate.

5 Concluding Remarks

In this paper, we have proposed a novel feature weighting scheme for kernel-based clustering. It has a clearly defined objective function and can be solved iteratively. Rather than relying on the possible spurious “intrinsic” properties that may have been corrupted by irrelevant features, the weight assigned to each feature has direct relation to the clustering task. Experimental results have shown that our integrated feature weighting and clustering algorithm consistently improves the conventional kernel-based clustering without considering the feature selection. Moreover, when the feature weighting is used as a feature ranking method, it outperforms the state-of-art unsupervised one, the Laplacian score, in the presence of a lot of irrelevant features.

³ The gene expression datasets are obtained from [9].

It is necessary to point out that the performance of the integrated algorithm (*integrated*) is generally inferior to the best performance of normalized cut with the most relevant features selected by our weighting scheme (*wrank+Ncut*). The reason is that, the clustering quality of the integrated method essentially depends on the similarity matrix formed by a linear combination of individual kernel matrix constructed from each feature, but the interaction among features has not been presented. In contrast, in *wrank+Ncut*, the enhanced similarity matrix formed with the most relevant features selected by our weighting, does not omit the correlation among them. However, the most difficult issue for the ranking based approach in clustering is the determination of the number of relevant features to be selected. Since the cross-validation, a commonly used model selection technique in supervised learning, cannot be directly applied in unsupervised environment, in which the ground truth class labels are unavailable. Therefore the *integrated* method may be superior to the *wrank+Ncut* from the practical viewpoint. It will be the future work to incorporate the nonlinear interaction among features into our algorithm.

References

1. Zha, H., He, X., Ding, C., Simon, H.: Spectral relaxation for k-means clustering. In: NIPS, pp. 1057–1064 (2001)
2. Fan, K.: On a theorem of Weyl concerning eigenvalues of linear transformations. In: PNAS, pp. 652–655 (1949)
3. Chen, H.F., Belhumeur, P.N., Jacobs, D.W.: In search of illumination invariants. In: CVPR, pp. 254–261 (2000)
4. Dash, M., Choi, K., Scheuermann, P., Liu, H.: Feature selection for clustering-a filter solution. In: ICDM, pp. 115–122 (2002)
5. Dy, J., Brodley, C.: Feature selection for unsupervised learning. JMLR 5, 845–889 (2004)
6. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: NIPS, pp. 507–514 (2005)
7. Law, M.H., Jain, A.K., Figueiredo, M.A.T.: Feature selection in mixture-based clustering. In: NIPS, pp. 609–616 (2002)
8. Mitra, P., Murthy, C.A., Pal, S.K.: Unsupervised feature felection using feature similarity. PAMI 24(3), 301–312 (2002)
9. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning 52(1), 91–118 (2003)
10. Ng, A., Jordan, M., Weiss, Y.: On spectral slustering: analysis and an algorithm. In: NIPS (2002)
11. Yu, S.X., Shi, J.: Multiclass spectral clustering. In: ICCV, pp. 313–319 (2003)
12. Zhao, Z., Liu, H.: Spectral feature selection for supervised and unsupervised learning. In: ICML, pp. 1151–1158 (2007)

Availability of Web Information for Intercultural Communication

Takashi Yoshino¹, Kunikazu Fujii², and Tomohiro Shigenobu³

¹ Faculty of Systems Engineering, Wakayama University,
930 Sakaedani, Wakayama, Japan
yoshino@sys.wakayama-u.ac.jp

<http://www.wakayama-u.ac.jp/~yoshino/>

² Graduate School of Systems Engineering, Wakayama University,
930 Sakaedani, Wakayama, Japan

³ Language Grid Project,

National Institute of Information and Communications Technology,
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan
shigenobu@nict.go.jp

Abstract. Language and cultural differences pose significant barriers to intercultural communications. In our study, we specifically focus on sharing of semantic information as a method for overcoming cultural differences between. It was our belief that users must create their own annotations manually. However, we found that it is difficult for users to add annotations during chat communication. Therefore, we have developed a system that can automatically acquire these annotations. We performed experiments using this system for the case of communication between Japanese and foreign students. We present the results of these experiments regarding the users' evaluations and discussions. (1) The annotation contents of an image obtained by the automatic acquirement function obtained a relatively good evaluation. (2) With regard to annotation sentences, the content acquired through a multilingual links was given the highest evaluation. (3) With regard to the annotation images, the content acquired through an image search service was given the highest evaluation.

Keywords: intercultural communication, Web information, annotation, machine translation.

1 Introduction

The progress of globalization promotes various activities that exceed a country and its culture. In particular, there is a possibility that such an activity can become more familiar along with the global diffusion of the Internet. The approach of the intercultural communication using the Internet has already been started in various groups. Thus, the opportunity of intercultural communication increases by the advance of the information technology. However, there are two barriers to intercultural communication: language and culture. In order to overcome the language barrier, researches on machine translation are being actively

advanced. A user can communicate in the native language of another party by using machine translation.

A chat system named AmiChat and an electronic bulletin board System named TransBBS have been developed as communication tools based on machine translation[1,2]. Moreover, a language grid that can connect the language resources, such as a machine translation system and a bilingual dictionary on the Internet, has been started from 2006[3]. However, still, there is still no clear method of overcoming the cultural barrier.

We believe that the clear communication between users can be achieved by providing an image and two or more annotations to a word and a phrase in a sentence. We have developed a chat communication tool, named AnnoChat, for intercultural communication. AnnoChat supports chat communications through machine translation. In addition, it can provide an annotation to an image and text to arbitrary words and phrases in a chat environment. However, providing an annotation imposes a heavy load on a user while chatting. Therefore, we have developed a function to automatically acquire annotations from the Web in order to decrease user's load.

In this paper, we present the results of the experiments on this automatic annotation acquisition function.

2 Related Works

The research on the specification and enactment of metadata by applying annotations to contents is being actively conducted in various fields. There are some description frameworks for metadata, for example, OWL, which uses ontologies, and RDF (Resource Description Framework), which is enacted by W3C[4]. Funakoshi proposed semantic annotations (RDF schema) for resource sharing between multilingual collaboration tools[5]. ThirdVoice and Amaya[6] are systems that enable users to provide and share annotations on the Web. Moreover, there is a system called WebAnn that can provide annotations on the Web[7]. We had developed a prior chat system for intercultural communication[8]. The chat system has the function of text-based annotation manually. Most users felt that the annotation function is useful. However, it took much time to add annotations, and they complained about the manual operation.

In our research, chat is the main communication channel for a discussion. We consider annotations as a subchannel to exchange knowledge and/or information. The purpose of our research is to achieve better intercultural communication by giving annotations to specific words and phrases.

3 AnnoChat

3.1 Design Policy

Here, we present the design policy of the automatic annotation acquisition function:

1. Search for an annotation by giving a word and a phrase

This function searches and extracts words and phrases (common and proper nouns) that should comprise the annotation in a chat message.

2. Acquisition of annotation contents

This function acquires sentences and images that become explanations of the given words and phrases and is obtained through a Web search.

3.2 Annotation Function

The annotation of this system sets the annotation anchor (link) to arbitrary words and phrases. The explanation of words and phrases can be described as the annotation content. The character and image data can be appended the annotation content.

As a result, this function can be realized only by textual information, the explanation of the difficult content expressed in a photograph and an illustration. With this function, it is also possible to provide the explanation by using an image. There are two methods of providing annotations:

1. Manual

A user can select arbitrary words and phrases in a chat log and can provide an annotation to these words and phrases.

2. Automatic

This function acquires an annotation from a message by using the automatic annotation acquisition function and presents the annotation candidates to the user. The user can give an appropriate annotation only by selecting from among the annotation candidates.

3.3 Automatic Acquisition Function

The proposed automatic acquisition function consists of a words and phrases acquisition function and an annotation contents acquisition function.

1. Words and phrases acquisition function

This function extracts the targeted words and phrases that may give the annotation in a chat message by using morphological analysis.

2. Annotation contents acquisition function

This function uses words and phrases for a search request that are obtained by the words and phrases acquisition function. The retrieval results are collected from the Web and the function presents them to a user as the annotation candidates. This system acquires one paragraph at the beginning from Wikipedia. Automatic acquisition of the Yahoo! image uses the image search API.

3.4 User Interface of AnnoChat

Figure 1 shows a screenshot of the AnnoChat client. The words and phrases that comprise an annotation are displayed by the under line addition bold-faced type

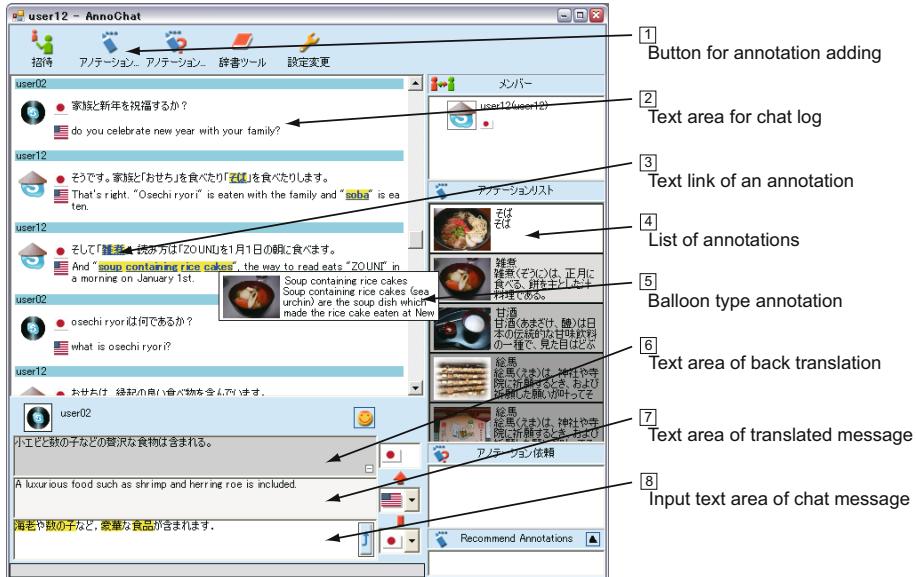


Fig. 1. Example screenshot of AnnoChat

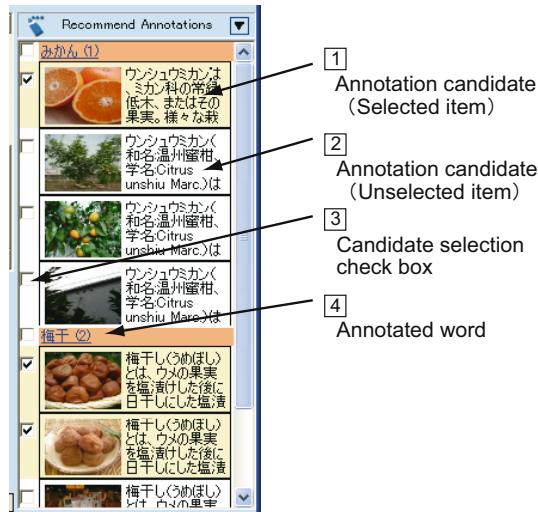


Fig. 2. Example screenshot displaying the recommended annotation candidates acquired by the automatic annotation candidate acquisition function of AnnoChat

in the chat log in Figure 1 [2]. When the mouse pointer is mouse over this part, the annotation item of the annotation list (Figure 1 [3]) is displayed Figure 1 [4]). The displayed annotation item automatically scrolls downward. When a

user gives the annotation manually, the user selects arbitrary words and phrases in the chat log and then clicks on the annotation add button. The user can input a chat message to an input area shown in Figure 1 [5]. The automatic annotation acquisition function displays the recommended annotation candidates on the screen.

Figure 2 shows the AnnoChat screen displaying the recommended annotation candidates. A check box is displayed beside each candidate, which comprises an image and a sentence, and is used to select the annotation. A user can edit the sentence of a recommended annotation candidate.

4 Evaluation Experiment Concerning Annotation Automatic Acquisition

We experimentally evaluated the annotation automatic acquisition function by comparing it with the manual addition of an annotation.

4.1 Experiment Method

In the experiments, a Japanese university student and an international student have some chat using AnnoChat. The international student is Chinese or belongs to an English-speaking country. In the experiments, there are ten pairs: seven pairs, each involving a Chinese user and a Japanese user; and three pairs, each involving an English user and a Japanese user.

The task of the experiment is that a Japanese user educates a foreigner about some aspects of Japan that are unknown to the foreigner. In this paper, a foreigner subject who questions is called the “Listener,” and the Japanese subject who answers the question is called the “Talker.”

The experimental procedure is as follows.

1. First, we explain the experiment for subjects. The subjects practice the usage of our system (AnnoChat) for some time. They introduce themselves through chat. They confirm the annotation adding method (both manual and automatic), the display, operation, etc.
2. The first chat session starts. The chat on the specified theme is executed for about 20 min.
3. After experiments, they answer a questionnaire.
4. The second chat session starts. The method of giving an annotation is different from that used in the first session.

The topic of the chat has one of the following themes. We made the frequency in an automatic and in a manual the same number.

- Topic A: Japanese Food culture in (meals, dishes, ingredients, etc.)
- Topic B: Japanese Festivals (New Year festival, Bon festival, Star Festival, etc.)

4.2 Experiment Result

We carried out the evaluation by using two types of questionnaires: “chat session questionnaire” evaluated after each chat session and “complete questionnaire” evaluated for the entire experiment. We prepared separate questionnaires for a talker subject and a listener subject. We used the Likert scale of five scores: 1. Strongly disagree, 2. Disagree, 3. Neutral, 4. Agree, 5. Strongly agree.

Talker subjects’ results for the questionnaire. Table 1 shows the individual results of talker subjects (Japanese) for the questionnaire and the significance probability of the results.

We use the Mann-Whitney significance test. The evaluation of automatic annotation is higher than that of a manual, giving the significance level of 5% from Table 1(1), (2). The automatic annotation method imposes less load that imposed by the manual method, giving the significance level of 5% from Table 1(3). However, the evaluation of the manual annotation method is higher than that of the automatic method, giving the significance level of 5% from Table 1(4). We consider that these results are due to the following reasons.

- Because the system extracted words and phrases automatically, the users did not give the annotations for the words that they wanted to.
- When experimenting on the automatic annotation method, the option of giving the annotations manually was not allowed.

Because AnnoChat can originally use both functions together, the above issue is easily improvable.

Table 2 shows talker (Japanese) subjects’ summary results for the questionnaire. Moreover, Table 3 shows the results of the free-description questionnaire on automatic acquisition. Table 2(a) shows the results of a paired comparison test between the manual annotation method the automatic annotation method. Almost all subjects answered that the automatic method imposed less load and the procedure of giving annotations was easy, from Table 2(a)(2) and (3). The annotation quality and amount were evaluated separately, as can be seen from Table 2(a)(1) and (4). The usefulness of the image candidates, the usefulness of the explanation document, and the acquirement latency were intermediate evaluations, as seen from Table 2(b).

Table 1. Talker subjects’ result for the questionnaire in each experiment

Questionnaire items	Average		Significance Probability
	Manual addition	Automatic addition	
(1) The operation of adding the annotation was easy.	3.4	4.6	0.002*
(2) The annotation could be added in a short time.	2.9	4.4	0.001*
(3) I felt the load in adding annotations.	3.1	2.0	0.018*
(4) I added annotations for all words and phrases that I felt must be explained.	4.1	2.9	0.013*

* $p < 0.05$.

Table 2. Talker subjects' comparison results for the questionnaire

(a) Evaluation results by paired comparison			
Questionnaire items	Manual addition	Automatic addition	Neutral
(1) Which method of providing annotation was useful for the other person?	3	4	3
(2) Which method is easy for me?	1	9	0
(3) Which method imposed less load on me?	0	10	0
(4) Which method was able to add the appropriate annotation?	5	5	0

(b) Results for the questionnaire of subjectivity evaluation (Average value of five-score evaluation)

Questionnaire items	Average
(5) I felt time stress until the automatic acquisition candidate was displayed.	3.3
(6) I felt that the image candidates from automatic acquisition function provided at least one useful candidate for the explanation of words and phrases for annotations.	3.2
(7) The explanation text that the automatic acquisition function acquired was appropriate to be used as it was—with a negligible correction—as an explanation of words and phrases.	3.5

Table 3. Free description concerning talker subjects' annotation automatic operation acquisition function(summary)

- I have not understood whether the annotation was translated accurately. Moreover, there was little time to mend it, too. It is good that the automatic acquisition function saves the time of the annotation adding.
- I want the automatic acquisition function to improve the accuracy of an image.
- Because the number of sentences was large, I did not strictly check the sentences. Because the chat is real-time communication, the annotation sentence should be made concise.
- I want a function by which a user can control the words and phrases of automatic acquisition. The function should allow the user to select words and phrases and to acquire annotations from the Web.

We found the following from the free description questionnaire.

- There is no comment on the acquirement latency, and we consider it to be acceptable.
- There is a problem in image candidates' accuracy.
- The user is not confident of the accuracy of the translation result.
- The amount of explanation is large.

Listener subjects' results of for the questionnaire. Table 4 shows the results for the questionnaire of listener (foreigner) subjects. Table 5 shows a free description of listener (foreigner) subjects.

The manual method of giving annotations was evaluated higher than the automatic method for the appropriateness of annotation sentences and images, as seen from Table 4(a)(1)(2).

Moreover, the images were given higher evaluation than sentences in both the methods. This may be due the inaccuracy of machine translation. That is, the users were often unable to understand the translated explanation from Table 5. The results of the paired comparison of easiness of content between the automatic and manual methods are shown in Table 4(b).

Table 4. Listener subjects' result of the questionnaire in each experiment

(a) Result of the questionnaire of subjectivity evaluation (Five stage evaluation average value)

Questionnaire items	Manual addition	Automatic addition
(1) Sentences of the annotation were appropriate as the explanation of words and phrases.	3.4	3.0
(2) The image of the annotation was appropriate as the explanation of words and phrases.	3.9	3.7
(3) The annotation (sentence & image) helped understand the chat conversation.	3.9	4.0
(4) The annotation was annotated on most of words and phrases that you hoped for the explanation.	3.3	3.4

(b) Evaluation result by paired comparison

Questionnaire items	Manual addition	Automatic addition	Neutral
Which method is it that the content of the annotation was comprehensible?	4	4	2

Table 5. Free description concerning listener subjects' annotation automatic acquisition function (summary)

– Opinion of listener subjects

- Opinion of listener subjects who answered that manual adding function is better because of the following reasons:
 - * The explanation of words and phrases is as comprehensible as Chinese.
 - * The chat topic is easy and comprehensible.
 - * An annotation image is relevant to the explanation.
- Opinion of listener subjects who answered that annotation automatic acquisition function is better:
 - * I think it is the most detailed. The image is also more appropriate.
 - * In a manual method, the explanation provided for words and phrases was insufficient. The photographs were also incomprehensible.
 - * An automatic adding function provided a detailed explanation. The image is also comprehensible.
- Opinion concerning annotation function:
 - Annotation sentences and images were comprehensible and were useful for the understanding the chat conversation.
 - I want a more comprehensible image. I want a function to enlarge an image. English sentences were incorrect or occasionally difficult to understand.
 - It is convenient that there is an annotation, particularly because it can be referred when the proper noun appears in the chat.
 - I think that the image was comprehensible; however, the explanation was very strange.

Table 6. Ranking frequency distribution table of a paired comparison test

(a) Content of sentences is appropriate.

	Automatic acquisition	Multilingual link	Manual 1	Manual 2
Rank 1	2	5	1	2
Rank 2	4	2	3	1
Rank 3	3	0	5	2
Rank 4	1	0	1	5
Σar	23	9	26	30

(b) The image (photograph) is appropriate.

	Automatic acquisition	Multilingual link	Manual 1	Manual 2
Rank 1	4	1	1	1
Rank 2	1	1	2	3
Rank 3	1	0	3	3
Rank 4	1	3	1	0
Σar	13	15	18	16

We found the following from the free-description questionnaire of Table 5.

- The accuracy of the manual method depends on the ability of a user to give the correct annotation.
- The influence of the chat topic is large.

Evaluation of appropriateness of annotation. Table 6(a) shows the results of the paired comparison of the appropriateness of the annotation sentences. It shows that the evaluation with a small number of ranks is high. The evaluation of the annotation that uses a multilingual link is high, as seen from Table 6(a).

In a word, the annotations acquired using the multilingual links of Wikipedia were better than other methods. The multilingual links of Wikipedia use the explanation written in Listener subjects' language as an annotation. Therefore, the explanation of the annotation became more comprehensible because there was no machine translation processing.

As for the evaluation of the appropriateness of an image, automatic acquisition was given the highest evaluation: It obtained the highest number of "rank 1," namely four.

The method of obtaining the best annotation contents is the following:

- Sentences that use Wikipedia articles through multilingual links
- Images obtained from image search services on the Web

However, the following problem exists for the use of multilingual links of Wikipedia: In Wikipedia, the editor of a page is different for each language. The content written in language A may be not the same as that written in language B.

Therefore, there is a possibility of misunderstanding due to the language differences when the web contents of two or more languages are used.

Discussion of experimental results. The present automatic acquisition function achieved the first purpose to lower the load of the user who gave an annotation; however, the image and sentences automatically acquired have low accuracies. Table 4(a)(3) shows that the evaluation of an image tends to be higher than those of sentences. We believe that an image improves the understanding level.

5 Conclusion

We have developed AnnoChat that supports intercultural chat communication communication. It has a function to add annotations to the words and phrases in chat messages.

We compared the automatic annotation acquisition function with the manual addition of annotations.

We found the following from the results of intercultural communication experiments.

1. The sentences of the annotation through a multilingual link such as Wikipedia had the highest evaluation.
2. The annotation images acquired through an image search service had the highest evaluation.
3. We showed that it was possible to give annotations without requiring the users to have the Web Information.

Acknowledgments. This work was partly supported by Grant-in-Aid for Scientific Research (B), No. 19300036, 2007–2009.

References

1. Flournoy, R.S., Callison-Burch, C.: Secondary Benefits of Feedback, User Interaction in Machine Translation Tools, Workshop paper for MT 2010: Towards a Roadmap for MT of the MT Summit VIII (2001)
2. Funakoshi, K., Yamamoto, A., Nomura, S., Ishida, T.: Lessons Learned from Multilingual Collaboration in Global Virtual Teams. In: HCII 2003 (2003)
3. Ishida, T.: Language Grid: An Infrastructure for Intercultural Collaboration. In: IEEE/IPSJ Symposium on Applications and the Internet (SAINT 2006), pp. 96–100 (2006)
4. Resource Description Framework (RDF), <http://www.w3.org/RDF/>
5. Funakoshi, K., Sugiyama, K., Ishida, T., Yoshino, T., Munemori, J., Zhang, H., Shi, Z.: Semantic Interoperability in Tools for Intercultural Collaboration. In: Third International Conference on Active Media Technology (AMT 2005), pp. 187–192 (2005)
6. Amaya Project, <http://www.w3.org/Amaya/>
7. Brush, A.J.B., Bergeron, D., Grudin, J., Borning, A., Gupta, A.: Supporting Interaction Outside of Class: Anchored Discussions vs. Discussion Boards. In: Proceedings of CSCL 2002, pp. 425–434 (2002)
8. Fujii, K., Yoshino, T., Shigenobu, T., Munemori, J.: Development of an Intercultural Collaboration System with Semantic Information Share Function. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3681, pp. 425–430. Springer, Heidelberg (2005)

Mining Weighted Frequent Patterns in Incremental Databases

Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer,
Byeong-Soo Jeong*, and Young-Koo Lee

Department of Computer Engineering, Kyung Hee University
1 Seochun-dong, Kihung-gu, Youngin-si, Kyunggi-do, 446-701, Republic of Korea
`{farhan, tanbeer, jeong, yklee}@khu.ac.kr`

Abstract. By considering different weights of the items, weighted frequent pattern (WFP) mining becomes an important research issue in data mining and knowledge discovery. However, existing algorithms cannot be applied for incremental and interactive WFP mining because they are based on a static database and require multiple database scans. In this paper, we present a novel tree structure IWFPT_{WA} (Incremental WFP tree based on weight ascending order) and an algorithm IWFPT_{WA} for incremental and interactive WFP mining using a single database scan. Extensive performance analyses show that our tree structure and algorithm are efficient for incremental and interactive WFP mining.

Keywords: Data mining, knowledge discovery, weighted frequent pattern mining, incremental mining, interactive mining.

1 Introduction

Weighted pattern mining [2], [3], [4], [5] can discover more important knowledge compared to the traditional frequent pattern mining [1], [6] by considering different weights of the items. It plays an important role in the real world scenarios. For example, in a real world business database, frequency of gold ring is very low compared to the frequency of pen sold. Therefore, knowledge about the patterns having low frequency but high weight remains hidden by finding only frequent patterns. The main contribution of the weighted frequent pattern mining is to retrieve this hidden knowledge from database.

Existing weighted frequent pattern mining algorithms [2], [3], [4], [5] considered fixed database and need multiple database scans to find the weighted frequent patterns. They did not consider that one or more transactions could be deleted/inserted/modified in the database. However, in our real world databases, new transactions can be added and old transactions can be deleted or modified any time. Therefore, single database scan approach to maintain the incremental updating of databases is essentially needed. Moreover, the data structures presented in the previous works do not have “*build once mine many*” property (by

* This study was supported by a grant of the Korea Health 21 R&D Project, Ministry For Health, Welfare and Family Affairs, Republic of Korea(A020602).

building the data structure only once, several mining operations can be done) which is very necessary for interactive mining.

Motivated from these real world scenarios, in this paper, we propose a novel tree structure IWFPT_{WA} (Incremental weighted frequent pattern tree based on weight ascending order) and a new algorithm IWFPA for incremental and interactive weighted frequent pattern mining. By using incremental and interactive weighted frequent pattern mining we can use the previous data structures and mining results and avoid unnecessary calculations when the database is updated or the mining threshold is changed. IWFPA can handle the incremental data in a single scan of database without any restructuring operation. IWFPT_{WA} has the “*build once mine many*” property for interactive mining. It arranges the items in weight ascending order and gets advantage in candidate pattern generation by keeping the highest weighted item in the bottom.

The remainder of this paper is organized as follows. In Section 2, we describe background. In Section 3, we develop our proposed tree structure and algorithm for incremental and interactive weighted frequent pattern mining. In Section 4, our experimental results are presented and analyzed. Finally, in Section 5, conclusions are drawn.

2 Background

A weight of an item is a non-negative real number which is assigned to reflect the importance of each item in the transaction database [2], [3]. For a set of items $I = \{i_1, i_2, \dots, i_n\}$, weight of a pattern $P\{x_1, x_2, \dots, x_m\}$ is given as follows:

$$Weight(P) = \frac{\sum_{q=1}^{length(P)} Weight(x_q)}{length(P)} \quad (1)$$

A weighted support of a pattern is defined as the resultant value of multiplying the pattern’s support with the weight of the pattern [2], [3]. A pattern is called a weighted frequent pattern if the weighted support of the pattern is greater than or equal to the minimum threshold [2], [3].

In the very beginning some weighted frequent pattern mining algorithms WARM [4], WAR [5] have been developed based on the Apriori algorithm [1] using candidate generation-and-test paradigm. WFIM [2] is the first FP-tree [6] based weighted frequent pattern mining algorithm using two database scans over a static database. They have used a minimum weight and a weight range. Items are given fixed weights randomly from the weight range. They have arranged the FP-tree [6] in weight ascending order and maintained they *downward closure* property [1] on that tree. To extract the more interesting weighted frequent patterns, WIP [3] algorithm introduces a new measure of weight-confidence to measure strong weight affinity of a pattern. Research [7], [8] has been done to develop incremental and interactive mining algorithms based on traditional frequent pattern mining. However, they cannot be applied for weighted frequent pattern mining. Therefore, we propose novel tree structure and algorithm for incremental and interactive mining for weighted frequent patterns.

3 Our Proposed Tree Structure and Algorithm

In our proposed tree structure IWFPT_{WA}, we maintain a header table to keep an item order. Each entry in a header table explicitly maintains item-id, frequency and weight information for each item. However, each node in a tree only maintains item-id and frequency information. Fig. 1 shows an example of a transaction database with weight table and also demonstrates incremental updating (insertion/deletion/modification) of this database. At first we create the IWFPT_{WA} for the original database shown in Fig. 1(a). We create a header table to keep all the items in weight ascending order. After that we scan the transactions one by one, sort the items in a transaction according to header table sort order and then insert into the tree. The first transaction T_1 has the items “a”, “b”, “c”, “d”, “g” and “h”. After sorting, the new order will be “c”, “d”, “h”, “g”, “b”, “a” and we insert T_1 in the IWFPT_{WA}. Similarly we insert all the transactions (upto T_6) in the the IWFPT_{WA} (shown in Fig. 2(a)). Fig. 1(a) also shows the original database is incremented by adding two groups of transactions db_1^+ and db_2^+ . Fig. 2(b) and Fig. 2(c) show that the IWFPT_{WA} can easily be incremented by inserting db_1^+ and db_2^+ respectively.

Fig. 1(b) shows the database is updated by deleting T_4 and T_7 , and by modifying T_5 (item “h” is replaced by “g”). Now we delete db^- (T_4 and T_7) from the current IWFPT_{WA}. T_4 is present in the tree as a path “c d b a”. To remove this transaction we have to reduce the frequency value of all the nodes in that path by one. After reducing the frequency value of node “a” and “b” become zero at that path. So we have to delete these two nodes (shown in Fig. 2(d)). Fig. 2(d) also shows T_7 has been removed in the same way and T_5 is modified by replacing item “h” by item “g” in the path “d h b a”.

Now we describe the mining process of our proposed algorithm IWFPT_{WA}. We use the global maximum weight (denoted by $GMAXW$) and the local maximum weight (denoted by $LMAXW$) to main the *downward closure* property [1]. $GMAXW$ is the maximum weight of all the items in the whole database. For example, in Fig 1(c), the item “a” has the global maximum weight of 0.6. $LMAXW$ is needed when we are doing the mining operation for a particular item. As IWFPT_{WA} is sorted in weight ascending order, we get advantage in the

The diagram illustrates the incremental updating of a transaction database. On the left, a bracket labeled "Original DB" groups the first six transactions: T_1 (a,b,c,d,g,h), T_2 (a,e,f), T_3 (b,e,f,g,h), T_4 (a,b,c,d), T_5 (a,b,d,h), T_6 (a,b,d,e). Below this, a bracket labeled " db_1^+ " groups transactions T_7 (a,b,c), T_8 (a,b,c,d,g), T_9 (a,b,g). Another bracket labeled " db_2^+ " groups transaction T_{10} (a,c). Part (a) "Incremented database" shows the full set of transactions T_1 through T_{10} . Part (b) "Updated database" shows the database after removing T_4 and modifying T_5 (replacing "h" with "g"). Part (c) "Weight table" lists items with their weights: a (0.6), b (0.5), c (0.2), d (0.35), e (0.5), f (0.3), g (0.4), h (0.38).

TID	Transactions
T_1	a,b,c,d,g,h
T_2	a,e,f
T_3	b,e,f,g,h
T_4	a,b,c,d
T_5	a,b,d,h
T_6	a,b,d,e
T_7	a,b,c
T_8	a,b,c,d,g
T_9	a,b,g
T_{10}	a,c

(a) Incremented database

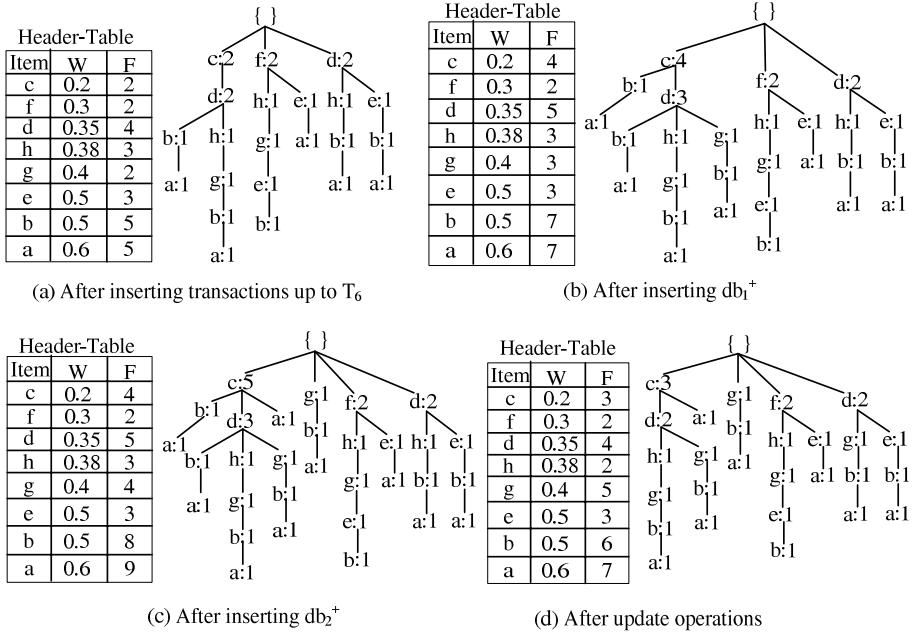
TID	Transactions
T_1	a,b,c,d,g,h
T_2	a,e,f
T_3	b,e,f,g,h
T_5	a,b,d,g
T_6	a,b,d,e
T_8	a,b,c,d,g
T_9	a,b,g
T_{10}	a,c

(b) Updated database

Items	W
a	0.6
b	0.5
c	0.2
d	0.35
e	0.5
f	0.3
g	0.4
h	0.38

(c) Weight table

Fig. 1. Incremental updating of a transaction database with weight table

**Fig. 2.** Incremental and update operations in $IWFPT_{WA}$

bottom up mining operation. For example, after mining the weighted frequent patterns prefixing the item “*a*”, when we go for mining operation prefixing the item “*b*”, then the item “*a*” will never come in any conditional trees. As a result, now we can easily assume that the item “*b*” has the maximum weight. This type of maximum weight in mining process is known as *LMAXW*.

Consider the current database presented at Fig. 1(b), $IWFPT_{WA}$ constructed for that database in Fig. 2(c), the weight table at Fig 1(c) and minimum threshold = 2.2. Here the $GMAXW = 0.6$ and after multiplying the frequency of each item with $GMAXW$, the weighted frequency list is $< c : 1.8, f : 1.2, d : 2.4, h : 1.2, g : 3.0, e : 1.8, b : 3.6, a : 4.2 >$. As a result, the candidate items are “*d*”, “*g*”, “*b*” and “*a*”. At first the conditional tree of the bottom most item “*a*” (shown in

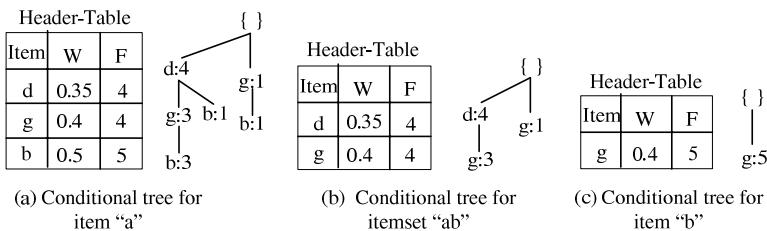
**Fig. 3.** Mining process in $IWFPT_{WA}$

Fig. 3(a)) is created by taking all the branches prefixing the item “*a*” and deleting the nodes containing an item which cannot be a candidate pattern with the item “*a*”. For item “*a*”, $LMAXW = 0.6$ and we can get the weighted frequency list for item “*a*” by multiplying the other item’s frequency with $LMAXW$. So, the weighted frequency list for the item “*a*” is $< d : 2.4, g : 2.4, b : 3.0 >$ and candidate patterns “*ad*”, “*ag*”, “*ab*” and “*a*” are generated. In the similar fashion, conditional tree for the pattern “*ab*” is created in Fig. 3(b) and candidate patterns “*abd*” and “*abg*” are generated. For item “*b*” the $LMAXW = 0.5$, the weighted frequency list is $< d : 2.0, g : 2.5 >$ and the candidate pattern “*bg*” is generated from its conditional tree (shown in Fig. 3(c)). After testing all the candidate patterns with their actual weights and the weighted frequency the resultant weighted frequent patterns are $< a : 4.2, b : 3.0, ab : 2.75, bg : 2.25 >$.

4 Experimental Results

To evaluate the performance of our proposed tree structure and algorithm, we have performed several experiments on the real-life mushroom and kosarak datasets from the frequent itemset mining dataset repository (<http://fimi.cs.helsinki.fi/data/>). These datasets do not provide the weight values of each item. As like the performance evaluation of the previous weight based frequent pattern mining [2], [3], [4], [5], we have generated random numbers for the weight values of each item, ranging from 0.1 to 0.9. Our programs were written in Microsoft Visual C++ 6.0 and run with Windows XP operating system on a Pentium dual core 2.13 GHz CPU with 1GB main memory.

Mushroom (0.56 MB, 8124 transactions, 119 distinct items) is a dense dataset having transaction length 23 for its every transaction. The running time comparison in mushroom dataset for the worst case of interactive mining, i.e. thresholds in descending order (at first 30% then 25% and so on) is shown in Fig. 4. Our algorithm gets benefit after the first mining threshold. After the first threshold our trees do not have to be constructed again. As the threshold decreases, new mining operation is needed. Fig. 4 demonstrates that IWFP_{WA} outperforms WFIM by using a single database scan approach and interactive mining.

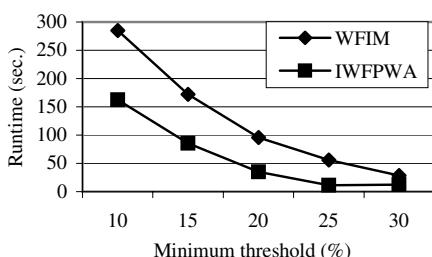


Fig. 4. Runtime comparison on mushroom

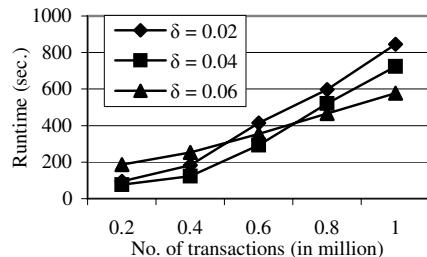


Fig. 5. Incremental mining on kosarak

We have tested the effectiveness of IWFP_{WA} in incremental mining with the kosarak dataset (30.5 MB). It has almost 1 million transactions (990002) and 41270 distinct items. At first we have created the IWFPT_{WA} for 0.2 million transactions of this dataset and then perform mining operation with the minimum threshold 6% ($\delta = 0.06$). Then we perform the interactive mining on that IWFPT_{WA} with other two thresholds 4% and 2%. Another 0.2 million transactions are added in the tree and perform mining operations with the same minimum thresholds and in same order. In the same way all the transactions in the kosarak dataset are added and mining operations are performed at each stage in the similar fashion. This result is shown in Fig. 5. Fig. 5 also shows that as the database is increasing, the tree construction and mining time is increasing, and in each stage tree construction cost is only needed for $\delta = 0.06$.

5 Conclusions

The main contributions of this paper are to provide a novel tree structure and algorithm for incremental and interactive weighted frequent pattern mining. Our proposed tree structure is easy to construct and maintain for the incremental updating of the databases. It has the “*build once mine many*” property and highly suitable for interactive mining. Our proposed algorithm needs only one database scan. Moreover, it is capable of using previous tree structure and mining results to reduce the calculations by a remarkable amount. Extensive performance analyses show that our tree structure and algorithm are very efficient for incremental and interactive weighted frequent pattern mining.

References

1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: 20th Int. Conf. on Very Large Data Bases (VLDB), pp. 487–499 (1994)
2. Yun, U., Leggett, J.J.: WFIM: weighted frequent itemset mining with a weight range and a minimum weight. In: Fourth SIAM Int. Conf. on Data Mining, USA, pp. 636–640 (2005)
3. Yun, U.: Efficient Mining of weighted interesting patterns with a strong weight and/or support affinity. Information Sciences 177, 3477–3499 (2007)
4. Tao, F.: Weighted association rule mining using weighted support and significant framework. In: 9th ACM SIGKDD, USA, pp. 661–666 (2003)
5. Wang, W., Yang, J., Yu, P.S.: WAR: weighted association rules for item intensities. Knowledge Information and Systems 6, 203–229 (2004)
6. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Mining and Knowledge Discovery 8, 53–87 (2004)
7. Leung, C.K.-S., Khan, Q.I., Li, Z., Hoque, T.: CanTree: a canonical-order tree for incremental frequent-pattern mining. Knowledge and Information Systems 11(3), 287–311 (2007)
8. Tanbeer, S.K., Ahmed, C.F., Jeong, B.-S., Lee, Y.-K.: CP-tree: A tree structure for single-pass frequent pattern mining. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 1022–1027. Springer, Heidelberg (2008)

Revision of Spatial Information by Containment

Omar Doukari, Robert Jeansoulin, and Eric Würbel

Laboratoire LSIS, UMR CNRS 6168 MARSEILLE CEDEX 20 - France

{omar.doukari,robert.jeansoulin,eric.wurbel}@lsis.org

Abstract. In this paper, we define a revision approach based on the space decomposition using one of the properties of geographical information called the containment property. This approach allows us to calculate a correct and complete solution which respects the principle of minimal change.

Keywords: Belief revision, Spatial information, Containment property.

1 Introduction

The revision problem is known as a difficult problem, and there does not exist any revision approach, really efficient, that can treat real applications with a huge amount of data. In the general case the theoretical complexity of revision belongs to the \prod_2^p class in the framework of propositional logic [1,2].

In this paper, we define a syntactic revision approach based on decomposition, by taking into account one of the geographical information properties called the containment property which allows us to decompose the problem into subproblems and to revise each one separately while keeping a solution respecting the principle of minimal change.

After giving some definitions in Section 2, we devote Section 3 to define a decomposition of the geographical space. In Section 4, we define the containment property through a necessary and sufficient condition called the hypothesis H0. Then, we present a thorough study of confined conflicts and their processing before defining the new revision strategy based on the containment property. We conclude in Section 5.

2 Preliminaries

Let \mathcal{L} be a propositional language defined over some finite set of propositional variables \mathcal{V} and the usual connectors ($\neg, \vee, \wedge, \rightarrow, \leftrightarrow$). A literal is a variable $v \in \mathcal{V}$ or its negation $\neg v$. A clause is a disjunction (\vee) of literals and $\mathcal{V}(\alpha)$ denotes the set of variables composing the clause α . Similarly for a set X of clauses, i.e., $\mathcal{V}(X) = \bigcup_{\alpha \in X} \mathcal{V}(\alpha)$. We will denote the cardinality of a set S as $|S|$. α is an unary clause if $|\mathcal{V}(\alpha)| = 1$, it is a binary clause if $|\mathcal{V}(\alpha)| = 2$, and it is said n -ary clause if $|\mathcal{V}(\alpha)| \geq 3$.

We shall denote by ξ the considered geographical space, and by $S1 \cup S2$ the set of clauses attached to ξ . If B is a subspace of ξ , then $Var(B)$ denotes the

set of variables defined in B , and $SB(B)$ denotes the subset of clauses attached to B , i.e., $SB(B) = \{c | c \in S1 \cup S2, \text{ and } \mathcal{V}(c) \subseteq Var(B)\}$.

The “revision problem” for spatial knowledge can be expressed as follows. Let $S1$ be a finite set of clauses representing initial knowledge attached to ξ , and $S2$ be a second set of clauses. It is also attached to ξ , and more reliable than $S1$. Furthermore, $S1$ and $S2$ are consistent. If the union $S1 \cup S2$ is consistent, then the revision of $S1$ by $S2$ consists in adding $S2$ to $S1$. Otherwise, we need to keep $S2$ and removing the least possible information from $S1$ in order to maintain consistency. This can be interpreted by the identification of minimal subsets R of $S1$ such that $(S1 \setminus R) \cup S2$ is consistent [3,4]. The approach that naturally comes to mind is to calculate the collection $Min(S1 \cup S2)$: the set of all minimal inconsistent subsets (minimal conflicts) of $(S1 \cup S2)$ and to construct the minimal sets S such that $\forall M \in Min(S1 \cup S2), S \cap M \neq \emptyset$ and $\forall S' \subset S, \exists M \in Min(S1 \cup S2)$ such that $S' \cap M = \emptyset$. It remains then to choose which set S to pick to restore consistency. One of the approaches which implement this strategy is the Reiter revision approach (REM approach) detailed in [4]. It calculates the minimal hitting sets (MHSs) of a collection of sets. Its worst-case complexity is: $O(C^3 \times 2^{2 \times C})$, where $C = |S1 \cup S2|$.

Definition 1. Let \mathcal{F} be a collection of sets. F is a HS of \mathcal{F} iff $F \subseteq \bigcup_{S \in \mathcal{F}}$ such that for all $S \in \mathcal{F}, F \cap S \neq \emptyset$. F is a MHS of \mathcal{F} , iff $\forall F' \subset F, F'$ is not a HS of \mathcal{F} .

We establish the correspondence between MHSs and the information to be removed to restore consistency as follows.

Theorem 1 [5]. Let $R \subseteq S1$, R is the minimal subset s.t., $(S1 \setminus R) \cup S2$ is consistent iff R is a MHS of the collection of minimal inconsistent subsets of $(S1 \cup S2)$ and $R \cap S2 = \emptyset$.

3 Splitting Up Space into Blocks

Our decomposition strategy is based on the external connection of parcels denoted by \mathfrak{R} [6].

Definition 2. If p_i, p_j are two neighbor parcels of ξ , then $dist(p_i, p_j) = \min\{m | \mathfrak{R}^m(p_i, p_j)\}$, otherwise, $dist(p_i, p_j) = \infty$. The relation \mathfrak{R}^m is s.t.: $\forall p_i, p_j \in \xi, \mathfrak{R}^m(p_i, p_j)$ iff $\exists p_1, p_2, \dots, p_m \in \xi$ s.t., $\mathfrak{R}(p_i, p_1), \mathfrak{R}(p_1, p_2), \dots, \mathfrak{R}(p_{m-1}, p_m)$, and $\mathfrak{R}(p_m, p_j)$.

Definition 3. Let p be a parcel in ξ . A block B with radius k ($k \geq 0$) and a center p is s.t., $B = \{p_i \in \xi | dist(p_i, p) \leq k\} \cup \{p\}$.

We say that a block B is “tractable” if the revision of the amount of information $SB(B)$ is possible by using simply the Reiter approach.

Partition Algorithm. A decomposition of ξ into blocks with radius k , denoted Partition(ξ, k), with respect to \mathfrak{R} is a set ρ of blocks with radius k and different centers, s.t.: (i) $\bigcup_{B \in \rho} = \xi$; and (ii) $\forall B_1, B_2 \in \rho : B_1 \cap B_2 = \emptyset$.

We now define the covering whose thickness is equal to r for a block B .

Definition 4. A covering with thickness r , of a block B with radius k and center p ($k < r$), denoted by $Cov_r(B)$, is $Cov_r(B) = \{p_i \in \xi \mid k < dist(p_i, p) \leq r\}$.

We define a minimal conflict, and its size as follows.

Definition 5. Let $C \subseteq (S1 \cup S2)$ be an inconsistent subset of clauses. C is a minimal conflict iff $\forall C' \subset C, C'$ is consistent. The size of C , denoted by $size(C)$, is $size(C) = \max\{dist(p_i, p_j) \mid p_i, p_j \in \xi, V(C) \cap Var(\{p_i\}) \neq \emptyset, \text{ and } V(C) \cap Var(\{p_j\}) \neq \emptyset\}$.

4 Containment-Based Revision

The containment property limits the effects of an inconsistency such that it cannot have an unbounded influence on other information. Instead, this influence is restricted to a “local area” which depends on the nature of the data, the structure or topology of information and the constraints defined on this structure. This property can be expressed through the following condition.

Hypothesis H0: If T_{max} represents the maximal size of minimal conflicts existing in $S1 \cup S2$, and R_B represents the radius of the largest tractable block, then $T_{max} \leq \frac{2 \times R_B}{3}$.

A direct consequence of this hypothesis is that if the existing minimal conflicts in a block B are *independent* of the minimal conflicts containing at least one clause belonging to the covering of B , then these minimal conflicts are *independent* of all minimal conflicts existing outside B .

In the following, all what we assume is that the hypothesis H0 is satisfied, and to simplify notation, we shall not specify neither the radius of blocks, nor the thickness of the coverings. So, if B is a block, we just put $Cov(B)$ to denote its covering.

We now present a thorough study of conflicts and their processing locally.

4.1 Independent Conflicts

Space Independent Conflicts. In this case, there does not exist minimal conflicts whose some clauses belonging to the block and some others to the covering. To verify the presence of such an independence, we define a new procedure for the search of MHSs, called *REM-Special*.

REM-Special Algorithm. The procedure *REM-Special* ($C1$: set of clauses, $C2$: set of clauses) is a slightly modified version of Reiter approach such that, when calculating the MHSs, it takes into account only the minimal conflicts belonging to $C1$ and which contains at least one clause in $C2$.

Definition 6. Let B be a block. The minimal conflicts existing in B are spatially independent of the other minimal conflicts iff: $\forall mhb \in REM\text{-Special}(SB(Cov(B) \cup B), SB(B)), mhb \cap SB(Cov(B)) = \emptyset$.

Information Independent Conflicts. In this case, there exists some minimal conflicts whose some clauses belonging to the block and some others to the covering, but the intersection between these minimal conflicts and those existing in the block should be empty. For detecting this independence we have proved the following lemma and proposition:

Lemma 1. *Let S be a collection of sets of clauses s.t., $\forall s_1, s_2 \in S, s_1 \not\subset s_2$ and $s_2 \not\subset s_1$. Let E be the set of MHSs of S . Then, $\forall s \in S, \forall c \in s, \exists e \in E$ s.t., $c \in e$.*

Proposition 1. *Let $E1$ and $E2$ be two sets of the MHSs of the two collections of sets of clauses $S1$ and $S2$, respectively. $S1$ and $S2$ are information independent (i.e., $\forall s_1 \in S1, \forall s_2 \in S2, s_1 \cap s_2 = \emptyset$) iff $\forall e_1 \in E1, \forall e_2 \in E2, e_1 \cap e_2 = \emptyset$.*

Definition 7. *Let B be a block. The minimal conflicts existing in B are informational independent of the other minimal conflicts iff: $\forall mhb \in REM(SB(B)), \forall mhp \in REM-Special(SB(Cov(B) \cup B), SB(Cov(B))) : mhb \cap mhp = \emptyset$.*

The detection of the independence property, spatial and/or informational, between two collections of sets of clauses, make the computation of the global MHSs of the union of the two collections easier. Thus, to calculate the MHSs of the two collections, we calculate them for each collection independently, then we concatenate the resulting MHSs of the two collections. Formally, we have the following result:

Proposition 2. *Let S be a collection of sets of clauses and E be the set of MHSs of S . If $S = \bigcup_{i=1}^n S_i$ s.t., $\forall s \in S_i, \forall s' \in S_j, i \neq j$ we have: $s \cap s' = \emptyset$, then: $E = \{e_1 \cup \dots \cup e_n | (e_1, \dots, e_n) \in E1 \times \dots \times En$ s.t $E1, \dots, En$ are the sets of MHSs of S_1, \dots, S_n , respectively }.*

4.2 Dependent Conflicts

In this case, the intersection between the minimal conflicts existing in the block and those containing, at least, one clause belonging to the covering of block, is not empty. For treating this case, we define a simple approach. It consists in partitioning the space into blocks, then, searching the local MHSs of the minimal conflicts existing in each subspace composed of a block and its covering by ignoring the minimal conflicts belonging entirely to the covering, because they will be considered later in one of the next block. The construction of global MHSs is made incrementally, at the same time than the search of sets of local MHSs of different subspaces. When we obtain the local MHSs of a given subspace composed of a block and its covering, we concatenate them with the set of MHSs obtained in the previous iterations, and for avoiding the reprocessing of the corresponding minimal conflicts, we discard the block from the list of the blocks to process. After each concatenation, we make a test of minimality, to keep only the MHSs among the resulting ones. We have proved the following result which validates this approach.

Algorithm 1. Containment-Revision

Require: ξ - geographic space , T_{max} - the maximal size of MIs
 1: [First step for filtering independent MIs to improve the efficiency : omitted here]
 2: $Conf \leftarrow$ the clauses involved in the treated MIs during the first step.
 3: $GMH \leftarrow$ the set of MHSs obtained in the first step.
 4: $blocs \leftarrow Partition(\xi, r)$ [r is an arbitrary radius]
 5: **for** all B in $blocs$ **do**
 6: $TMH \leftarrow \emptyset$
 7: $MHB \leftarrow REM-Special(SB(Cov_{T_{max}}(B) \cup B) \setminus Conf, SB(B))$
 8: **for** all $gmh \in GMH$ **do**
 9: $TMH \leftarrow TMH \cup (gmh \bullet MHB)$
 10: **end for**
 11: **if** $(\exists tmh, tmh' \in TMH)$ and $(tmh \subset tmh')$ **then**
 12: to remove tmh'
 13: **end if**
 14: $GMH \leftarrow TMH$
 15: $blocs \leftarrow blocs \setminus \{B\}$
 16: **end for**
 17: **return** GMH

Proposition 3. Let F be a collection of sets of clauses s.t., $F = \bigcup_{i=1}^n Fi$, and $\forall Fi, Fj, i \neq j, Fi \cap Fj = \emptyset$. If $E1, E2, \dots$, and En , are the sets of MHSs of $F1, F2, \dots$, and Fn , respectively, then E , the set of global MHSs, is $E = \min^1\{e_1 \cup \dots \cup e_n | (e_1, \dots, e_n) \in E1 \times \dots \times En\}$.

From the previous study, we define the *Containment-Revision* algorithm for the search of the MHSs of minimal conflicts existing in the geographical space. This strategy is an hybrid approach between the two resolutions that we have seen previously, namely the resolutions of the independence and that of the dependence case.

In a first time, this strategy searches to detect all independences (informational and/or spatial) which can exist between minimal conflicts (omitted part in Algorithm 1 for the sake of simplicity). Thereafter, processing the remaining minimal conflicts is made with respect to the dependence case of minimal conflicts. Finally, we concatenate the local MHSs resulting from the two previous steps to constitute global MHSs of the minimal conflicts existing in the whole geographical space.

For greater clarity of the *Containment-Revision* algorithm, we introduce the following definition:

Definition 8. Let C be a collection of sets, and E be a set. $E \bullet C = \{S | S = E \cup c, c \in C\}$.

If m denotes the cardinality of the space ($|\xi|$), then the worst-case complexity of the *Containment-Revision* algorithm is: $O(m \times |S1 \cup S2| \times 2^{2 \times |S1 \cup S2|})$.

¹ Minimality in the sense of set inclusion.

Proposition 4. If $COMP_{REM}$ denotes the complexity of the REM algorithm and $COMP_{CONT}$ denotes the complexity of the Containment-Revision algorithm, then $COMP_{REM} = \frac{|S1 \cup S2|^2}{m} \times COMP_{CONT}$ such that $m = |\xi|$.

To show that this ratio is a nice amelioration of the REM algorithm, let's consider the real scale "flooding" application [4]: 300 parcels, 2 attributes per parcel, defined over a sampled domain of water heights (about 10–20 useful sample level by parcel). The representation of this problem leads to about 100.000 propositional clauses, hence $|S1 \cup S2| = 10^5$, and $r = (10^{10}/300) = 3.10^7$. It is 30 million times faster.

5 Conclusion

The global treatment of real applications, notably revision, is difficult and may be impossible. This is due mainly to two reasons: (i) the huge amount of information in real applications, and (ii) the exponential complexity of existing revision algorithms.

In this paper, we have proposed a revision strategy based on the decomposition of the geographical space into blocks and coverings. Then, to make the revision of blocks separately possible, we have exploited one of the geographical information properties called the containment property.

However, it should be noted that it is not always possible to make an assumption on the maximal size of minimal conflicts. This question is staying open with the validation of our theoretical results by implementing *Containment-Revision* on real applications.

References

1. Eiter, T., Gottlob, G.: On the complexity of propositional knowledge base revision, updates, and counterfactuals, pp. 261–273 (1992)
2. Liberatore, P., Schaerf, M.: The complexity of model checking for belief revision and update. In: AAAI/IAAI, vol. 1, pp. 556–561 (1996)
3. Papini, O.: A complete revision function in propositional calculus. In: ECAI 1992, New York, NY, USA, pp. 339–343. John Wiley & Sons, Inc., Chichester (1992)
4. Würbel, E., Papini, O., Jeansoulin, R.: Revision: an application in the framework of gis. In: KR 2000, Breckenridge, Colorado, USA, pp. 505–516 (April 2000)
5. Würbel, E., Papini, O., Jeansoulin, R.: Spatial information revision: A comparison between three approaches. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 454–465. Springer, Heidelberg (2001)
6. Randell, D.A., Cui, Z., Cohn, A.: A spatial logic based on regions and connection. In: Nebel, B., Rich, C., Swartout, W. (eds.) KR 1992, San Mateo, California, pp. 165–176. Morgan Kaufmann, San Francisco (1992)

Joint Power Control and Subcarrier Allocation in MC - CDMA Systems - An Intelligent Search Approach

Le Xuan Dung

Ministry of Information and Communication,
Project Management Unit,
2A Vo Van Dung, Dong Da, Hanoi, Vietnam
1xdung@mic.gov.vn

Abstract. In this paper we consider the problem of frequency assignment and power allocation for multiple users in a MC-CDMA system when the channel is known at the transmitter. Two intelligent search algorithms (Simulated Annealing and Tabu Search) were applied to the framework to find solution to the joint optimization problem. Simulations showed that by applying the framework the optimizing algorithms are capable of providing good solutions within a reasonable amount of time.

Keywords: CDMA, Multicarrier, Simulated Annealing, Tabu Search, Power Control, Subcarrier Allocation.

1 Problem Statement

One of the inherent problems in a CDMA system is multi-user interference. Because CDMA codes are not totally orthogonal, the signals from other users are seen as interference at the desired receiver [1][10]. In the recent years, there are many works that addressed these problems [2][3][4][5][6]. These works studied various aspects of power control for single carrier CDMA systems. However, to the author knowledge until now there is no prior work discussing the problem of jointly optimizing power control and sub-carrier allocation in a MC-CDMA system. Assume that there are K user in the systems. Let $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K \in \mathbf{C}^N$ be the CDMA codes assigned to users $1, 2, \dots, K$, where N is the length of each CDMA code. Because the channel on each sub-carrier is flat, denote $\mathbf{H} \in \mathbf{C}^{L \times K}$ as the channel matrix of the downlink channel at the base station, where L is the number of sub-carriers in the system. The channel response from the base station to the k^{th} mobile station on the l^{th} sub-carrier is the element corresponds to the l^{th} row and the k^{th} column of the channel matrix, $\mathbf{H}_{l,k}$.

Let us denote u as the carrier assignment function, i.e. the carrier of user k is $u(k)$. The received signal at k^{th} the mobile station is

$$\mathbf{y}_k = \sqrt{p_k} \mathbf{H}_{u(k),k} \mathbf{c}_k b_k + \sum_{\substack{j \in \{1, 2, \dots, K\} \\ u(j)=u(k)}} \sqrt{p_j} \mathbf{H}_{u(k),k} \mathbf{c}_j b_j + \mathbf{n}_k, \quad (1)$$

where p_k is the power level allocated for user k .

The signal to noise plus interference ratio at the output matched filter is therefore

$$\gamma_k = \frac{p_k \mathbf{H}_{u(k),k}^2 \|\mathbf{c}_k^H \mathbf{c}_k\|^2}{\sum_{\substack{j \in \{1, 2, \dots, K\} \\ u(j)=u(k)}} p_j \mathbf{H}_{u(k),k}^2 \|\mathbf{c}_k^H \mathbf{c}_j\|^2 + (\mathbf{c}_k^H \mathbf{c}_k)} \quad (2)$$

Assuming that m_l users are assigned to sub-carrier l , denote these users as $v_l(1), v_l(2), \dots, v_l(m_l)$. Applying Equation (4) for each user using subcarrier l , we have

$$\gamma_{v_l(i)} = \frac{p_{v_l(i)} \mathbf{H}_{u(v_l(i)),v_l(i)}^2 \|\mathbf{c}_{v_l(i)}^H \mathbf{c}_{v_l(i)}\|^2}{\sum_{\substack{j=1, \dots, m_l \\ j \neq i}} p_{v_l(j)} \mathbf{H}_{u(v_l(i)),v_l(i)}^2 \|\mathbf{c}_{v_l(i)}^H \mathbf{c}_{v_l(j)}\|^2 + (\mathbf{c}_{v_l(i)}^H \mathbf{c}_{v_l(i)})}, \quad i = 1, 2, \dots, r \quad (3)$$

Denote $\boldsymbol{\gamma}_l = [\gamma_{v_l(1)}, \gamma_{v_l(2)}, \dots, \gamma_{v_l(m_l)}]^T$, $\mathbf{p}_l = [p_{v_l(1)}, p_{v_l(2)}, \dots, p_{v_l(m_l)}]^H$, $\boldsymbol{\Gamma}_l = \text{diag}(\boldsymbol{\gamma}_l)$.

Equations (3) can be written in matrix form as $(\mathbf{C}_l - \mathbf{A}_l) \mathbf{p}_l = \frac{1}{\mathbf{H}_{v_l(i),v_l(i)}^2} \boldsymbol{\Gamma}_l \mathbf{r}_l$, where

\mathbf{C}_l is a diagonal matrix with diagonal elements representing the desired power gain of each user user, \mathbf{A}_l with element (i, j) representing the interference gain from user i to user j on subcarrier l , and \mathbf{r}_l is a column vector with the i^{th} element equals $\mathbf{c}_i^H \mathbf{c}_i$. Therefore

$$\mathbf{p}_l = \frac{1}{\mathbf{H}_{v_l(i),v_l(i)}^2} (\mathbf{C}_l - \mathbf{A}_l)^{-1} \boldsymbol{\Gamma}_l \mathbf{r}_l. \quad (4)$$

The total transmit power is

$$P = \sum_{l=1}^L \mathbf{1}^T \mathbf{p}_l \quad (5)$$

The objective function is (5). If the mapping function u is known, (5) can be computed backward directly using Equations (4), (3), (2), and (1). Recall that u performs the mapping from the set of users to the set of subcarriers. The total number of functions available is L^K .

2 The Intelligent Search Approach

As we explained in previous sections, the search space of the joint power control and subcarrier optimization problem is very large and cannot be implemented using exhaustive search for a typical system. In this section we introduce a general approach to the problem. We all know that many optimization algorithms, including Simulated Annealing and Tabu Search are based on topological structure of the problem by moving from the current solution to another solution that is defined as a neighbor to it. This approach has been applied successfully to many NP-hard problems, including the well known Traveling Salesman Problem [7][8][9].

If we change the subcarrier assignment of user k from l_1 to l_2 this assignment is still valid as long as $0 \leq l_1, l_2 < L$. Motivated by this observation, we define an n -opt neighbor of u as a mapping from the user set to the subcarrier set in which all but n users' carrier assignments are kept the same as those in u .

```

Initialization
    u ← random
    T ← T0
For iter = 1: MAX_ITER
    unew ← random(Nn(u))
    Pold ← P(u), using Equations (1)–(7)
    Pnew ← P(unew), using Equations (1)–(7)
    ΔP ← Pnew – Pold
    t ← random[0, 1]
    If (t < exp(-ΔP/T))
        u ← unew
    End
    T ← βT
End

```

Fig. 1. The Simulated Annealing Algorithm

Denote the set of n -opt neighbor function of u as $N_n(u)$. The basic framework of the intelligent search for the joint power control and subcarrier assignment can be described as following. At first, a random value of u is chosen, then at each iteration, a value of $N_n(u)$ is select. If the value satisfies some heuristic conditions then update the value of u to the new value. The key here is instead of doing exhaustive search over the set of u , we only perform search on only a subset of it. Because of the limit of the space, we will not describe how Simulated Annealing and Tabu Search algorithms work. Interested readers can find the description of these algorithm in [7][8][11]. The Simulated Annealing and Tabu Search Algorithms are presented on Figures 1 and 2.

```

Initialization
    u  $\leftarrow$  random
    tabulist  $\leftarrow \emptyset$ 
For iter = 1 : MAX_ITER
    ubest  $\leftarrow$  random(Nn(u))
    Pbest  $\leftarrow$  P(unew)
    For each unew in Nn(u)
        Pnew  $\leftarrow$  P(unew)
        If Pbest > Pnew and unew  $\notin$  tabulist
            Pbest  $\leftarrow$  Pnew
            ubest  $\leftarrow$  unew
        End
    End
    u  $\leftarrow$  ubest
    if tabulist is full
        remove the oldest element from tabulist
    tabulist  $\leftarrow$  tabulist  $\cup$  u
End

```

Fig. 2. The Tabu Search Algorithm

3 Numerical Results

A random network of $K = 40$ users was generated. The code words $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K$ are randomly generated using ± 1 as the real and imaginary components of each code. Each code has the length $N = 64$. The target SINR is 5 dB. The number of subcarriers is $L = 10$. The channel is assumed to be Raleigh fading with equal pathloss.

The initial temperature of the Simulated Algorithm is $T=200$. The temperature control factor was chosen $\beta=0.99$. The length of the tabulist is 32. The total number of iterations is 500. Four algorithms was implemented, the Simulated Annealing, the Tabu Search algorithm, the random assignment algorithm and the greedy algorithm. In the random assignment algorithm, at each iteration, each node is assigned a random subcarrier while in the greedy search algorithm, on another hand, at each iteration, the algorithm looks at its neighbor set to find the one with smallest total transmit power and at the next iteration move to that solution.

Table 1. Total transmit power of the four algorithms

Algorithm	$L = 10, K = 40$	$L = 10, K = 50$	$L = 10, K = 60$
Simulated Annealing	50.1	67.3	71.0
Tabu Search	50.2	65.9	71.0
Random Search	195.6	250.1	288.7
Greedy	198.9	249.2	272.5

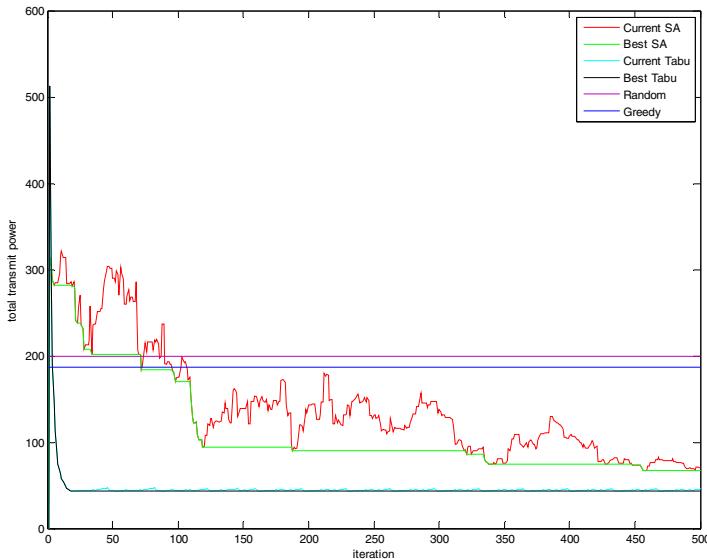


Fig. 3. The total transmit powers of the four algorithms

Figure 3. shows the total transmit power of the four algorithms. The Figure shows that the solution provided by Tabu Search is the best. The Simulated Annealing also provided good total transmit power while the random and the greedy search method do not provide good enough output.

It should be noted that the comparison between the Tabu search and the Simulated Annealing algorithms based on iteration indices is not fair because, at each step the Tabu Search algorithm require much more computation than that of the Simulated Annealing (including searching for all the best neighbors, and performing Tabu list manipulation). To make a fair comparison, we implemented all four algorithms in Matlab (version 7.6 running on Windows Vista). Each algorithm performed 10^7 flops (basic operation in Matlab) before producing the output. For each case, 100 random networks were generated and the results were averaged. The data on Table 2 showed that Simulated Annealing and Tabu Search produced comparable outputs and the total transmit powers are much smaller than those of the random and the greedy search.

4 Conclusion

In this paper we proposed a new framework for the problem of joint power control and subcarrier allocation for MC-CDMA systems. The key idea is to define the subcarrier mapping function from the user set to the subcarrier set and its neighbor set. It is shown that the framework can be easily applied to Simulated Annealing and Tabu Search algorithms. It is obvious that by modifying the n-opt operation the framework can be applied to other intelligent search algorithms such as Genetic Algorithm and Ant Colony Optimization. Numerical results showed that these algorithms provide much better solutions than conventional methods.

References

1. Verdu, S.: Multiuser Detection. Cambridge University Press, Cambridge (1998)
2. Sampath, A., Kumar, P.S., Holtzman, J.M.: Power control and resource management for a multimedia CDMA wireless system. In: Proc. IEEE PIMRC, vol. 1 (1995)
3. Viterbi, A.M., Viterbi, A.J.: Erlang capacity of a power controlled CDMA system. IEEE Journal on Selected Areas in Communications 11(6) (1993)
4. Alpcan, T., Basar, T., Srikant, R., Altman, E.: CDMA Uplink Power Control as a Noncooperative Game. Wireless Networks 8(6) (2002)
5. Gilhousen, K.S., Jacobs, I.M.: On the capacity of a cellular CDMA system. IEEE Transactions on Vehicular Technology 40(2) (1991)
6. Liu, Z., El Zarki, M.: SIR-based call admission control for DS-CDMA cellular systems. IEEE Journal on Selected Areas in Communications 12(4) (1994)
7. Kirkpatrick, C.D., Gelatt, M.P.V.: Optimization by Simulated Annealing. S - Science, 1983 - sciencemag.org (1983)
8. Aarts, E., Korst, J., Abu-Mostafa, Y.S., Jacques, J.S.: Simulated Annealing and Boltzmann Machines. Mathematical Programming Technical Report, Computer Sciences Dept., Univ. of Wisconsin (1989)
9. Glover, F., Laguna, M.: Tabu Search. Springer, Heidelberg (1998)
10. Tse, D.: Fundamental of Wireless Communications. Cambridge University Press, Cambridge (2005)
11. Osman, I.H., Kelly, J.P.: Meta-Heuristics: Theory and Applications. Springer, Heidelberg (1996)

Domain-Independent Error-Based Simulation for Error-Awareness and Its Preliminary Evaluation

Tomoya Horiguchi¹ and Tsukasa Hirashima²

¹ Faculty of Maritime Sciences, Kobe University,
5-1-1, Fukaeminami, Higashinada, Kobe, Hyogo, 658-0022 Japan
horiguti@maritime.kobe-u.ac.jp

² Department of Information Engineering, Hiroshima University,
1-4-1, Kagamiyama, Higashihiroshima, Hiroshima 739-8527 Japan
tsukasa@isl.hiroshima-u.ac.jp

Abstract. Error-based Simulation (EBS) is a framework for assisting a learner to become aware of his errors. It makes a simulation based on his erroneous hypothesis to show *what unreasonable phenomena would occur if his hypothesis were correct*, which has been proved effective as counterexamples to cause cognitive conflict. In making EBS, it is necessary (1) to make a simulation by dealing with a set of inconsistent constraints because erroneous hypotheses often contradict the correct knowledge, and (2) to estimate the 'unreasonableness' of phenomena in a simulation because it must be recognized as 'unreasonable.' We previously proposed a technique (called 'Partial Constraint Analysis (PCA)') for making EBS based on any inconsistent simultaneous equations, and a set of domain-independent heuristics to estimate the 'unreasonableness' of physical phenomena. In this paper, we describe a prototype EBS-system implemented by using these, and show the results of preliminary test which verified the usefulness of our method.

Keywords: intelligent tutoring system, simulation-based learning environment, learner's error, counterexample, truth maintenance system.

1 Introduction

The critical issue in assisting constructivist learning is to provide a learner with feedback which causes cognitive conflict when he makes errors [8]. We call this 'the assistance of error-awareness,' and think there are two kinds of methods for it. The one is to show the correct solution and to explain how it is derived. The other is to show *an unreasonable result which would be derived if his erroneous idea/solution were correct*. We call the former 'indirect error-awareness,' and the latter 'direct error-awareness.' [3]

Usual simulation-based learning environments (SLEs, for short) [12] give the assistance of indirect error-awareness because they always provide the correct solution (i.e., correct phenomena) a learner should accept finally. The understanding by such assistance is, however, 'extrinsic' because they show only the same physically correct phenomena whatever erroneous idea a learner has. In addition, in usual SLEs, a learner must translate his (erroneous) hypothesis into

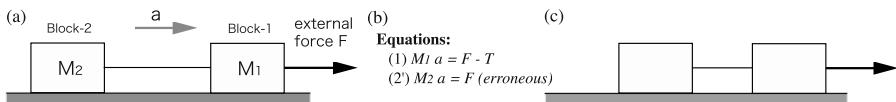


Fig. 1. An example of EBS

the input which doesn't violate the constraints used by a simulator (e.g., even if he thinks 'the quantities x and y are proportional,' he can only increase x and observe y , instead of inputting the equation ' $y = cx (c > 0)$ '). This makes it difficult to identify what kind of phenomena a learner predicts. It is, therefore, difficult to estimate the 'seriousness' of the difference between the correct phenomena and his prediction.

Error-based Simulation (EBS, for short) [2] is a framework for assisting such direct error-awareness in SLEs. It makes simulations based on the erroneous ideas/solutions externalized by a learner (we call them 'erroneous hypotheses'), which results in unreasonable (unacceptable) phenomena and makes him be aware of his errors. For example, consider the mechanical system shown in Fig.1a. Assume that a learner set up equations of motion shown in Fig.1b for this system. The EBS made based on this 'erroneous hypotheses' is shown in Fig.1c, in which the string connecting the two blocks shrinks. These unnatural phenomena become counterexamples to a learner's erroneous ideas and are expected to motivate him to reflect on his solutions. We have developed a few EBS-systems for such mechanics problems [4,6] and tested their effectiveness in a laboratory experiment (subjects were university students) [4] and a field test (subjects were junior high school students) [6]. In both cases, it has been proved that EBSs caused strong cognitive conflict and lead learners to a deeper understanding.

For designing EBS-systems, in general, two technical issues must be addressed. (1) The representation of erroneous hypotheses often contradicts the constraints necessary for making a simulation (i.e., the representation of correct knowledge of the domain). (2) The result of a simulation must be recognized to be 'unreasonable' by a learner. As for (1), if a contradiction is detected, some of the correct constraints should be relaxed (i.e., deleted) to make the rest consistent (i.e., EBS shows that if a learner's erroneous hypothesis were correct, it would be inevitable for some correct constraints to be violated). The module which deals with such inconsistency is called the 'robust simulator.' As for (2), while the phenomena in EBSs are physically impossible because they include an erroneous hypothesis (and/or some correct constraints are deleted), a learner who has only incomplete knowledge of the domain doesn't always recognize their 'unreasonableness.' It is, therefore, necessary to estimate the 'unreasonableness' of phenomena in EBS by using explicit criteria.

The techniques used in the EBS-systems previously developed, however, were domain-dependent. That is, they could deal with only a limited class of errors by a learner, and used domain-specific heuristics to avoid contradiction in calculation. We, therefore, proposed a domain-independent technique which can deal with any erroneous simultaneous equations/inequalities (which are the powerful means of externalizing scientific ideas) to make EBS. It is called 'Partial Constraint Analysis (PCA, for short)' [5], which detects and eliminates contradictions in a set of constraints given by simultaneous equations/inequalities. We

also proposed a set of heuristics to estimate the 'unreasonableness' of phenomena in EBS in a general way [5]. It describes the meaning of typical equations/inequalities of physical systems and is used for predicting what kind of physical phenomena would occur if they were violated. The usefulness of these methods was suggested with some examples.

In this paper, we show the results of preliminary test which verified the usefulness of PCA and the heuristics empirically. Some technical issues concerning the prototype system implemented for the test are also discussed. We think these results are encouraging in applying EBS to other domains.

2 PCA: Partial Constraint Analysis

In this section, we outline PCA (see [5] for more detail). In previous EBS-systems, the procedure for avoiding contradiction was specified before calculation by utilizing the assumption of the domain. Under general conditions, however, it is necessary to detect the cause of contradiction in a set of constraints explicitly and to eliminate it appropriately. PCA is a method for doing such a calculation. It can deal with simultaneous equations/inequalities which may include contradiction.

2.1 The Algorithm

PCA first constructs the 'Constraint Network' (CN, for short) of given simultaneous equations S , then searches for the consistent part of it (we call it 'Partial Constraint Network,' PCN, for short). CN is a graph structure which represents the dependency between equations and variables in S . CN has two types of nodes: equation node (e-node) and variable node (v-node). An e-node stands for an equation and a v-node stands for a variable in S . A v-node which stands for an exogenous variable is called ex-v-node. Examples of CN are shown in Fig.2c and Fig.3c.

Taking an ex-v-node (or a v-node which is given its temporary value) as an initial PCN, PCA then extends it by adding the nodes step by step to which the value can be regularly propagated. When S is consistent, the value of each v-node is uniquely calculated by a unique path which propagates the given values of ex-v-nodes to it (If CN has a loop, the values of v-nodes in the loop are determined simultaneously by solving the simultaneous equations in it). To each v-node in PCN, the method for calculating its value is attached symbolically using the values of ex-v-nodes and e-nodes on the path of the propagation (it is called 'calc-method,' for short).

When S is inconsistent, PCA meets the following irregularities and resolves them:

- (under-constraint) There are some e-nodes in each of which the sufficient values of v-nodes aren't supplied to calculate the value of a v-node. In other words, there are some v-nodes the values of which can't be determined by any path. In such a case, PCA gives them temporary values and continues the calculation by using the values (a v-node given its temporary value is called a 'dummy').

- (over-constraint) There are some v-nodes each of which has more than one path to determine its value. In other words, there are some e-nodes which have no simultaneous solution. In such a case, PCA deletes one of the e-nodes responsible for the contradiction from PCN to make the rest consistent.

The procedure above is continued until no propagation of values is possible any more. Contradiction occurs when PCA meets over-constraint. In order to detect a contradiction and identify the e-nodes responsible for it, PCA cooperates with TMS (Truth Maintenance System) [1]. That is, it makes a TMS maintain the justification of calc-method of each v-node. The algorithm with a basic JTMS (Justification-based TMS) [1] is described in [5]. By introducing a TMS, all the possible PCNs can be obtained independently of the choice of the initial PCN.

2.2 Other Mechanisms for Dealing with Learners' Erroneous Ideas

From a technical viewpoint, there have been proposed several mechanisms based on (A)TMS ((Assumption-based) TMS) for dealing with an inconsistent set of constraints (which are the representations of a learner's erroneous idea/solution and the correct knowledge of the domain) in the literature on learner modelling [11]. Though their ability in nonmonotonic reasoning is (more than) equivalent to our method, their functions are different (because of the difference of purposes). While the model-based cognitive diagnosis [9] can identify the erroneous part of a learner's knowledge, it can say nothing as to what would happen based on the erroneous knowledge. While SMIS [7] can reason the consistent part of a learner's knowledge (i.e., the learner model) and execute it to show what would result from it, it can say nothing as to the 'unreasonableness' of the result (by itself). In addition, though SMIS excludes some constraints to make a learner model consistent, it can't control the 'unreasonableness' by choosing the constraints to be excluded because it doesn't know the meanings of them. The robust simulator addresses these issues.

3 Heuristics for Estimating the 'Unreasonableness'

3.1 Four Heuristics for Deleting Constraints

When erroneous simultaneous equations/inequalities are unsolvable, PCA outputs a consistent PCN in which a (set of) equation(s) is deleted from them. The choice of equation(s) to be deleted considerably influences what unreasonable phenomena would occur in simulation. From the viewpoint of direct error-awareness, we proposed four domain-independent heuristics in making such a choice in order to estimate the 'unreasonableness' of EBS [5]. In this section, we outline them.

(H1) Don't delete the erroneous equations. Since an erroneous equation reflects a learner's error, it must not be deleted. Only when there is more than one erroneous equation and they contradict each other, some of them may be deleted.

(H2) Delete the equations which represent the topic of learning prior to others. When the topic of learning is a relation between some physical constants/variables (e.g., physical law/principle), it is useful to delete the equation which represents it because the phenomena in which the relation doesn't hold focus on the topic.

(H3) *Delete the equations/inequalities which describe the values of physical constants or the domains of physical variables prior to others.* The equations/inequalities which describe the values of physical constants or the domains of physical variables often represent the most basic constraint of the domain, such as the meaning of the constants/variables. The phenomena in which these constraints are violated, therefore, are easily recognized as 'unreasonable.'

(H4) *Delete the fundamental circuit equations and cut set (incidence) equations of the system prior to others.* In physical systems, fundamental circuit equations of across variables and cut set (incidence) equations of through variables [10] often represent the most basic constraints of the domain, such as the conservation of basic physical amounts, or the relations between components of the system to be held. The phenomena in which these constraints are violated, therefore, are easily recognized as 'unreasonable.'

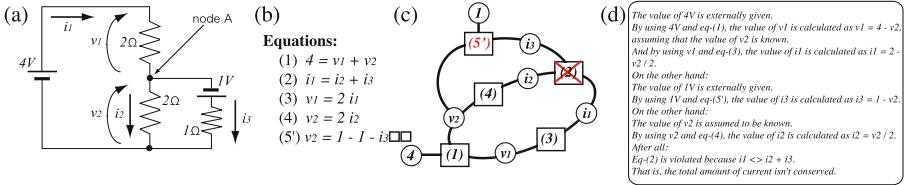
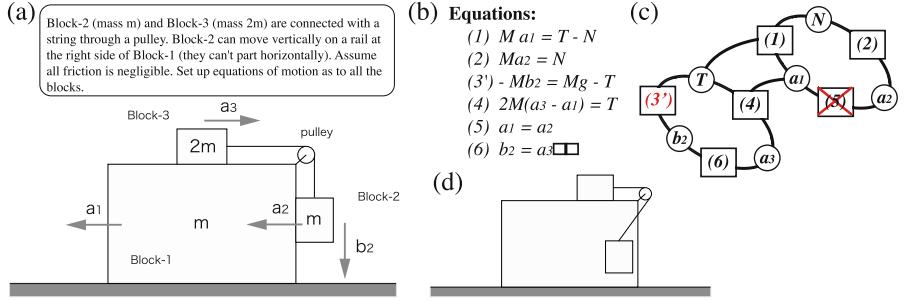
3.2 Examples of Making EBS by Using the Heuristics

We implemented a prototype robust simulator which makes EBS by using PCA and the heuristics above. Examples in elementary electric circuits and mechanics are also prepared in which a learner is asked to set up equations for a system. In this section, we describe its implementation and illustrate how it works.

How to interpret equations by a learner. The interpreter of equations by a learner in our prototype system is very simple. It deals with equations/inequalities which include only polynominal expressions and are consistent in physical dimensions. All labels (i.e., names) of physical constants/variables are given in each problem. It judges whether an equation/inequality is correct or not by matching it with the template of correct solution (a correct equation/inequality and some variations of it). By using similar templates, it also identifies the types of equations/inequalities (i.e., those of the topic of learning, those of the values/domains of physical constants/variables, the fundamental circuit or cut set (incidence) equations). In order to prevent a learner from omitting some equations/inequalities or transforming their expressions in advance, a learner is asked to use all of the constants/variables given in each problem. In this way, all the equations/inequalities which represent the basic constraints of the domain are set up.

Generation of explanation. By tracing the dependency network of justifications made by PCA, the system can make an explanation of why the simultaneous equations aren't (or are) solvable. By using the heuristics above, it can also explain how unnatural phenomena will occur in the EBS. A dependency network and the meaning of a deleted equation are translated into natural language by using a simple template of explanation.

An example in an electric circuit. As for the electric circuit shown in Fig.2a, assume that a learner set up the (erroneous) equations shown in Fig.2b. These simultaneous equations are unsolvable because the two loops in their constraint network (i.e., the loop with variables v_1, v_2, i_1 and i_2 , and the loop with variables v_2, i_2 and i_3) are simultaneously unsolvable (Fig.2c). The robust simulator, therefore, tries to delete some of the equations in these loops (i.e., equations

**Fig. 2.** An example in electric circuit**Fig. 3.** An example in mechanics

(1), (2), (3), (4) and (5')). According to (H1), equation (5') is not an option. Since equations (1), (3) and (4) are fundamental circuit equations and equation (2) is an incidence equation, equation (2) is deleted according to (H4) (in this case, (H2) and (H3) are inapplicable). The calculation by using the rest (i.e., equations (1), (3), (4) and (5')) yields an 'unreasonable' phenomenon in which the total amount of electric current at node A isn't conserved (i.e., $i_1 = 2.25(A)$, $i_2 = -0.5(A)$ and $i_3 = 1.5(A)$). Fig.2d shows the explanation made by the system.

An example in mechanics. As for the mechanical system shown in Fig.3a, assume that a learner set up the (erroneous) equations shown in Fig.3b. These simultaneous equations are unsolvable because the two loops in their constraint network (i.e., the loop with variables a_3, b_2 and T , and the loop with variables a_1, a_2 and N) are simultaneously unsolvable (Fig.3c). The robust simulator, therefore, tries to delete some of the equations on these loops (i.e., equations (1), (2), (3'), (4), (5) and (6)). According to (H1), equation (3') is not an option. Since equations (1), (2) and (4) are incidence equations and equations (5) and (6) are fundamental circuit equations, equation (5) is, for example, deleted according to (H4) (in this case, (H2) and (H3) are inapplicable). The calculation by using the rest (i.e., equations (1), (2), (3'), (4) and (6)) yields an 'unreasonable' phenomenon in which the relative velocity between Block-1 and Block-2 isn't held (i.e., $a_1 = g/4$, $a_2 = 9g/4$, $a_3 = b_2 = 3g/2$, $T = 5Mg/2$ and $N = 9Mg/4$), that is, these blocks overlap each other (Fig.3d).

4 Preliminary Test

A preliminary test was made for verifying the usefulness of our method by using the prototype system.

We prepared four problems in elementary physics, three of which are in mechanics (problems 1-3) and one in electric circuit (problem 4). We also prepared seven examples of erroneous solutions for them, three of which are for problem 1, 2 and 4 respectively and four for problem 3. These examples (i.e., erroneous equations) were made by combinating the errors frequently observed in learners' erroneous equations (e.g., missing/extraneous terms, incorrect sign of vector quantities, erroneous use of trigonometric functions). EBSs and explanations for them were generated by using the prototype system and shown to ten subjects who were (under)graduate students majoring in engineering. From the viewpoint of a 'tutor,' they were asked to judge (1) whether such erroneous equations probably occur when a learner tries to solve these problems, and (2) whether the EBSs and explanations are effective in assisting a learner who made such errors.

Tab.1 shows the EBSs generated by the prototype system for the erroneous solutions in problems 1-4. By using PCA and the heuristics, EBSs can be generated for various types of erroneous equations. As for these EBSs and explanations, subjects' response to question (2) is shown in Tab.2 (in which, only the responses of subjects who gave positive answers to question (1) (i.e., who chose '(the error is) very probable' or 'probable') are aggregated).

Tab.2 shows that most subjects agreed the effectiveness of EBSs and the explanations in almost all the cases (we leave the cases of erroneous equations 1 and 2 in problem 3 out of consideration because they have few samples). Most of the subjects who didn't agree pointed out that the explanations were difficult

Table 1. Problems, errors and generated EBSs

Problem/error	Error(s) in equations	Applied heuristics (deleted eq./ineq.)	Generated EBS
problem-1	erroneous term	H4(fundamental circuit eq.)	a string shrinks
problem-2	trigonometric function	H2(conservation of energy)	energy isn't conserved
problem-3	error-1/EBS-1	H4(fundamental circuit eq.)	blocks overlap
	error-1/EBS-2	H4(fundamental circuit eq.)	a string shrinks
	error-2	H4(fundamental circuit eq.)	connected blocks part
	error-3	H4(fundamental circuit eq.)	a string extends
	error-4	H4(fundamental circuit eq.)	a string shrinks
problem-4	sign of vector quantity	H4(incidence eq.)	current isn't conserved

Table 2. Result of the questionnaire

	very effective	effective	in balance	less effective	not effective
problem-1	1	6	1	0	0
problem-2	0	4	2	1	0
problem-3	1	1	0	0	0
	1	1	0	0	0
	2	1	1	0	0
	0	4	3	0	0
	3	4	1	1	0
problem-4	0	4	2	1	0

to understand. In addition, as to problem 2, they also pointed out that the unnaturalness of the phenomenon (i.e., total energy isn't conserved) was hard to understand because it wasn't sufficiently visualized by EBS (which shows the motion of objects by animation). Even these subjects, however, agreed the aim of generated EBS itself and suggested that these could be improved by using a better template and by choosing the media suitable for the unreasonableness to be visualized (the latter is addressed in [3]).

We, therefore, think the results of this preliminary test suggest the usefulness of our method.

5 Conclusion

In this paper, we presented the results of preliminary test which verified the usefulness of the domain-independent method of making EBS. We think these results are encouraging in applying EBS to other domains for assisting direct error-awareness.

References

1. Forbus, K.D., de Kleer, J.: *Building Problem Solvers*. MIT Press, Cambridge (1993)
2. Hirashima, T., Horiguchi, T., Kashihara, A., Toyoda, J.: Error-based Simulation for Error-Visualization and Its Management. *Int. J. of Artificial Intelligence in Education* 9, 17–31 (1998)
3. Hirashima, T., Horiguchi, T.: Error-Visualization for Learning from Mistakes. *Transactions of Japanese Society for Information and Systems in Education* 21, 178–186 (2004) (in Japanese)
4. Horiguchi, T., Hirashima, T., Okamoto, M.: Conceptual Changes in Learning Mechanics by Error-based Simulation. In: Proc. of ICCE 2005, pp. 138–145 (2005)
5. Horiguchi, T., Hirashima, T.: Robust Simulator: A Method of Simulating Learners' Erroneous Equations for Making Error-based Simulation. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006. LNCS*, vol. 4053, pp. 655–665. Springer, Heidelberg (2006)
6. Horiguchi, T., Imai, I., Toumoto, T., Hirashima, T.: A Classroom Practice of Error-based Simulation as Counterexample to Students Misunderstanding of Mechanics. In: Proc. of ICCE 2007, pp. 519–526 (2007)
7. Ikeda, M., Mizoguchi, R., Kakusho, O.: Student Model Description Language SMDL and Student Model Inference System SMIS. *IEICE Transactions J72-D-II*(1), 112–120 (1989) (in Japanese)
8. Perkins, H.J.: *Learning From Our Mistakes: Reinterpretation of Twentieth Century Educational Theory*. Greenwood Press, Westport (1984)
9. Self, J.: Model-Based Cognitive Diagnosis. *User-Modeling and User-Adapted Interaction* 3, 89–106 (1993)
10. Shearer, J.L., Murphy, A.T., Richardson, H.H.: *Introduction to System Dynamics*. Addison-Wesley Publishing Company, Reading (1971)
11. Webb, G., Pazzani, M., Billsus, D.: Machine Learning for User Modeling. *User-Modeling and User-Adapted Interaction* 11, 19–29 (2001)
12. Wenger, E.: *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufmann, San Francisco (1990)

A Characterization of Sensitivity Communication Robots Based on Mood Transition

Chika Itoh, Shohei Kato, and Hidenori Itoh

Dept. of Computer Science and Engineering, Graduate School of Engineering,
Nagoya Institute of Technology,
Gokiso-cho Showa-ku Nagoya 466-8555 Japan
`{titoh,shohey,itooh}@juno.ics.nitech.ac.jp`

Abstract. Changing the mood transition of robots can change the perceptions people have of their characteristics. We present an emotion generation model that represents a robot's psychological state of mind. This model can assess the robot's individuality through mood transitions. We also report an experiments of emotional conversation with a robot that had this model installed.

1 Introduction

For a robot to live and to communicate with people in a natural way, it requires its own personality or individuality. Otherwise, it would seem awkward and out of place. We present an emotion generation model that represents a robot's psychological state of mind. This model can assess the robot's individuality through mood transitions. Mood is defined as a weak but relatively lasting affective state. It is an allowed or rejected expression of emotions [1,2,3]. Human emotion is a comparatively intense, transient affective state, which is produced rapidly and is generally accompanied by clear expressive change [4].

We divide the robot's affective state into mood and emotion. Mood is a long lasting affective state that influences emotion. Emotion is a short lasting affective state that is clear and shown directly through expressions. Emotion is assessed from the amount of pleasure-displeasure of the robot and occurs in response to the influence of mood. Mood dynamically changes with the accumulation of past emotions. Changing the mood transition of robots can produce various expressions of personality as well as individuality in them that enables the robots to demonstrate a wide range of behaviors.

2 A Sensitivity Communication Robot Ifbot

A novel robot, the Ifbot, can communicate with humans through expressive and engaging conversations and emotional expressions that have been developed by our industry-university joint research project [5]. The Ifbot can converse with

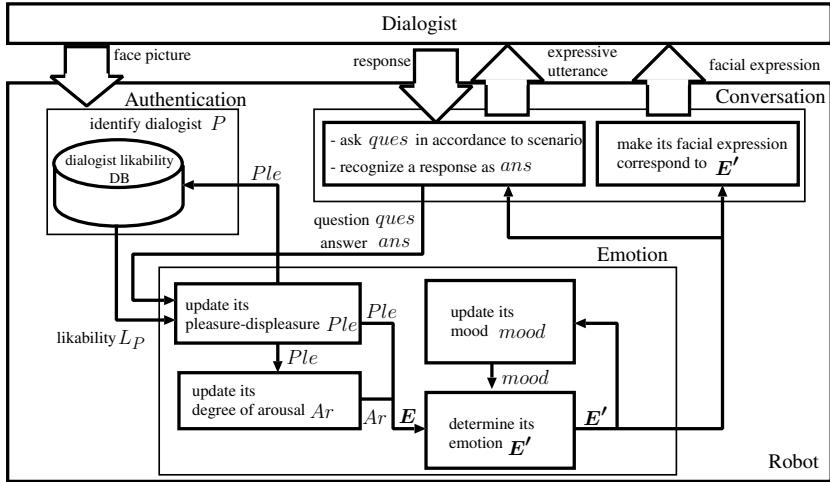


Fig. 1. Emotion generation model

a person through fundamental voice recognition and voice synthesis engines. It can also communicate with a person, while showing its “emotions” through facial expression mechanisms and gestures.

3 Emotion Generation Model

Figure 1 shows the overview of our emotion generation model. This model controls robot’s emotions based on its mood and what is talked. The interior of the robot consists of three main processes: authentication, emotion, and conversation. In the authentication process, a robot identifies a dialogist P and refers to the dialogist likability DB for likability L_P to P held in old conversations. In the emotion process, the robot determines its emotion E . By adding the influence of the robot’s mood md to this, its emotion E' is determined. In the conversation process, the robot asks a question, recognizes the response of P , and expresses expressions corresponding to E' .

4 Emotion Process

We use Russel’s circumplex model of affect[6] for determining emotion. On the basis of this model, we built an emotional space of the robot this was defined in terms of two orthogonal dimensions, pleasure-displeasure and degree of arousal. We calculate the robot’s emotion $E = (r, \theta)$ in plotting the pleasure-displeasure Ple and the degree of arousal Ar . Here, (r, θ) is the polar coordinate expression of the position vector (Ple, Ar) in the robot’s emotional space. Please refer to [7][8] for the calculations of Ple and Ar from conversations.

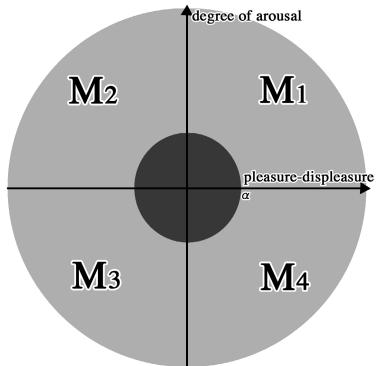


Fig. 2. Allowed domain of emotions affected by mood

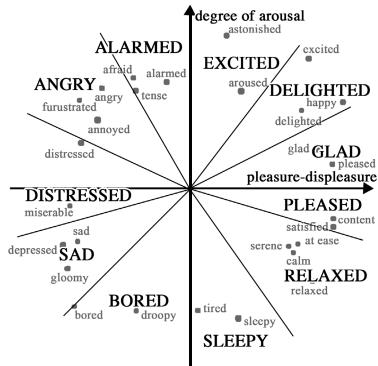


Fig. 3. Classification of emotional space of robot's expressions: eleven domains with typical eleven emotions

4.1 Mood

In this paper, we account for the robot's mood md as a transition model that changes between domains by accumulation of past emotions \mathbf{E}' . The emotional space is divided into four domains with the axis of pleasure-displeasure and degree of arousal. Each domain is considered as one mood domain and is called M_1, M_2, M_3, M_4 , beginning at the first quadrant. The robot's mood md changes between these domains.

The robot has stacks for the changes in each mood, and the emotion $\mathbf{E}' = (r', \theta')$ calculated in Section 4.2 is accumulated in a corresponding stack. The robot updates stacks for every question and answer set by accumulating the quantity proportional to the magnitude r' in a stack that is determined by the direction θ' . The robot changes its md when the stack is over its threshold. Suppose that the robot's current md is M_i . For example, if the stack corresponding to M_j exceeds its threshold Th_{ij} , md will change from M_i to M_j . It becomes easy to change when Th is small. We give various expressions of personality to the robot by changing Ths . Because we consider that the influence on mood weakens like the emotions accumulated in the past, the stacks decrease little by little.

4.2 Emotion

The intensity of the robot's emotion is affected by its mood. We set up an allowed domain of the emotion corresponding to each mood. The robot changes the intensity of its emotion according to this allowed domain. The allowed domain of the emotions affected by the robot's md is shown in Figure 2. The logical sum of the domain that is composed of one quadrant and the internal domain of the inner circle of radius α is one mood domain. The coefficient α ($0 < \alpha \leq 1$) shows the strength of the emotions affected by md .

The emotion $\mathbf{E}' = (r', \theta')$ is calculated from the emotion $\mathbf{E} = (r, \theta)$ and the allowed domain (Figure 2) of the emotions affected by the mood md . The intensity

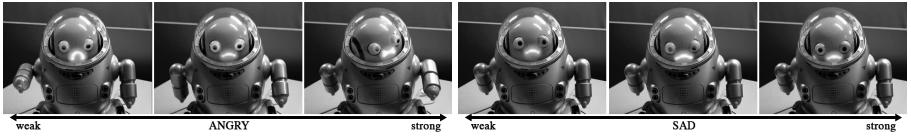


Fig. 4. Examples of facial expressions of Ifbot

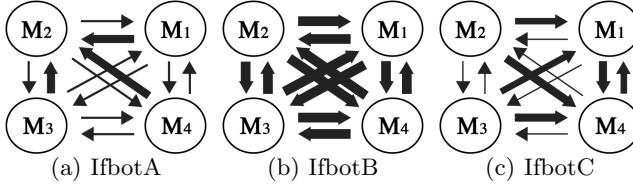


Fig. 5. Threshold Ths of each Ifbot

of emotion is influenced by md , but the kind of emotion is not influenced. If \mathbf{E} is in the domain corresponding to md , The intensity of \mathbf{E} is kept stable by maintaining the magnitude r constant. In contrast, if \mathbf{E} is not in the domain corresponding to md , the intensity of \mathbf{E}' is reduced by multiplying r and α .

5 Experiment

Conversation experiments were conducted after our emotion generation model was installed in an Ifbot. The kinds of expressions that the robot was capable of are shown in Figure 3. We classified the robot's emotional space this figure into eleven domains according to the kind θ' of the robot's emotion \mathbf{E}' , and we matched the typical emotions in each domain with the expressions. We divided the intensity of the robot's emotion into three steps and matched the different expressions with the corresponding in each step. In these experiments, we used the facial expressions that were input into the Ifbot beforehand. Examples of the facial expressions corresponding to each emotion expressed in the experiments are shown in Figure 4.

We evaluated the psychological impact that the robots made on the dialogist. A subjective evaluation on the emotional changes was conducted after each dialogist had conversations with IfbotA, IfbotB and IfbotC, each of which had a corresponding character image, and an adjustment in their thresholds Ths , IfbotN, in the original system, was also used. The Ths of IfbotA, IfbotB, and IfbotC are shown in Figure 5. In this figure, the size of Th is indicated by the thickness of the arrow. If Th is small, the arrow is thick and the robot easily changes its mood. We gave IfbotA a “short tempered” characteristic. Its mood easily changes to M_2 . We gave IfbotB an “arbitrary” characteristic. Its mood easily changes. And gave IfbotC a “cheerful and peaceful” characteristic. Its mood easily changes to M_1 and M_4 .

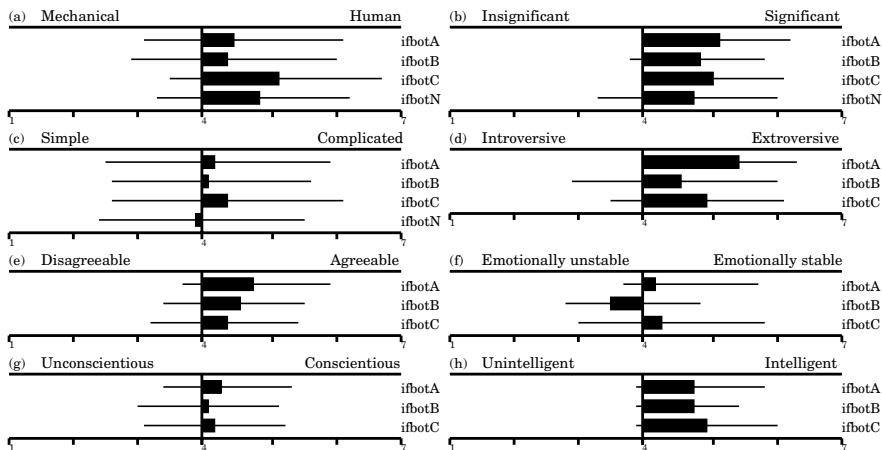


Fig. 6. Results of sensitivity evaluation

In these experiments, we used the semantic differential [9] for evaluating sensitivity. Thirty subjects observed conversations with the four Ifbots and answered questionnaires after each conversation. We determined the effect of our model by graphing out the average and standard deviation of the evaluations (1-7) for four Ifbots, item by item. We used the Mann-Whitney U test as an inferential test of the four Ifbots. The significance level of the test was set to 0.05. The results are shown in Figure 6. IfbotN was not evaluated on the big five[10] (Figure 6(d)-Figure 6(h), which is widely accepted in personality theory, because these items were evaluated only in the Ifbots with special characteristics.

On “mechanical - human”, shown in Figure 6(a), although IfbotB was slightly low, every Ifbot was rated as comparatively “human.” On “insignificant - significant”, shown in Figure 6(b), and “simple - complicated”, shown in Figure 6(c), the evaluation of each characterized Ifbot varied, but compared with IfbotN, each of these Ifbots was rated as more “significant” and “complicated”. This is because giving moods to Ifbot can make its facial expressions more meaningful and can facilitate a partner perceiving it to be more complex. On “introversive - extroversive” shown in Figure 6(d), IfbotA was rated as comparatively “extroversive” in the three Ifbots. On “disagreeable - agreeable” shown in Figure 6(e), “agreeable” was supported in order of IfbotA, IfbotB, and IfbotC. On “emotionally unstable - emotionally stable”, shown in Figure 6(f), IfbotB was rated as more “emotionally unstable” than IfbotA and IfbotC, which were rated as being comparatively “emotionally stable”. This is because since in IfbotB, the thresholds for mood translation Th_s were set low, IfbotB changes its mood more easily than IfbotA and IfbotC. We found a statistically significant difference between IfbotA and IfbotB and between IfbotB and IfbotC. This shows that the personality of our robots can be expressed, whether it is emotionally stable or not. On “unconscious - conscientious”, shown in Figure 6(g), and on “unintelligent - intelligent”, shown in Figure 6(h), the perceived differences were not significant.

6 Conclusion

We proposed an emotion generation model for robots so that they could provide more human-like communications. This model can give the robot personality through mood transitions this change with the accumulation of past emotions. Conversation experiments were conducted after this model was installed in an Ifbot. Our experimental results showed that personality could be effectively expressed by changing the characteristic parameters on mood transitions.

In this paper, we reported on experiments of emotional conversations with three different characterizations in Ifbot. However, that is not enough to show the effectiveness of this model. In future work, we will put robots that have more personality variations through many more tests of this model.

Acknowledgments

This work was supported in part by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research under grant #17500143 and #20700199, and by the Tatematsu Foundation.

References

1. Moore, B.E., Fine, B.D.: PSYCHOANALYTIC TERMS & CONCEPTS by The American Psychoanalytic Association. Yale University Press (1990)
2. Lofgren, L.B.: Psychoanalytic theory of affects. *Journal of the American Psychoanalytic Association* 16, 638–650 (1970)
3. Weinshel, E.M.: Some Psychoanalytic Considerations on Moods. *The International Journal of Psycho-analysis* 55, 313–320 (1968)
4. Strongman, K.T.: The Psychology of Emotion. Everyday Life to Theory (2003)
5. Business Design Laboratory Co. Ltd. Communication Robot Ifbo,
<http://www.business-design.co.jp/product/001/index.html>
6. Russel, J.A.: A circumplex model of affect. *Journal of Personality and Social Psychology* 39, 1161–1178 (1980)
7. Takeuchi, S., Sakai, A., Kato, S., Itoh, H.: An Emotion Generation Model Based on the Dialogist Likability for Sensitivity Communication Robot. *Journal of the Robotics Society of Japan* 25(7), 103–111 (2007) (in Japanese)
8. Senoo, Y., Kato, S., Itoh, H.: An emotion control method based on likability for sensibility communication robot. In: Proceedings of The 8th Annual Conference of JSKE 2006, p. 182 (2006) (in Japanese)
9. Osgood, C.E., Tannenbaum, P.H., Suci, G.J.: The Measurement of Meaning. University of Illinois Press, Urbana (1957)
10. Goldberg, L.R.: An alternative "description of personality": The big-five factor structure. *Journal of Personality and Social Psychology* 59, 1216–1229 (1990)

Recommendation Algorithm for Learning Materials That Maximizes Expected Test Scores

Tomoharu Iwata¹, Tomoko Kojiri²,
Takeshi Yamada¹, and Toyohide Watanabe²

¹ NTT Communication Science Laboratories,
2-4, Hikaridai, Seika-cho, “Keihanna Science City,” Kyoto, 619-0237, Japan
{iwata,yamada}@cslab.kecl.ntt.co.jp

² Graduate School of Information Science, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya, 464-0803, Japan
{kojiri,watanabe}@watanabe.ss.is.nagoya-u.ac.jp

Abstract. We propose a recommendation algorithm for learning materials that enhances learning efficiency. Conventional recommendation methods consider user preferences and/or levels, but they do not directly consider the learning efficiency. With our method, the learning efficiency is quantified by the expected improvement in the test score, and materials are recommended so as to maximize this expected improvement. The expected improvement is calculated with logistic regression models that employ the user’s test result obtained before learning as input. Experimental results using fill-in-the-blank exercises for English learning show that our method yields major improvements in performance compared with random material recommendation.

Keywords: e-learning, personalization, adaptive tutoring systems, logistic regression.

1 Introduction

With the rapid growth of network and database technologies, e-learning systems have become widely used in various domains. With e-learning systems, the recommendation of suitable materials or exercises for each user is important because users have different learning levels, prior knowledge, and goals. In a textbook, the sequence of materials and exercises is fixed. This is very inefficient because some users find themselves undertaking exercises that are too easy or reading materials that are not examined in the target test.

In this paper, we propose a recommendation algorithm for learning materials that directly enhances learning efficiency. Although many personalized learning material recommendation algorithms have been proposed, they do not directly maximize learning efficiency. Instead, they consider user preferences or/and levels [1,2,3,4]. With our method, the learning efficiency is quantified by the expected

improvement in the test score, and materials are recommended so as to maximize the expected improvement. Intuitively speaking, our method recommends materials so that users provide correct answers to questions that they answered incorrectly prior to learning.

The expected improvement is calculated by logistic regression models [5] using the questions that the user answered incorrectly before learning as input. By training the logistic regression models using the learning log data and user test results, we can automatically extract information about which learning materials contribute to improvements in the test score. Since our method does not require meta data of the materials, it is applicable to any course.

2 Proposed Method

2.1 Preliminaries

Let x_i and y_i be variables that represent whether question i is correctly or incorrectly answered before and after the learning phases, respectively, as follows:

$$x_i = \begin{cases} 1/-1 & \text{if question } i \text{ is correctly/incorrectly answered before learning,} \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

$$y_i = \begin{cases} 1/-1 & \text{if question } i \text{ is correctly/incorrectly answered after learning,} \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where $x_i = 0/y_i = 0$ means question i has not been answered before/after the learning phase. The results of the set of test questions \mathbf{V} before and after the learning phase are represented by vectors $\mathbf{x} = (x_i)_{i \in \mathbf{V}}$ and $\mathbf{y} = (y_i)_{i \in \mathbf{V}}$, respectively.

Let z_j be a variable that represents whether material j is recommended in the learning phase as follows:

$$z_j = \begin{cases} 1 & \text{if material } j \text{ is recommended in learning,} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The recommended materials are represented by a vector $\mathbf{z} = (z_j)_{j \in \mathbf{M}}$, where \mathbf{M} represents a set of learning materials.

2.2 Recommendation Algorithm

The goal of our method is to select materials to be used in the learning phase from a set of materials in order to enhance learning efficiency. Learning efficiency is quantified by the expected improvement in the test score.

The expected improvement in the test score given recommended materials \mathbf{z} is written as follows:

$$E(\mathbf{z}) = \sum_{i \in \mathbf{V}} P(i)P(x_i = -1)P(y_i = 1|x_i = -1, \mathbf{z}), \quad (4)$$

Table 1. The material recommendation procedure with our method

- | |
|---|
| <ol style="list-style-type: none"> 1. Input the test result before learning \mathbf{x}, 2. Initialize the recommended material vector: $\mathbf{z} = (0, \dots, 0)$, 3. Select a material to be recommended \hat{j} by (6), 4. Update the recommended material vector: $\mathbf{z} = \mathbf{z}^{+\hat{j}}$, 5. Return to step 3 unless an end condition is satisfied. |
|---|

where $P(i)$ represents the probability that question i is asked in the test, $P(i) + \bar{P}(i) = 1$, in which $\bar{P}(i)$ represents the probability that question i is not asked in the test, $P(x_i = -1)$ represents the probability that question i is incorrectly answered before learning, and $P(y_i = 1|x_i = -1, \mathbf{z})$ represents the probability that question i is correctly answered after the learning phase when the question i is incorrectly answered before the learning phase and materials \mathbf{z} are recommended. In (4), $E(\mathbf{z})$ is regarded as the expected number of questions that are incorrectly answered before the learning phase and correctly answered after the learning phase given recommended materials.

When the probabilities that questions are asked are uniform, and questions that are incorrectly answered before learning are known, the expected improvement in the test score can be simplified as follows:

$$E(\mathbf{z}|\mathbf{x}) \propto \sum_{i \in \mathbf{V}} I(x_i = -1) P(y_i = 1|x_i = -1, \mathbf{z}), \quad (5)$$

where $I(A)$ represents an indicator function, i.e. $I(A) = 1$ if A is true, $I(A) = 0$ otherwise. We use (5) as the expected improvement in the test score in the following sections for the simplicity.

Our method sequentially selects a material that maximizes the expected improvement from materials that have not yet been recommended as follows:

$$\hat{j} = \arg \max_{j: z_j=0} E(\mathbf{z}^{+j}|\mathbf{x}), \quad (6)$$

where $\mathbf{z} = (z_j)_{j \in \mathbf{M}}$ represents currently recommended materials, and \mathbf{z}^{+j} represents recommended materials when material j is newly recommended, or $z_{j'}^{+j} = 1$ if $j = j'$ and $z_{j'}^{+j} = z_{j'}$ if $j \neq j'$. Table 1 shows the material recommendation procedure with our method. Examples of end conditions include those where the number of recommended materials, the expected improvement, or the time period of the learning phase exceeds a certain threshold.

2.3 Improvement Model

When recommending materials, our method requires improvement model $P(y_i = 1|x_i = -1, \mathbf{z})$, which is the probability of the improvement of question i given recommended materials \mathbf{z} . The improvement is modeled based on logistic regression [5] as follows:

$$P(y_i = 1|x_i = -1, \mathbf{z}) = \frac{1}{1 + \exp(-(\mu_i + \boldsymbol{\theta}_i^\top \mathbf{z}))}, \quad (7)$$

where μ_i and $\boldsymbol{\theta}_i = (\theta_{ij})_{j \in \mathcal{M}}$ are unknown parameters. Intuitively speaking, μ_i represents the ease with which question i is improved, and θ_{ij} represents the influence of material j on the improvement of question i .

The unknown parameters $\boldsymbol{\Theta} = \{\mu_i, \boldsymbol{\theta}_i\}_{i \in \mathcal{V}}$ can be estimated by maximizing the following log likelihood using the learning log data and test results for a set of users \mathcal{N} :

$$L(\boldsymbol{\Theta}) = \sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{V}} \left(I(x_{ni} = -1 \wedge y_{ni} = 1) \log P(y_{ni} = 1 | x_{ni} = -1, \mathbf{z}_n) + I(x_{ni} = -1 \wedge y_{ni} = -1) \log P(y_{ni} = -1 | x_{ni} = -1, \mathbf{z}_n) \right), \quad (8)$$

where x_{ni} and y_{ni} indicate whether question i is correctly or incorrectly answered by user n before and after the learning phases, respectively, and $P(y_{ni} = -1 | x_{ni} = -1, \mathbf{z}_n)$ represents the probability that question i is incorrectly answered by user n after the learning phase when question i is incorrectly answered before the learning phase and materials \mathbf{z} are recommended. The global optimum solution is guaranteed because the above log likelihood based on logistic regression models is a convex function.

3 Experiments

3.1 Setting

We evaluated our method using fill-in-the-blank questions for English learning, which are employed in many tests designed to evaluate grammatical knowledge, such as TOEIC and TOEFL. In the fill-in-the-blank questions used here, users select appropriate words with the correct grammar for the blank in the sentence from four options.

We implemented a web-based e-learning system for the evaluation. In the experiment, a user takes a test before and after the learning phase (pre-test and post-test) to measure the effect of learning on improving the test score. One question is presented to a user on one web page, and the user answers each question in series. The questions in the pre- and post-tests are the same, and there are 40 questions, $|\mathcal{V}| = 40$.

The materials recommended in the learning phase are fill-in-the-blank questions with solutions and explanations. One question is recommended to each user on one web page for learning, and the solution and explanation are presented on one web page after the user has answered the question. Please note that the users are not supplied with solutions and explanations in the pre- and post-tests. There are 80 learning materials, or questions, $|\mathcal{M}| = 80$, and 40 of these are recommended to each user in the learning phase. The learning materials are different from the questions in the pre- and post-tests. However, about half of the materials are related to test questions, for example they involve questions about the same idioms and grammatical rules.

Table 2. AUC of improvement models based on the Bernoulli distribution and the logistic regression

Bernoulli distribution	Logistic regression
0.556	0.592

3.2 Evaluation of Improvement Models

Our method requires improvement model $P(y_i = 1|x_i = -1, \mathbf{z})$. We constructed and evaluated improvement models using the log data of 52 users with random material recommendations, $|\mathcal{N}| = 52$. We compared improvement models based on the logistic regression in (7) with the Bernoulli distribution. The Bernoulli model assumes that the improvement does not depend on the recommended materials \mathbf{z} . The parameters can be estimated based on the maximum likelihood.

For the evaluation measurement, we used the area under the ROC curve (AUC) of the problem to predict whether or not questions that were incorrectly answered in the pre-test are correctly answered in the post-test. A higher AUC represents better predictive performance. We computed AUC using leave-one-out cross-validation, which means that we used 52 evaluation data sets, in each of which one user's data are used for the evaluation and the data of the other 51 users are used for the training. Table 2 shows the AUC. The AUC of the improvement model based on the logistic regression is higher than that of the Bernoulli model, which implies that the recommended materials are important in terms of predicting the improvement in the score, and we can predict the improvement in the test score with the logistic regression model.

The highest and second highest θ_{ij} were $\theta_{i_1j_1} = 0.388$ and $\theta_{i_2j_2} = 0.208$, respectively, where question i_1 and material j_1 are about the idiom ‘stop by’, and question i_2 and material j_2 are about the idiom ‘across the street’. This result is natural because the recommendation of materials about the same idioms can improve the test score. Even though our method does not use information related to question or material content, it automatically extracts the relationship between questions and materials using the learning log data and test results.

We analyzed the relationship between questions and materials using question i_1 and material j_1 , which are about the idiom ‘stop by’, as an example. In the pre-test, 32 users answered question i_1 incorrectly. The probability of the user answering question i_1 correctly in the post-test when material j_1 was recommended was $\hat{P}(y_{i_1} = 1|x_{i_1} = -1, z_{j_1} = 1) = 15/17$. In contrast, the probability when material j_1 was not recommended was $\hat{P}(y_{i_1} = 1|x_{i_1} = -1, z_{j_1} = 0) = 2/15$. This result indicates that the recommendation of material j_1 is effective in improving the response to question i_1 .

3.3 Evaluation of Recommendation Algorithms

We evaluated the learning efficiency of the proposed recommendation algorithm by comparing it with a random recommendation algorithm. 38 users studied materials recommended at random, and 49 users studied materials recommended

Table 3. Average improvements in the test scores with random recommendation and our method

Random	Our method
4.474	8.125

by our method. Table 3 shows the average improvements in the test scores on a 100-point scale. Our method provided statistically significant increases compared with the random recommendation (one-tailed t-test, $p < 0.04$). In the questionnaires, 86% of users answered that there were helpful materials in the learning phase with our method.

4 Conclusion

We proposed a method for recommending learning materials that maximizes learning efficiency, or the expected improvement in the test score. The experimental results encourage us to believe that our learning material recommendation approach is promising and will become a useful tool for e-learning. Although we modeled the expected improvement in the test score with logistic regression using only the learning log data and test results, we could also use content information about the learning materials and test questions such as difficulties, and user attributes such as levels. We plan a further verification of our method by applying it to other courses of learning.

References

1. Chen, C.M., Hong, C.M., Chang, M.H.: Personalized learning path generation scheme utilizing genetic algorithm for web-based learning. *WSEAS Transactions on Information Science and Applications* 3(1), 88–95 (2006)
2. Li, X., Chang, S.K.: A personalized e-learning system based on user profile constructed using information fusion. In: Proceedings of the 11th International Conference on Distributed Multimedia Systems, Banff, Canada, pp. 109–114 (September 2005)
3. Stern, M., Woolff, B.P.: Curriculum sequencing in a web-based tutor. In: Goettl, B.P., Halff, H.M., Redfield, C.L., Shute, V.J. (eds.) *ITS 1998. LNCS*, vol. 1452, pp. 574–583. Springer, Heidelberg (1998)
4. Tang, T.Y., McCalla, G.: Smart recommendation for evolving e-learning system. In: Proceedings of the 11th International Conference on Artificial Intelligence in Education, Workshop on Technologies for Electronic Documents for Supporting Learning, pp. 699–710 (2003)
5. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, New York (2001)

A Hybrid Kansei Design Expert System Using Artificial Intelligence

Jyun-Sing Chen, Kun-Chieh Wang, and Jung-Chin Liang

Department of technological Product Design, Ling Tung University
1, Lingtung Road, Nantun, 40852, Taichung City, Taiwan, R.O.C
ching_sing@mail.ltu.edu.tw

Abstract. This paper aims to propose a novel approach for high heel design using the integral schemes of Kansei engineering and grey-based artificial intelligence. First, to meet the market's needs, the Kansei engineering scheme is adopted in order to translate the customer's preferences into the product's form elements. Secondly, to speed up and enhance the translation performance, the grey system theory and radial basis function neural network schemes are used. Thirdly, a bi-directional evaluation hybrid Kansei engineering system is constructed via the aforementioned methodology. Finally, a form design expert system is proposed in consideration of designer's usage. To illustrate the functions of the proposed design expert system, an example of the development of new high heels is demonstrated.

Keywords: High heels design, Kansei engineering, design expert system, artificial intelligence, radial basis function neural network, the grey system theory.

1 Introduction

Nowadays consumers are strict in choosing products in terms of their demands and preferences [1]. Obviously, the key factor that influences the success of a new product is how to capture the “voice of the customer.” In order to help designers develop a suitable product form for a given product image, some models, such as design support models [2] and consumer-oriented technologies [3], have been proposed to capture the relationship between the product form and the product image perceived by consumers.

In particular, Kansei engineering [4] has been developed as a comprehensive consumer-oriented technology for new product development. Kansei is a Japanese word which means a consumer's psychological feeling and image regarding a new product. It is defined as “translating technology of a consumer's feeling and image for a product into design elements.” It has been successfully applied in the field of product design [5] to explore the relationship between the feeling of the consumers and the design elements of the product.

The aim of this study is to present a new approach, using high heels as the target object, to establish the relationship between the perception of product image and the product's form elements by applying the Kansei engineering in conjunction with the grey system theory and the radial basis function neural network scheme.

2 Theoretical Basis

2.1 Kansei Engineering

In hybrid Kansei engineering, the customer's requirements can be transferred into feasible alternative combinations of the desired product and vice versa. The operational procedures of hybrid Kansei engineering can be found in [5].

2.2 Radial Basis Function Neural Network (RBFN)

In this study, the artificial intelligence of the RBFN method is used to construct the relationship between product form elements and product images. Detail of the RBFN can be found in [6].

3 Product Form Classification and Decomposition

3.1 Collection of All Kinds of Products

In the first stage, 157 high heel samples were collected from the current market, and after deleting too exaggerated, strange, specific, or similar forms, a total number of 50 different high heels were chosen and made into show cards for further manipulation.

In the second stage, the 50 obtained high heel samples were intended to be separated into 3-12 groups by their degree of similarity. An experiment was done with 50 female subjects, aged from 18 to 40. The collected data were further analyzed using hierarchical cluster analysis (HCA) methodology and the obtained stress results show that the 6-dimensional data with stress smaller than 0.05 (experience suggested) is suitable as the classification basis. Accordingly, a total number of 50 high heel samples are classified into 6 groups. Meanwhile, the representative of each group can be calculated via the K-means data mining scheme.

3.2 Form Decomposition

The form elements of high heels, extracted by morphological analysis on previously obtained 6 groups, are classified into 6 categories with 1~9 types for each category

Structure	Type of element
Head(X1)	none 1 2 3 4 5 6 7
Front(X2)	1 2 3 4 5 6 7 8
Waist(X3)	none 1 2 3 4 5 6 7
Rear(X4)	none 1 2 3 4 5 6 7
Head root(X5)	none 1 2 3 4 5 6
Rear root(X6)	1 2 3 4 5 6 7 8 9

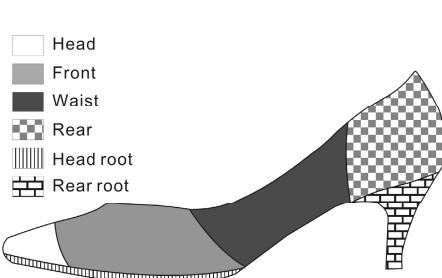


Fig. 1. Morphological analysis

Fig. 2. Definition of construction components

and expressed as X1~X6, as shown in Fig 1. The details of parameters regarding each form element are shown in Fig. 2.

4 Customer's Preference

4.1 Collection of Kansei Words

Using high heels as the target product, a total of 120 low-level Kansei adjective words were collected from magazines, literature, manuals, experts, and experienced users. After deleting those too exaggerated, specific, or similar words, a total group of 63 middle-level Kansei words were built up.

4.2 Second Reduction in Kansei Words

To further reducing the collected middle-level Kansei words, a questionnaire interview was done with 50 female subjects, aged 18-40, and a finally 10 high-level Kansei word pairs were obtained as follows: fashionable-traditional, elegant-crude, luxurious-cheap, formal-informal, light-heavy, cool-warm, young-old, unique-general, wearable-not wearable, and simple-complex.

4.3 Final Reduction in Kansei Words

A questionnaire interview and then a factor analysis were applied to the obtained 10 Kansei words against 6 previously obtained representative samples and the final 3 group-representative image word pairs are obtained as fashionable-traditional (Y1, contribution: 56.75%), unique-general (Y2, contribution: 20.29%), and simple-complex (Y3, contribution: 19.55%), as shown in Table 1.

Table 1. Factor analysis results

Image		Factor1	Factor2	Factor3	Eigenvalue	Contribution	Accumulated Contribution
2.Elegant-Crude		-0.757					
4.Formal-Informal		-1.071					
1.Fashional-Traditional	1.019				5.557	56.75%	55.79%
5.Light-Heavy		0.937					
9.Wearable-NotWearable		0.731					
7.Young-Old		0.854					
6.Cool-Warm		0.842					
8.Unique-General	0.96				2.17	20.29%	76.67%
10.Simple-Complex		0.943			1.962	19.55%	95.42%
3.Luxurious-Cheap		0.872					

5 Relationship

5.1 Evaluation Matrix

To obtain the evaluation index for the three image word pairs of high heels, a 7-point scale (1-7) of the SD method is used, where as an example, 1 is the most simple and 7

is the most complex for the simple to complex image word pair. The resultant Kansei evaluation matrix, obtained from the questionnaire survey results of 16 female subjects, is shown in Table 2.

Table 2. Kansei evaluation matrix

Sample	Fsahional-Traditional(Y1)	Unique-General(Y2)	Simple-Complex(Y3)
1	2.15	5.51	4.15
2	2.85	4.84	4.6
3	3.07	4.37	4.77
4	2.5	5.1	4.51
5	3.11	4.96	3.97
6	2.37	5.88	4.52
7	2.98	5.82	4.26
8	3.31	4.08	4.5
9	2.41	3.98	3
10	2.79	4.55	4.15
11	2.22	4.3	3.9
12	3.18	4.81	4.43
13	2.27	4.07	4.1
14	2.54	4.25	4.23
15	2.61	4.36	4.67
16	3.07	4.12	4.15

5.2 Influence of Form Elements on Kansei Adjective Words

According to the GM(1,N) model of the grey system theory[7],the influence weight-ing($b_1 \sim b_6$) of the form elements(X1~X6) on Kansei words(Y1~Y3) can be obtained as following.

(1)For Y1 (fashionable-traditional), $b_1=26.7$, $b_2=41.6$, $b_3=2.9$, $b_4=0.8$, $b_5=11.8$, $b_6=16.2$.By taking the norm of each value, the resultant rank of the coefficient sequences can be obtained: $X_2 > X_1 > X_6 > X_5 > X_3 > X_4$. According to the relative norm levels of $b_2 \sim b_2$, only the form elements with high rank need to be considered to map the form elements to Kansei words in order to quickly as well as accurately establish the mapping relationship.

(2) For Y2 (unique-general), $b_1=15.5$, $b_2=38.7$, $b_3=33.2$, $b_4=0.8$, $b_5=9.1$, $b_6=2.6$.The coefficient sequence is: $X_2 > X_3 > X_1 > X_6 > X_5 > X_4$.

(3) For Y3 (simple-complex), $b_1=31.2$, $b_2=24.7$, $b_3=26.3$, $b_4=1.9$, $b_5=12.2$, $b_6=3.7$,The coefficient sequences can be obtained: $X_1 > X_3 > X_2 > X_5 > X_6 > X_4$.

5.3 Mathematic Modeling

In modeling, we choose five high-ranking form elements : X1,X2,X3,X4,X5,X6 (common to all three Kansei adjective words) and three Kansei adjective words as input and output variables of RBFN and vice versa. The spread constant is chosen as 0.09 and the maximum number of neurons is set as 100. The sum of the squared error is assigned 0.01 as the convergence criterion. A total of 10 cases are used as training and 6 cases for checking. It takes 15 and 10 epochs to reach convergence in training and checking respectively.

6 Performance Evaluation

To identify our developed hybrid Kansei system, a questionnaire survey of 50 females, aged from 20-30 and experienced in wearing high heels, was performed. In this survey, three 2D drawing cards (15cm*10cm) of high heels, which are proper combinations of different form elements, are shown to the test subjects. Using a seven-scale SD method as the evaluation basis, the gathered survey data from all test subjects was analyzed via SPSS. Eventually, the comparison between tested and predicted results (shown in Fig. 3) indicates that the predicted errors are all within 0.44. This confirms the reliability of our developed Kansei system.

7 Design Expert System

The VB editor of CATIA released by Dassault Co. Ltd. is used as a human-machine interface tool to construct a design expert system. Through this system, any designer can easily translate the customer's needs into product form elements and vice versa. Further, one can obtain a basic outline of the product's form according to certain kinds of customer's preference, and then add creativity into this original outline to accomplish a final innovative product form. Following is the illustration of our developed design expert system.

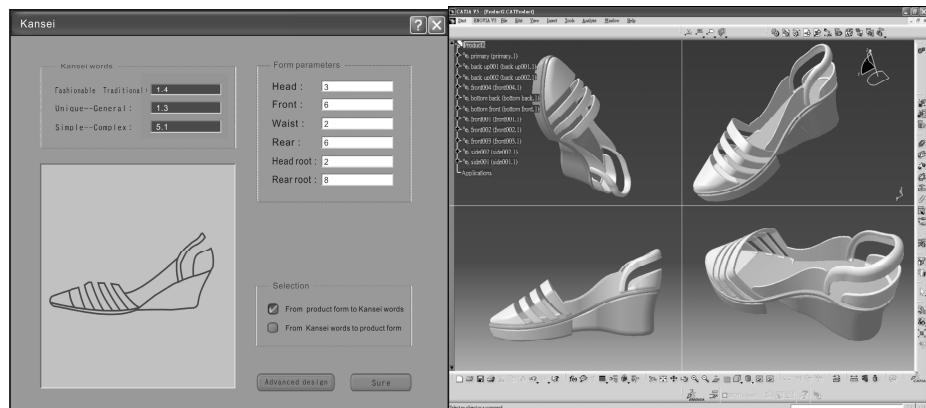


Fig. 3. Hybrid Kansei evaluation page

Fig. 4. Advanced design page

The home page of our developed expert system includes two major evaluation functions: (1) Kansei to Form Elements and (2) Form elements to Kansei. When we pick the 'Rear' form parameter (shown in the top right section of Fig. 3), then give proper points for each of the three Kansei words, as soon as we push the "Sure" button, the correspondent Kansei words and high heels drawing appear simultaneously. If one wants to further apply any modification or add creative thinking to this high heel prototype, just push the "Advanced Design" button, a 3D drawing of this prototype shows up (shown in Fig. 4), waiting for further manipulation.

8 Conclusions

- (1) A bi-directional Kansei evaluation system has been constructed in which predictions from Kansei to product form and from product form to Kansei can be easily and accurately accomplished.
- (2) Based on this hybrid Kansei engineering system, designers can immediately catch the market's trends as well as the customers' voices so as to perform a good product form design.
- (3) This hybrid Kansei engineering system uses the artificial intelligence and the grey system theory as its mapping schemes, and has verified good prediction ability.
- (4) A design expert system is also developed using the above hybrid Kansei engineering system as its kernel.
- (5) This design expert system provides a user-friendly interface. Any user can easily manipulate this system to develop a suitable product form meeting customers' requirement and meanwhile combine this with creative thinking.

References

1. Jiao, J., Zhang, Y., Helander, M.: A Kansei mining system for affective design. *Expert Systems with Applications* 30, 658–673 (2006)
2. Chung, M.C., Chang, C.C., Hsu, S.H.: Perceptual elements underlying user preferences toward product form of mobile phones. *International Journal of Industrial Ergonomics* 27, 247–258 (2001)
3. Hsu, C.H., Jiang, B.C., Lee, E.S.: Fuzzy neural network modeling for product development. *Journal of Mathematical and Computer Modeling* 29, 71–81 (1999)
4. Nagamachi, M.: Kansei engineering: a new ergonomics consumer-oriented technology for product development. *Journal of Industrial Ergonomics* 15(1), 3–11 (1995)
5. Lai, H.H., Lin, Y.C., Yeh, C.H.: Form design of product image using grey relational analysis and neural network models. *Computers & Operations Research* 32, 2689–2711 (2005)
6. Ray, K.S., Ghoshal, J.: Neuro Fuzzy Approach to pattern Recognition. *Neural Network* 10(1), 161–182 (1997)
7. Deng, J.: Introduction to Grey System Theory. *The Journal of Grey System* 1, 1–24 (1989)

Solving the Contamination Minimization Problem on Networks for the Linear Threshold Model

Masahiro Kimura¹, Kazumi Saito², and Hiroshi Motoda³

¹ Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp

² School of Administration and Informatics, University of Shizuoka
Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp

³ Institute of Scientific and Industrial Research, Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We address the problem of minimizing the spread of undesirable things, such as computer viruses and malicious rumors, by blocking a limited number of links in a network. This optimization problem called the contamination minimization problem is, not only yet another approach to the problem of preventing the spread of contamination by removing nodes in a network, but also a problem that is converse to the influence maximization problem of finding the most influential nodes in a social network for information diffusion. We adapted the method which we developed for the independent cascade model, known for a model for the spread of epidemic disease, to the contamination minimization problem under the linear threshold model, a model known for the propagation of innovation which is considerably different in nature. Using large real networks, we demonstrate experimentally that the proposed method significantly outperforms conventional link-removal methods.

1 Introduction

Networks can mediate diffusion of various things such as innovation and topics. However, undesirable things can also spread through networks. For example, computer viruses can spread through computer networks and email networks, and malicious rumors can spread through social networks among individuals. Thus, developing effective strategies for preventing the spread of undesirable things through a network is an important research issue. Previous work studied strategies for reducing the spread size by removing nodes from a network. It has been shown in particular that the strategy of removing nodes in decreasing order of out-degree can often be effective [1,2,3]. Here notice that removal of nodes by necessity involves removal of links. Namely, the task of removing links is more fundamental than that of removing nodes. Therefore, preventing the spread of contamination by blocking links from the underlying network is an important problem.

In contrast, finding a limited number of influential nodes that are effective for the spread of information through a social network is also an important research issue in

terms of sociology and “viral marketing” [4,5,6]. Widely-used fundamental probabilistic models of information diffusion through networks are the *independent cascade (IC) model* and the *linear threshold (LT) model* [7,6]. Researchers have recently studied a combinatorial optimization problem called the *influence maximization problem* on a network under these models [7,8]. Here, the influence maximization problem is the problem of extracting a set of k nodes to target for initial activation such that it yields the largest expected spread of information, where k is a given positive integer. Note also that the IC and LT models are fundamental models of contamination diffusion process on networks [6].

The problem we address in this paper is a problem that is converse to the influence maximization problem. The problem is to minimize the spread of contamination by blocking a limited number of links in a network. More specifically, when some undesirable thing starts with any node and diffuses through the network, we consider finding a set of k links such that the resulting network by blocking those links minimizes the expected contamination area of the undesirable thing, where k is a given positive integer. This combinatorial optimization problem is referred to as the *contamination minimization problem* [9]. For the contamination minimization problem under the IC model, Kimura, Saito and Motoda [9] presented a method for efficiently finding a good approximate solution on the basis of a naturally greedy strategy.

In this paper, we propose a method for efficiently finding a good approximate solution to the contamination minimization problem under the LT model by adapting the greedy method developed for the problem under the IC model. Note here that the IC and LT models considerably differ in quality. First, the LT model is originally a model for the propagation of innovation through the network, while the IC model can be identified with the *SIR model* [10] for the spread of epidemic disease in the network. Moreover, the LT model is viewed as a probabilistic model defined on some continuous space, while the IC model is viewed as that on some finite set (i.e., a discrete space) [7,8]. Therefore, the effectiveness of the greedy method for the problem under the LT model is not self-evident. To compare methods of solving the problem for various networks in performance, we newly introduce the *contamination reduction rate* as a performance measure. Using large real social networks, we experimentally demonstrate that the proposed method significantly outperforms link-removal heuristics that rely on the well-studied notions of betweenness and out-degree in the field of complex network theory.

2 Problem Formulation

In this paper, we address the problem of minimizing the spread of some undesirable thing in a network represented by a directed graph $G = (V, E)$. Here, V and E ($\subset V \times V$) are the sets of all the nodes and links in the network, respectively. We assume the LT model to be a mathematical model for the diffusion process of this undesirable thing in the network, and investigate the contamination minimization problem on G . We call nodes *active* if they have been contaminated by the undesirable thing.

2.1 Linear Threshold Model

We define the *linear threshold (LT) model* on graph G according to [7].

In this model, for any node $v \in V$, we specify, in advance, a *weight* $\omega_{u,v}$ (> 0) from its parent node u such that $\sum_{u \in \Gamma(v)} \omega_{u,v} \leq 1$, where $\Gamma(v)$ is the set of all the parent nodes of v , $\Gamma(v) = \{u \in V; (u, v) \in E\}$. The diffusion process from a given initial set of active nodes proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0, 1]$. At time-step t , an inactive node v is influenced by each of its active parent nodes, u according to weight $\omega_{u,v}$. If the total weight from active parent nodes of v is at least threshold θ_v , that is, $\sum_{u \in \Gamma_t(v)} \omega_{u,v} \geq \theta_v$, then v will become active at time-step $t + 1$. Here, $\Gamma_t(v)$ stands for the set of all the parent nodes of v that are active at time-step t . The process terminates if no more activations are possible.

Note that the threshold θ_v models the tendency of node v to adopt the information when its parent nodes do. Note also that the LT model is a probabilistic model associated with the uniform distribution on $[0, 1]^{|V|}$. Thus, the LT model is viewed as a probabilistic model on the continuous space $[0, 1]^{|V|}$. Here, $|A|$ stands for the number of elements of a set A .

For an initial active node v , let $\sigma(v; G)$ denote the expected number of active nodes at the end of the random process of the LT model on G . We call $\sigma(v; G)$ the *influence degree* of node v in graph G .

2.2 Contamination Minimization Problem

Now, we give a mathematical definition of the contamination minimization problem on graph $G = (V, E)$.

First, we define the *contamination degree* $c(G)$ of graph G as the average of influence degrees of all the nodes in G , that is,

$$c(G) = \frac{1}{|V|} \sum_{v \in V} \sigma(v; G). \quad (1)$$

For any link $e \in E$, let $G(e)$ denote the graph $(V, E \setminus \{e\})$. We refer to $G(e)$ as the graph constructed by *blocking* e in G . Similarly, for any $D \subset E$, let $G(D)$ denote the graph $(V, E \setminus D)$. We refer to $G(D)$ as the graph constructed by *blocking* D in G . We define the *contamination minimization problem* on graph G as follows: Given a positive integer k with $k < |E|$, find a subset D^* of E with $|D^*| = k$ such that $c(G(D^*)) \leq c(G(D))$ for any $D \subset E$ with $|D| = k$.

For a large network, any straightforward method for exactly solving the contamination minimization problem suffers from combinatorial explosion. Therefore, we consider approximately solving the problem.

3 Proposed Method

We propose a method for efficiently finding a good approximate solution to the contamination minimization problem on graph $G = (V, E)$. We consider adapting the method which we developed for the IC model to the contamination minimization problem under the LT model which is considerably different in nature. Let k be the number of links to be blocked in this problem.

3.1 Greedy Algorithm

We approximately solve the contamination minimization problem on $G = (V, E)$ by the following greedy algorithm:

1. Set $D_0 \leftarrow \emptyset$.
2. Set $E_0 \leftarrow E$.
3. Set $G_0 \leftarrow G$.
4. **for** $i = 0$ to $k - 1$ **do**
5. Choose a link $e_* \in E_i$ minimizing $c(G_i(e))$, ($e \in E_i$).
6. Set $D_{i+1} \leftarrow D_i \cup \{e_*\}$.
7. Set $E_{i+1} \leftarrow E_i \setminus \{e_*\}$.
8. Set $G_{i+1} \leftarrow (V, E_{i+1})$.
9. **end for**

Here, D_k is the set of links blocked, and represents the approximate solution obtained by this algorithm. G_k is the graph constructed by blocking D_k in graph G , that is, $G_k = G(D_k)$.

To implement this greedy algorithm, we need a method for calculating $\{c(G_i(e)); e \in E_i\}$ in Step 5 of the algorithm. However, the LT model is a stochastic process model, and it is an open question to exactly calculate influence degrees by an efficient method [7]. Therefore, we develop a method for estimating $\{c(G_i(e)); e \in E_i\}$.

Kimura, Saito, and Nakano [8] presented the bond percolation method that efficiently estimates the influence degrees $\{\sigma(v; \tilde{G}); v \in \tilde{V}\}$ for any directed graph $\tilde{G} = (\tilde{V}, \tilde{E})$. Thus, we can estimate $c(G_i(e))$ for each $e \in E_i$ by straightforwardly applying the bond percolation method. However, $|E_i|$ becomes very large for a large network unless i is very large. Therefore, we propose a method that can estimate $\{c(G_i(e)); e \in E_i\}$ in a more efficient manner on the basis of the bond percolation method.

3.2 Estimation Based on Bond Percolation Method

It is known that the LT model is equivalent to the following bond percolation process [7]: For any $v \in V$, we pick at most one of the incoming links to v by selecting link (u, v) with probability $\omega_{u,v}$ and selecting no link with probability $1 - \sum_{u \in \Gamma(v)} \omega_{u,v}$. Then, we declare the picked links to be “occupied” and the other links to be “unoccupied”. Note here that the equivalent bond percolation process for the LT model is considerably different from that of IC model.

In the bond percolation method [8], we efficiently estimate the influence degrees $\{\sigma(v; G_i); v \in V\}$ in the following way. Let M be a sufficiently large positive integer. We perform the bond percolation process M times, and sample a set of M graphs, $\{G_i^m = (V, E_i^m); m = 1, \dots, M\}$, constructed by the occupied links. Then, using the strongly connected decomposition of each G_i^m , we efficiently estimate the influence degrees $\{\sigma(v; G_i); v \in V\}$ as

$$\sigma(v; G_i) = \frac{1}{M} \sum_{m=1}^M |\mathcal{F}(v; G_i^m)|, \quad (v \in V), \quad (2)$$

(see [8] in detail). Here, $\mathcal{F}(v; G_i^m)$ denotes the set of all the nodes that are *reachable* from node v in the graph G_i^m . We say that node u is reachable from node v if there is a path from u to v along the links in the graph.

We are now in a position to give a method for efficiently estimating $\{c(G_i(e)); e \in E_i\}$ in Step 5 of the greedy algorithm. For the LT model, the weights $\{\omega_{u,v}\}$ must be specified in advance. We uniformly set the weights as follows: For any node $v \in V$, the weight $\omega_{u,v}$ from a parent node $u \in \Gamma(v)$ is given by

$$\omega_{u,v} = \frac{1}{|\Gamma(v)| + 1}.$$

Here note that $\sum_{u \in \Gamma(v)} \omega_{u,v} < 1$ for any $v \in V$, that is, there exists a chance such that node v cannot become active even if all the parent nodes of v are active. Then, on the basis of Equations (1) and (2), and the independence of the bond percolation process, we estimate $\{c(G_i(e)); e \in E_i\}$ by

$$c(G_i(e)) = \frac{1}{|\mathcal{M}_i(e)|} \sum_{m \in \mathcal{M}_i(e)} \frac{1}{|V|} \sum_{v \in V} \mathcal{F}(v; G_i^m), \quad (e \in E_i)$$

without applying the bond percolation method for every $e \in E_i$, where $\mathcal{M}_i(e) = \{m \in \{1, \dots, M\}; e \notin E_i^m\}$. Namely, the proposed method can achieve a great deal of reduction in computational cost compared with the conventional bond percolation method.

4 Experimental Evaluation

4.1 Experimental Settings

In our experiments, we employed two sets of large real networks used in [9], the blog and Wikipedia networks, which exhibit many of the key features of social networks. These are bidirectional networks. The blog network had 12,047 nodes and 79,920 directed links, and the Wikipedia network had 9,481 nodes and 245,044 directed links.

For the proposed method, we need to specify the number M of performing the bond percolation process. In the experiments, we used $M = 10,000$ according to [8].

4.2 Comparison Methods

We compared the proposed method with two heuristics based on the well-studied notions of betweenness and out-degree in the field of complex network theory.

The *betweenness score* $b_{\tilde{G}}(e)$ of a link e in a directed graph $\tilde{G} = (\tilde{V}, \tilde{E})$ is defined as follows: $b_{\tilde{G}}(e) = \sum_{u,v \in \tilde{V}} n_{\tilde{G}}(e; u, v) / N_{\tilde{G}}(u, v)$, where $N_{\tilde{G}}(u, v)$ denotes the number of the shortest paths from node u to node v in \tilde{G} , and $n_{\tilde{G}}(e; u, v)$ denotes the number of those shortest paths that pass e . Here, we set $n_{\tilde{G}}(e; u, v) / N_{\tilde{G}}(u, v) = 0$ if $N_{\tilde{G}}(u, v) = 0$. Newman and Girvan [11] successfully extracted community structure in a network using the following link-removal algorithm based on betweenness:

1. Calculate betweenness scores for all links in the network.
2. Find the link with the highest score and remove it from the network.

3. Recalculate betweenness scores for all remaining links.
4. Repeat from Step 2.

In particular, the notion of betweenness can be interpreted in terms of signals traveling through a network. If signals travel from source nodes to destination nodes along the shortest paths in a network, and all nodes send signals at the same constant rate to all others, then the betweenness score of a link is a measure of the rate at which signals pass along the link. Thus, we naively expect that blocking the links with the highest betweenness score can be effective for preventing the spread of contamination in the network. Therefore, we apply the method of Newman and Girvan [11] to the contamination minimization problem. We refer to this method as the *betweenness method*.

On the other hand, previous work has shown that simply removing nodes in order of decreasing *out-degrees* works well for preventing the spread of contamination in most real networks [1,2,3]. Here, the out-degree of a node v means the number of outgoing links from the node v . Therefore, as a comparison method, we consider the straightforward application of this node removal method. Namely, we employ the method of choosing nodes in decreasing order of out-degree and blocking simultaneously all the links attached to the chosen nodes. We refer to this method as the *out-degree method*. Note that the out-degree method can not be applied for all values of k to the contamination minimization problem of blocking k links.

4.3 Experimental Results

We evaluated the performance of the proposed method and compared it with that of the betweenness and out-degree methods. Clearly, the performance of a method for solving the contamination minimization problem can be evaluated in terms of the *contamination reduction rate CRR* that is defined as follows:

$$CRR = 100 \frac{c(G) - c(G')}{c(G)},$$

where G' stands for a solution graph constructed by blocking a specified number of links from the original graph G . We estimated the value of c by the bond percolation method with $M = 10,000$ (see Equations (1) and (2)), and computed the value of CRR .

Figures 1 and 2 show the contamination reduction rate CRR of the resulting network as a function of the *fraction of links blocked*, FLB , for the blog and Wikipedia networks, respectively. Here, the circles, triangles and diamonds indicate the results for the proposed, betweenness and out-degree methods, respectively. In the right figures of Figures 1 and 2, the dashed line indicates the contamination reduction rate of the network obtained by the proposed method when the number of links blocked, k , is 500. Here note that $k = 500$ means $FLB = 0.63\%$ and $FLB = 0.20\%$ in the blog and Wikipedia networks, respectively. We see that the proposed method outperformed the betweenness and out-degree methods for both the blog and the Wikipedia networks.

These results imply that the proposed method works effectively as expected, and significantly outperforms the conventional link-removal heuristics, that is, the betweenness and out-degree methods. This shows that a significantly better link-blocking strategy for reducing the spread size of contamination can be obtained by explicitly incorporating

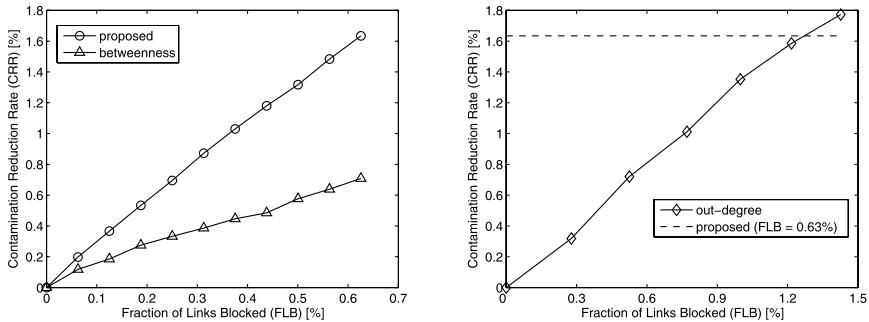


Fig. 1. Performance comparison of the proposed method with the betweenness and out-degree methods in the blog network

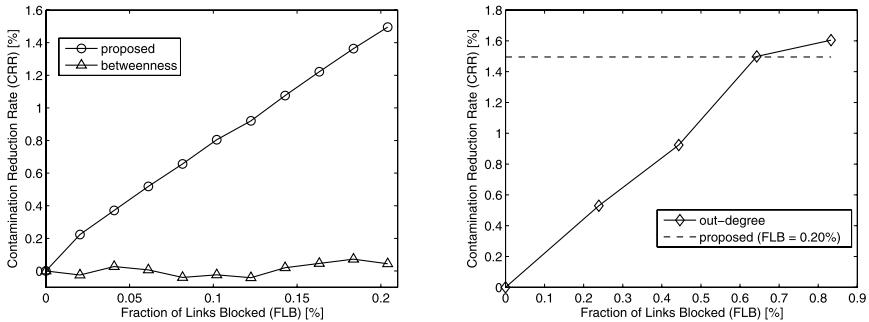


Fig. 2. Performance comparison of the proposed method with the betweenness and out-degree methods in the Wikipedia network

the diffusion dynamics of contamination in a network, rather than relying solely on structural properties of the graph.

In the task of removing nodes from a network, the out-degree heuristic has been effective since many links can be blocked at the same time by removing nodes with high out-degrees. However, we find that in the task of blocking a limited number of links, the strategy of blocking all the links attached to nodes with high out-degrees is not necessarily effective.

5 Conclusion

In an attempt to minimize the spread of undesirable things, such as computer viruses and malicious rumors, by blocking a limited number of links in a network, we have investigated the contamination minimization problem for the LT model that is a fundamental diffusion model on a network. This minimization problem is, not only yet another approach to the problem of preventing the spread of contamination by removing nodes in a network, but also a problem that is converse to the influence maximization

problem of finding the most influential nodes in a social network for information diffusion. We have adapted the method which we developed for the IC model, known for a model for the spread of epidemic disease, to the contamination minimization problem under the LT model, a model known for the propagation of innovation which is considerably different in nature. Using large-scale blog and Wikipedia networks, we have experimentally demonstrated that the proposed method effectively works, and also significantly outperforms the conventional link-removal heuristics based on the betweenness and out-degree.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027, and Grant-in-Aid for Scientific Research (C) (No. 20500147) from Japan Society for the Promotion of Science.

References

1. Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. *Nature* 406, 378–382 (2000)
2. Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J.: Graph structure in the web. In: Proceedings of the 9th International World Wide Web Conference, pp. 309–320 (2000)
3. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. *Physical Review E* 66, 035101 (2002)
4. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 57–66 (2001)
5. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 61–70 (2002)
6. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogsphere. In: Proceedings of the 13th International World Wide Web Conference, pp. 107–117 (2004)
7. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)
8. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence, pp. 1371–1376 (2007)
9. Kimura, M., Saito, K., Motoda, H.: Minimizing the spread of contamination by blocking links in a network. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence, pp. 1175–1180 (2008)
10. Newman, M.E.J.: The structure and function of complex networks. *SIAM Review* 45, 167–256 (2003)
11. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Physical Review E* 69, 026113 (2004)

A Data-Driven Approach for Finding the Threshold Relevant to the Temporal Data Context of an Alarm of Interest

Savo Kordic¹, Peng Lam¹, Jitian Xiao¹, and Huaizhong Li²

¹ The School of Computer and Information Science, Edith Cowan University, Perth 6050, Western Australia

{s.kordic,c.lam,j.xiao}@ecu.edu.au

² The School of Computer Science and Engineering, Wenzhou University Town, Zhejiang 325035, China

hli@wzu.edu.cn

Abstract. A typical chemical alarm database is characterized by a large search space with skewed frequency distribution. Thus in practice, discovery of alarm patterns and interesting associations from such data can be exceptionally difficult and costly. To overcome this problem we propose a data-driven approach to optimally derive the pruning thresholds which are relevant to the temporal data context of the particular tag of interest.

Keywords: Chemical plants, Data mining and Alarm database.

1 Introduction and Related Work

Poor alarm performance ultimately leads to plant shutdowns and accidents. An example is the 2005 explosion and fire at the BP Texas City Refinery [1], which left 15 people dead and hundreds more injured. BP North America paid a record fine of \$50 million and another \$1 billion for the inspection and refurbishment of all main process units in the refinery [2]. The cause of the explosion was finally traced to a redundant high level alarm which the operators had not detected [3].

Chemical plants present a real challenge for applying association rule [4] mining to data, as hundreds of distinct alarm tags may generate a large number of patterns and rules. Since possible associations tend to grow exponentially with the number of items, association rule mining algorithms usually generate too many rules. For this reason, Klemettinen et al. [5] proposed rule templates to filter out rules that may be considered uninteresting by the end user. Even though many interesting rules may be selected or rejected with the help of templates, still the number of association rules usually remains very high. The idea of Lent et al. [6] was to use clustering to reduce the large number of association rules. They merged the rules with ‘adjacent’ attribute values to form more general rules. Kryszkiewicz [7] described an approach for discovering representative association rules, in which redundancy reduction is achieved by using a concise set called closed itemsets.

In addition to support and confidence, many other objective statistical measures have been used in data mining applications. For example, Xiong et al. [8] described an approach for mining association patterns in datasets with skewed frequency distributions. Since both maximal and closed itemsets are incapable of pruning itemsets with drastically different support levels they proposed an objective measure called h-confidence to mine patterns with strong affinity.

In this paper we describe a time and cost effective approach to discover primary and consequential alarms, which can be used to support alarm rationalization by identifying redundant alarms and system bad actors. The main contribution of this paper is a method which allows users to find the threshold value most relevant to the tags of interest, while also dramatically reducing the total number of discovered patterns.

The rest of this paper is organized as follows: Section 2 introduces the problem with necessary notations and describes the proposed technique for finding groups of correlated alarm tags. The Vinyl Acetate process, experiments and relevant results are presented in Section 3 and lastly, the conclusion is found in Section 4.

2 Proposed Approach

2.1 Preliminaries

Note that we follow the basic concepts and definitions introduced by Manilla et al. [9] in defining event sequences. Given a class of event types T , an *alarm* is a pair of terms (a, t) where $a \in T$ and t is the *occurrence time* represented as an integer. An *alarm sequence* S is an ordered collection of alarms defined as $S = \{(a_1, t_1), (a_2, t_2), \dots, (a_n, t_n)\}$, for all $i=1,2,\dots,n$, $a_i \in T$, and $t_{i+1} \geq t_i$ for all $i=1,\dots,n-1$.

Definition 2.1. An *activation-return* (A-R) window W_{A-R} is a subsequence of an entire event sequence S with respect to an event interval $W_{A-R} = (S, t_{act}, t_{ret})$. It consists of the alarm pairs (a, t) from sequence S , where $a \in T$ and $t_{ret} > t \geq t_{act}$.

Definition 2.2. A verifying *return-activation/width* (R-A/w) window $W_{R-A/w}$ is a subsequence of an entire event sequence S with respect to an event interval $W_{R-A} = (S, t_{reb}, t_{act})$ or $W_{R-w} = (S, t_{reb}, t_w)$. It consists of the alarm pairs (a, t) from sequence S , where $a \in T$ and $t_{act} > t \geq t_{reb}$ or $t_w > t \geq t_{reb}$.

Definition 2.3. An *intersection event-set* Φ is a partially ordered set of alarm types $a_1 \leq a_2 \leq \dots$ containing the intersection between *activation* events in the activation-return (A-R) window and *return* events in the verifying return-activation (R-A) or return-time (R-w) window.

Definition 2.4. Let C be the collection of *intersection event-sets* with respect to the tag of interest a_1 . Let E be a set of the distinct alarm tags a_1, a_2, \dots, a_n such that $E \subseteq T$. An *intersection event-set* Φ is said to contain E if and only if $E \subseteq \Phi$, and $\Phi \in C$. We define the frequency $freq(E)$ as the number of *intersection event-sets* which contain all the members in E . For a given frequent event set $E = a_1, a_2, \dots, a_m$, $m \geq 2$, the consequent Y will be the event set $E - a_1$. The confidence $conf(a_1 \Rightarrow Y)$ is calculated as $freq(a_1 \cup Y) / freq(a_1)$.

2.2 Algorithm for Finding Thresholds

Typical chemical process alarm data is skewed owing to the presence of nuisance (i.e. repeated false) alarms. If low support is specified then not only could the total number of frequent patterns increase dramatically but also there could be many spurious relationships between different support level alarms. Due to the very high dimensionality of the data and the exponential number of possible combinations, even with a relatively high support the resulting set of rules could be too numerous to be understood by the user. For these reasons, our approach is to set sufficient minimum thresholds for each alarm tag independently.

We considered the following problem. Given a sequence S , an integer n representing the total number of activations, a return window width w , a threshold $minFreq$ for the frequency, and a minimum slope value χ , find *the optimum threshold value for tag a*. The overall structure of the algorithm is outlined below.

```
ALGORITHM SelectThreshold
FOR n = 1 to n from S DO
    A  $\leftarrow$  activation segment ( $W_{A-R}$  (n))
    R  $\leftarrow$  return segment ( $W_{R-A/R-w}$  (n), w)
    C  $\leftarrow$  intersection ( $A \cap R$ )
    L  $\leftarrow$  singletons frequency (C)
END FOR
WHILE  $L_i \neq \emptyset$  AND  $L_i \geq minFreq$ 
    Listi = ( $L_i / n$ ) * 100 //compute frequency
END WHILE
WHILE Listk  $\neq \emptyset$  AND NOT Found
    slope  $\leftarrow$  Listk-1 - Listk
    IF slope <  $\chi$  THEN
        Found  $\leftarrow$  TRUE
        ThresholdValue = Listk-1
    ELSE
        k  $\leftarrow$  k + 1
    END IF
END WHILE
Output ThresholdValue
```

During the shift of each interval window a set of ordered unique events is automatically recognized and extracted from the corresponding (A-R), (R-A) or (R-w) windows. Activation events in the (A-R) windows are filtered if their return events do not appear in the associated verifying (R-A) or (R-w) window. We calculated the threshold level for a particular tag with respect to minor interval changes in the curve's slope. The number of times the consequential alarm occurred within the windows of the primary tag and the rate of change in slope interval is shown to provide the user with guidance in selecting the threshold value. We obtain a rate of change by subtracting the scores of two adjacent alarm tags. The greater the divergence in frequency, then the greater the curve's slope value will be, and therefore, the less representative the confidence threshold will be. A higher slope value indicates a steeper decline; zero a straight line. In general, an extreme skewness to the right indicates a nuisance alarm. Once the change between consecutive tags on the x-axis is less than

the user specified *minimum slope value* χ , we look for the corresponding confidence value on the y-axis.

3 Experimentation, Results and Conclusions

The proposed approach was evaluated initially using simulated data produced from a “Matlab model of the Vinyl Acetate chemical process”. Vinyl data consists of records of alarm logs which characterize the actual state of the plant at particular points in time, representing the status of 27 alarm monitors (see Figure 1).

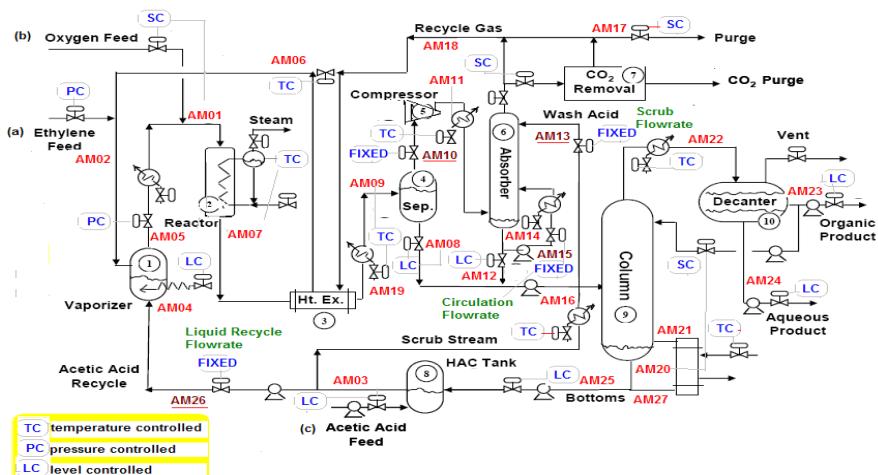


Fig. 1. Vinyl Acetate process flowsheet showing location of the simulated monitors (modified from [10])

It is assumed that a simulated alarm is activated if the following condition is satisfied:

$$Abs\left(\frac{(D_m - N_m)}{N_m}\right) \geq S_{am}$$

where D_m is the disturbed output magnitude, N_m is the normal output magnitude and S_{am} is the sensitivity of the simulated alarm monitor. Note that the simulated alarm will return to normal if the above condition is not satisfied. The signal detection sensitivity for all alarm monitors is set to be equal to 0.0005.

To evaluate the proposed method two simulated data sets were generated with varying complexity and fault durations. *Simulation data 1* - disturbances of one type, loss of oxygen (%O₂) in the Reactor Inlet (associated with alarm monitor **AM01**), were injected into the Vinyl Acetate process to induce a response in measurement outputs. *Simulation data 2* – a different type of disturbance was introduced, namely the loss of the fresh acetic acid (HAc) feed stream (associated with alarm monitor **AM03**), with various durations.

Firstly, we want to demonstrate the effect of the χ slope value on alarm pattern discovery, thus we set the minimum frequency very low at 1 occurrence. Figure 2 illustrates the effect of the χ slope value on *simulated dataset 1*. In these experiments we compare the quantities of discovered patterns without and then with the use of a verifying group and filtering. Figure 2 shows that the return-point filtering accompanying the χ slope selection leads to a dramatic reduction in the number of patterns. The number of patterns ranged from 4202659 for the simulated dataset 1 with *no filtering* and $\chi = 20$, to 179 for the simulated dataset 1 *with filtering* and $\chi = 20$.

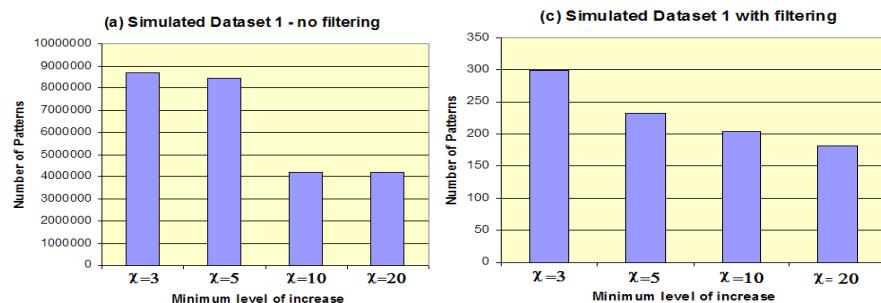


Fig. 2. The effect of the χ slope value on *simulated dataset 1* without and with return-point filtering

Finally, we examined the validity or “trustworthiness” of the discovered patterns. For *simulated dataset 2*, we set minimum frequency a bit higher - equal to 6 occurrences - to find more statistically significant (frequent) rules. Table 1 shows rules for the first 6 alarm tags. Note that we only present selected results, due to an obvious lack of space.

Table 1. Selected comparative results for *simulated dataset 2* with respect to minimum frequency = 6 occurrences, return-point filtering and verifying window width $w = 900$ seconds, and χ level = 10% and 20%

TAG	Rules Dataset 2	freq	χ	conf
1 %O2	1=>() below sup	-	-	-
2 Press	2=>() below sup	-	-	-
3 HAc-L (fault)	3=>16	10	20%	100%
...
6 Pre-T (repeating alarm)	6=>9 18	7	20%	39%
...

In terms of checking whether the group of associated alarms is correct, the results set associated with fault TAG 3 (**AM03, HAc-L**) can be checked against the Vinyl Acetate process flowsheet in Figure 1. A careful examination of the process reveals that as AM13 is *fixed*, thus the temperature change monitored by AM14 is not significantly affected by the temperature change in AM16. Furthermore, since AM26 is

fixed the change to level monitor AM04 should be minimal and will require a longer period for any changes to be registered. Therefore, the real process will have the following significant alarm correlation when alarm TAG 3 is the cause alarm: **TAG 3, TAG 16**.

4 Conclusions

In this paper we have presented our strategy for analyzing large alarm databases, which is required in application domains such as for chemical plant data. A data-driven approach for finding the threshold relevant to the temporal data context of an alarm of interest is used to analyse the relationships between alarm tags. Our preliminary experiments with simulated data showed the usefulness of the proposed approach.

References

1. OSHA, OSHA Fines BP Products North America More Than \$21 Million Following Texas City Explosion (accessed: 4 April, 2008),
<http://www.osha.gov/pls/oshaweb>
2. BP America Inc., BP America announces resolution of Texas City, Alaska, propane trading, law enforcement investigations (accessed: 12 April, 2008), <http://www.bp.com>
3. U.S. Chemical Safety and Hazard Investigation Board, Investigation Report: Refinery Explosion and Fire, Report No. 2005-04-I-TX (March 2007)
4. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD Conference on Management of Data (SIGMPD 1993), pp. 207–216. ACM, New York (1993)
5. Klementtinen, M., Mannila, M.: Finding Interesting rules from Large Sets of Discovered Association Rules. In: Third International Conf. on Information and Knowledge Management, pp. 401–407. ACM, New York (1994)
6. Lent, B., Swami, A., Widom, J.: Clustering association rules. In: 13th International Conference on Data Engineering (ICDE 1997), pp. 220–231. IEEE Press, Washington (1997)
7. Kryszkiewicz, M.: Closed Set Based Discovery of Representative Association Rules. In: 4th International Conference on Advances in Intelligent Data Analysis, pp. 350–359. Springer, Heidelberg (2001)
8. Xiong, T., Ning, P., Vipin, K.: Hyperclique pattern discovery. Data Mining and Knowledge Discovery 13, 219–242 (2006)
9. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovering frequent episodes in sequences. In: The First International Conference on Knowledge Discovery and Data Mining (KDD 1995), pp. 210–215. AAAI Press, Menlo Park (1995)
10. Chen, R., Dave, K., McAvoy, T.J., Luyben, M.: A Nonlinear Dynamic Model of a Vinyl Acetate Process. Ind. Eng. Chem. Res. 42, 4478–4487 (2003)

Branch and Bound Algorithms to Solve Semiring Constraint Satisfaction Problems

Louise Leenen^{1,2} and Aditya Ghose¹

¹ Decision Systems Laboratory, SCSSE, University of Wollongong, Australia

11916,aditya@uow.edu.au

² CSIR, South Africa

lleenan@csir.co.za

Abstract. The Semiring Constraint Satisfaction Problem (SCSP) framework is a popular approach for the representation of partial constraint satisfaction problems. Considerable research has been done in solving SCSPs, but limited work has been done in building general SCSP solvers. This paper is part of a series in which incremental changes are made to a branch and bound (BnB) algorithm for solving SCSPs. We present two variants of a BnB algorithm: a backjumping algorithm and a forward checking algorithm. These algorithms are based on the maximal constraints algorithms of Freuder and Wallace [1], and we show they perform better than the BnB algorithm on some problem instances.

1 Introduction and Background

Semiring Constraint Satisfaction is a general framework for constraint satisfaction where classical CSPs, Fuzzy CSPs, Partial CSPs, and others (over finite domains) can be cast [2]. Existing work on SCSP methods include [3,4,5,6,7]. Previously we presented a BnB algorithm to solve SCSPs [8] which is described in Section 2. Section 3 contains the main contribution of this paper: two new algorithms to solve SCSPs: a backjumping and a forward checking algorithm. In Section 4 we test the algorithms. An overview of the SCSP framework follows.

Definition 1. A *c-semiring* is a tuple $S = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ such that

- A is a set with $\mathbf{0}, \mathbf{1} \in A$;
- $+$ is defined over (possibly infinite) sets of elements of A as follows ¹:
 - for all $a \in A$, $\sum(\{a\}) = a$;
 - $\sum(\emptyset) = \mathbf{0}$ and $\sum(A) = \mathbf{1}$;
 - $\sum(\bigcup A_i, i \in I) = \sum(\{\sum(A_i), i \in I\})$ for all sets of indices I ;
- \times is a commutative, associative, and binary operation such that $\mathbf{1}$ is its unit element and $\mathbf{0}$ is its absorbing element;
- for any $a \in A$ and $B \subseteq A$, $a \times \sum(B) = \sum(\{a \times b, b \in B\})$.

¹ When $+$ is applied to sets of elements, we will use the symbol \sum in prefix notation.

The set A contains the preference values to be assigned to the constraint tuples. We derive a partial ordering \leq_S over the set A : $\alpha \leq_S \beta$ iff $\alpha + \beta = \beta$.² The minimum element is $\mathbf{0}$, and $\mathbf{1}$ is the maximum element.

Definition 2. A constraint system is a 3-tuple $CS = \langle S_p, D, V \rangle$, where $S_p = \langle A_p, +_p, \times_p, \mathbf{0}, \mathbf{1} \rangle$ is a c-semiring, V is an ordered finite set of variables, and D is a finite set containing the allowed values for the variables in V .

Definition 3. Given a constraint system $CS = \langle S_p, D, V \rangle$, where $S_p = \langle A_p, +_p, \times_p, \mathbf{0}, \mathbf{1} \rangle$, a constraint over CS is a pair $c = \langle \text{def}_c^p, \text{con}_c \rangle$ where $\text{con}_c \subseteq V$ (the type), and $\text{def}_c^p : D^k \rightarrow A_p$ (k is the cardinality of con_c). A Semiring Constraint Satisfaction Problem (SCSP) over CS is a pair $P = \langle C, \text{con} \rangle$ where C is a finite set of constraints over CS and $\text{con} = \bigcup_{c \in C} \text{con}_c$.

Definition 4. Given a constraint system $CS = \langle S_p, D, V \rangle$ with V totally ordered via \preceq , consider any k -tuple $t = \langle t_1, t_2, \dots, t_k \rangle$ of values of D and two sets $W = \{w_1, \dots, w_k\}$ and $W' = \{w'_1, \dots, w'_m\}$ such that $W' \subseteq W \subseteq V$ and $w_i \preceq w_j$ if $i \leq j$ and $w'_i \preceq w'_j$ if $i \leq j$. The projection of t from W to W' , written $t \downarrow_{W'}$, is defined as the tuple $t' = \langle t'_1, \dots, t'_m \rangle$ with $t'_i = t_j$ if $w'_i = w_j$.

Definition 5. Given a constraint system $CS = \langle S_p, D, V \rangle$ where $S_p = \langle A_p, +_p, \times_p, \mathbf{0}, \mathbf{1} \rangle$ and two constraints $c_1 = \langle \text{def}_{c_1}^p, \text{con}_{c_1} \rangle$ and $c_2 = \langle \text{def}_{c_2}^p, \text{con}_{c_2} \rangle$ over CS , their combination, written $c_1 \otimes c_2$, is $c = \langle \text{def}_c^p, \text{con}_c \rangle$ with $\text{con}_c = \text{con}_{c_1} \cup \text{con}_{c_2}$ and $\text{def}_c^p(t) = \text{def}_{c_1}^p(t \downarrow_{\text{con}_{c_1}}) \times_p \text{def}_{c_2}^p(t \downarrow_{\text{con}_{c_2}})$. Let $(\bigotimes C)$ denote $c_1 \otimes c_2 \otimes \dots \otimes c_n$ with $C = \{c_1, \dots, c_n\}$. Given a constraint $c = \langle \text{def}_c^p, \text{con}_c \rangle$ over CS , and a set I ($I \subseteq V$), the projection of c over I , written $c \downarrow I$, is the constraint $c' = \langle \text{def}_{c'}^p, \text{con}_{c'} \rangle$ over CS with $\text{con}_{c'} = I \cap \text{con}_c$ and $\text{def}_{c'}^p(t') = \sum_{\{t | t \downarrow_{I \cap \text{con}_c} = t'\}} \text{def}_c^p(t)$.

Definition 6. Given a SCSP $P = \langle C, \text{con} \rangle$ over CS , the solution of P is a constraint is $\text{Sol}(P) = (\bigotimes C) = \langle \text{def}_c^p, \text{con} \rangle$. The maximal solution of P is $MSol(P) = \{\langle t, v \rangle | \text{def}_c^p(t) = v \text{ and there is no } t' \text{ such that } v <_{S_p} \text{def}_c^p(t')\}$.

2 A Branch and Bound Algorithm for SCSPs

In this section we describe Algorithm 1 which appeared in [8]. Assume a SCSP $P = \langle C, \text{con} \rangle$ over $CS = \langle S_p, D, V \rangle$, where S_p has best/worst semiring values of $\mathbf{1}/\mathbf{0}$. The upper bound, UB , contains the semiring value of the best solution found so far and is initialised with $\mathbf{0}$. For each node in the search tree, its lower bound, LB is a semiring value associated with a search path from the root node up to a particular node, and is initialised with $\mathbf{1}$. If $LB \leq_{S_p} UB$, the subtree below the current node is pruned, and the algorithm backtracks to a node at a higher level in the tree.

Note that the starred parameters are variable parameters, and *Queue* contains the decision variables. In line 6 we calculate the semiring value *NewLB* associated with the current node by considering all the constraints which have *var* as a

² Singleton subsets of the set A are represented without braces.

member of its type, and is such that all the variables in their types have been instantiated (i.e. we do a back check). Simply combine the semiring values of the tuples of all these constraints. In line 7 we combine $NewLB$ with the lower bound value of the search path up to current node's ancestor, i.e. LB . In lines 8-11 we calculate an estimated associated (lower bound) semiring value for the search path below the current node up to a leaf. This estimated lower bound is a semiring value a such that $NewLB \otimes a$ is maximal. This value a is combined with $NewLB$. Note that the variables can be partitioned in disjoint sets, and each partition can be solved as a smaller instance of the original problem.

Algorithm 1. BnB(NoInstantiated*, Queue*, LB*, UB, BestSolution)

```

Require:  $V; C; D; S_p; N$  (number of variables).
1: if ( $NoInstantiated < N$ ) then
2:    $var = pop\ Queue$ ; [current decision variable]
3:   while ( $var\_domain$  not empty) do
4:      $var\_value =$  select best value from  $var\_domain$ ;
5:      $var\_domain = var\_domain - var\_value$ ;
6:      $NewLB =$  lower bound value for current node;
7:      $NewLB = \times_p(LB, NewLB)$ ; [lower bound value for search path including current node]
8:     if  $NoInstantiated \neq N-1$  then
9:        $FullLB = FindFullLB(NewLB)$ ; [estimated lower bound]
10:      else
11:         $FullLB = NewLB$ ; [complete assignment]
12:      if ( $UB <_{S_p} FullLB$ ) then
13:         $LB = FullLB$ ;
14:        if ( $NoInstantiated = N-1$ ) then
15:           $UB = LB$ ;
16:           $BestSolution =$  current assignment of values for decision variables;
17:          if ( $UB = 1$ ) then
18:            return 1;
19:          if ( $BBnB(Noinstantiated+1, Queue, LB, UB, BestSolution) = 1$ ) then
20:            return 1;
21: return 0;

```

Table 1. Constraint Definitions

t	$\langle 0, 0 \rangle$	$\langle 0, 1 \rangle$	$\langle 0, 2 \rangle$	$\langle 0, 3 \rangle$	$\langle 0, 4 \rangle$	$\langle 1, 0 \rangle$	$\langle 1, 1 \rangle$	$\langle 1, 2 \rangle$	$\langle 1, 3 \rangle$	$\langle 1, 4 \rangle$	$\langle 2, 0 \rangle$	$\langle 2, 1 \rangle$	$\langle 2, 2 \rangle$
$def_{c_1}^p(t)$	3	2	3	1	0	0	2	3	4	2	3	2	4
$def_{c_2}^p(t)$	3	2	3	0	2	1	1	1	3	3	1	4	0
$def_{c_3}^p(t)$	3	0	4	3	0	2	2	3	3	3	3	2	3
$def_{c_4}^p(t)$	4	0	1	2	2	2	2	1	2	2	2	0	2
$def_{c_5}^p(t)$	2	0	2	2	0	3	3	2	2	0	0	4	3
t	$\langle 2, 3 \rangle$	$\langle 2, 4 \rangle$	$\langle 3, 0 \rangle$	$\langle 3, 1 \rangle$	$\langle 3, 2 \rangle$	$\langle 3, 3 \rangle$	$\langle 3, 4 \rangle$	$\langle 4, 0 \rangle$	$\langle 4, 1 \rangle$	$\langle 4, 2 \rangle$	$\langle 4, 3 \rangle$	$\langle 4, 4 \rangle$	-
$def_{c_1}^p(t)$	1	2	2	2	0	0	0	0	2	2	1	2	-
$def_{c_2}^p(t)$	1	0	4	3	0	2	3	1	3	1	2	2	-
$def_{c_3}^p(t)$	0	2	3	3	0	2	0	0	3	1	3	4	-
$def_{c_4}^p(t)$	3	2	3	2	1	0	3	4	3	3	2	1	-
$def_{c_5}^p(t)$	0	0	0	3	0	2	2	1	3	3	4	0	-

Example 1. Consider a SCSP where $S_p = \langle \{0, \dots, 5\}, max, min, 0, 5 \rangle$, $V = con = \{A, B, C, D, E, F, G, H\}$, $D = \{0, 1, 2, 3, 4\}$, and $C = \{c_1, c_2, c_3, c_4, c_5\}$. Table 1 contains the constraint definitions with $c_1 = \langle def_{c_1}^p, \{A, B\} \rangle$, $c_2 = \langle def_{c_2}^p, \{C, D\} \rangle$, $c_3 = \langle def_{c_3}^p, \{E, A\} \rangle$, $c_4 = \langle def_{c_4}^p, \{F, G\} \rangle$, and $c_5 = \langle def_{c_5}^p, \{F, H\} \rangle$. The order of instantiation is: A, B, E, C, D, F, G, H.

Phase 1: Select tuple $\langle 1, 3 \rangle$ in row 2 to assign $A = 1$ ($LB = 4$). Then $B = 3$ ($LB = 4$). In row 10 select tuple $\langle 3, 1 \rangle$ to set $E = 3$. $LB = 3$. Then $C = 2$, $D = 1$, $F = 0$, $G = 0$, $H = 0$. $LB = 2$. $UB = 2$. **Phase 2:** Backtrack. $H=2$; $H=3$; backtrack; $G=3$; $G=4$; $G=2$; backtrack; $F=4$ ($LB=3$); $G=0$; $H=3$. $UB = 3$. **Phase 3:** $H=1$; $H=2$; $H=0$; backtrack; $G=1$; $G=2$; $G=3$; $G=4$; backtrack; $F=2$; $F=3$; $F=1$; backtrack; $D=0$; $D=3$; backtrack; $C=3$; $C=0$; $C=1$; $C=4$; backtrack; $E=4$; $E=1$; $E=2$; backtrack; $B=2$; $B=1$; $B=4$; backtrack; $A=2$ ($LB=4$); $B=2$; $E=0$; $C=2$; $D=1$; $F=0$. **Note phase 4:** $G=0$; $H=0$ ($LB=2$); backtrack; $H=2$; $H=3$; backtrack; $G=3$; $G=4$; $G=2$; backtrack. $F=4$ ($LB=4$); $G=0$; $H=3$. Maximal solution with associated semiring value of 4: $A=2$; $B=2$; $C=2$; $D=1$; $E=0$; $F=4$; $G=0$; $H=3$.

3 Two New Algorithms for Solving SCSPs

Backjumping (BJ) [9] remembers the depth of failure, i.e. the deepest level, l at which any of the values for a variable fails. When all values have been tried for a variable, BJ can proceed directly to the level l . The BJ algorithm for maximal constraint satisfaction (Max-CSP) [1] does not always jump back all the way to the deepest level of failure. If any values below the level l were inconsistent when chosen, it only jumps back to the deepest such level: other choices of values at level l may result in fewer inconsistencies. In our new BJ algorithm, Algorithm 2, *FailDepth1* contains the level where a variable's value may fail first (i.e. closer to the root.) *LD* (inconsistency depth) keeps track of the deepest level of failure for any value. *R_D* (return depth) is adjusted if a value fails: it is assigned the maximum value at which a failure has been detected. *R_D* is initialised with the value 0, and it controls the level to which recursion is rolled back. If the current node decreases *LB* (line 22), its level becomes the new *LD* value. *FailDepth2* (line 26) controls the extent to which recursion is rolled back.

Example 2. BJ and BnB proceed in exactly the same way up to Phase 4. After finding $UB = 3$, both algorithms backtrack until $A = 2$, and then extend the search up to the leaf, try various values for H and then backtrack. Here BnB tries various values for G , and backtracks before assigning $F = 4$. However, **BJ avoids backtracking from H to G : it backjumps from H to F .**

Forward checking (FC) [10] combines backtracking with a check for local consistency ahead in the search tree. In a partial CSP context, we require an initial value for a bound on the allowed number of unsatisfied constraints. For each value, the number of domains with no supporting values is counted and this arc consistency count (ac) is a lower bound on the increment in the expected number of unsatisfied constraints that will be incurred should this value be added to the solution. In a particular search path, the *ac* for a proposed value assignment to the current variable can be added to the number of unsatisfied constraints so far, and compared to the current bound *NB*. If the sum is not less than *NB*, then the current search path will not result in a better solution.

Our FC algorithm is similar to Algorithm 1 except for the addition just after line 5, of a call to *ForwardCheck(var,var-value)* to perform forward checking. This

procedure ensures that the newly instantiated decision variable has a consistent value in the domains of all (uninstantiated) variables that appear in the same constraints. For each of these constraints it ensures that every one of the uninstantiated variables of the constraint has at least one domain value such that the resulting value tuple's associated semiring value does not fail. Note that we have to restore domain values removed from domains during a forward check for a decision variable var with the value $var\text{-value}$, if this value fails.

Algorithm 2. $BJ(\text{NoInstantiated}^*, \text{Queue}^*, \text{Dom}^*, \text{LB}^*, \text{UB}, \text{BestSol}, \text{R_D}^*, \text{I_D}^*)$

Require: $V; C; S_p; N$.

- 1: if ($\text{NoInstantiated} < N$) then
- 2: $var = \text{pop Queue}$;
- 3: if ($var\text{-domain}$ not empty) then
- 4: $var\text{-value} = \text{select best value from } var\text{-domain}$;
- 5: $var\text{-domain} = var\text{-domain} - var\text{-value}$;
- 6: $NewLB = \text{lower bound for current node}$;
- 7: $NewLB = \times_p(LB, NewLB)$;
- 8: $FailDepth1 = \text{first node form root where } var\text{-value fails}$
- 9: if $\text{NoInstantiated} \neq N-1$ then
- 10: $FullLB = \text{FindFullLowerBound}(NewLB)$; [estimated lower bound]
- 11: else
- 12: $FullLB = NewLB$; [complete assignment]
- 13: else
- 14: return R_D ;
- 15: if ($UB <_{S_p} FullLB$) then
- 16: $FailDepth1 = \text{NoInstantiated}$;
- 17: if ($\text{NoInstantiated} = N-1$) [complete assignment] then
- 18: $UB = FullLB$;
- 19: $BestSol = \text{current assignment of values for decision variables}$;
- 20: if ($UB = 1$) then
- 21: return N ; /* finished */
- 22: if ($FullLB <_{S_p} LB$) then
- 23: $I_D = \text{NoInstantiated}$;
- 24: $FailDepth2 = BJ(\text{NoInstantiated}+1, \text{Queue}, \text{Dom}, FullLB, UB, BestSol, 0, I_D)$;
- 25: if ($FailDepth2 < \text{NoInstantiated}$) or ($FailDepth2 = N$) then
- 26: return $FailDepth2$;
- 27: else
- 28: $R_D = \max(FailDepth1, R_D, I_D)$;
- 29: return $BJ(\text{NoInstantiated}, \text{Queue}, \text{Dom}, LB, UB, BestSol, R_D, I_D)$;
- 30: $R_D = \max(FailDepth1, R_D, I_D)$;
- 31: return $BJ(\text{NoInstantiated}, \text{Queue}, \text{Dom}, LB, UB, BestSol, R_D, I_D)$;
- 32: return $\text{NoInstantiated} - 1$;

Example 3. FC and BnB both find $UB = 3$, backtrack and try 1 and 2 for H . BnB then sets $H = 0$, but **FC eliminates this value from the domain of H as part of its forward checking** at the point were it set $F = 4$. It used constraint c_4 to find this value for F , but value 0 for H fails in c_5 during forward checking. Both algorithms then backtrack to G and try values 1 and 2. BnB also tries 3 and 4 for G , while **FC has eliminated both these values from the domain of G because they fail in c_4** . Both algorithms proceed in the same way until Phase 4. After $A = 2$, BnB extends the search down to the leaf node, where $H = 0$. FC only extends the search up to $G = 0$. **FC removed all values from the domain of H when F was instantiated**. FC then backtracks one level, but **all values for G have been removed**, so it backtracks and set $F = 4$.

4 Experiments and Conclusion

We compare the BJ and FC algorithms (without variable partitioning) to the BnB algorithm with (BnPPart) and without variable partitioning. The experiments were performed on an Intel Core 2 Duo processor at 2GHz with 2GB RAM. We solved 3 sets of randomly generated binary SCSPs that are instances of fuzzy CSPs. All the problems have $S_p = \langle \{0, 0.3, 0.5, 0.8, 1\}, max, min, 0, 1 \rangle$, and 10 domain values. The problems in sets 1/ 2/ 3 have 80/100/120 variables, and 10/10/20 constraints. All sets have 50 problem instances with a tightness of 70% and 50 instances with a tightness of 90%. Set 3 also has 50 instances with a tightness of 85%. A tightness (T) of x% means that (100-x)% of all the tuples have been assigned the best semiring value. Table 2 shows the average runtimes. There is not a significant difference in the performance of the different algorithms for smaller problems, but the results of FC and BJ on Set 3 with 90% tightness are much better than BnB's results. Note that BPart's runtimes are at most 3 times faster than those of BnB, but on average at least 25 times faster than CON'FLEX [4], a fuzzy CSP solver.

We presented two new algorithms for SCSPs that are extensions of the well-known Max-CSP algorithms. Our new algorithms perform better than the BnB algorithm on some problems. In the future, we plan to develop a backmarking algorithm and to test our algorithms on non-trivial SCSPs.

Table 2. Average runtimes of the three algorithms (in seconds)

Set	1 T 70%	1 90%	2 70%	2 90%	3 70%	3 85%	3 90%
BnB	0.0124	0.0552	0.0206	0.0544	0.0688	0.1518	0.8078
BJ	0.0102	0.0774	0.0210	0.0772	0.0778	0.2142	0.6596
FC	0.0194	0.0696	0.0216	0.0698	0.0876	0.2056	0.4758
BPart	0.0164	0.0346	0.0204	0.0382	0.061	0.0916	0.3234

References

1. Freuder, E.C., Wallace, J.W.: Partial constraint satisfaction. *AI* 58, 21–70 (1992)
2. Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based constraint solving and optimization. *Journal of the ACM* 44(2), 201–236 (1997)
3. Delgado, A., Olarte, C., Perez, J., Rueda, C.: Implementing semiring-based constraints using Mozart. In: Van Roy, P. (ed.) *MOZ 2004*. LNCS, vol. 3389, pp. 224–236. Springer, Heidelberg (2005)
4. Georget, Y., Codognet, P.: Compiling semiring-based constraint with *clp(fd,s)*. In: Maher, M.J., Puget, J.-F. (eds.) *CP 1998*. LNCS, vol. 1520, p. 205. Springer, Heidelberg (1998)
5. Bistarelli, S., Fargier, H., Montanari, U., Rossi, F., Schiex, T., Verfaillie, G.: Semiring-based CSPs and valued CSPs: Basic properties and comparison. *Constraints* 4, 199–240 (1999)
6. Bistarelli, S., Rossi, F., Pilan, I.: Abstracting soft constraints: Some experimental results on Fuzzy CSPs. In: *Proceedings of CLCSP 2003* (2003)

7. Bistarelli, S., Fung, S., Lee, J., Leung, H.: A local search framework for semiring-based constraint satisfaction problems. In: Proceedings of Soft 2003 (2003)
8. Leenen, L., Anbulagan, A., Meyer, T., Ghose, A.K.: Modeling and solving semiring constraint satisfaction problems by transformation to weighted semiring max-SAT. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 202–212. Springer, Heidelberg (2007)
9. Gaschnig, J.: Experimental case studies of backtrack vs. Waltz-type vs. new algorithms for satisficing assignment problems. In: Proceedings of CCSCSI 1978 (1978)
10. Haralick, R., Elliot, G.: Increasing tree search efficiency for constraint satisfaction problems. Artificial Intelligence 14, 263–313 (1980)

Image Analysis of the Relationship between Changes of Cornea and Postmortem Interval*

Fang Liu¹, Shaohua Zhu², Yuxiao Fu¹, Fan Fan²,
Tianjiang Wang¹, and Songfeng Lu¹

¹ School of Computer Science and Technology,
Huazhong University of Science and Technology,
Wuhan, China 430074

² Department of Forensic Medicine, Tongji Medical College,
Huazhong University of Science and Technology,
Wuhan, China 430030
Liufang727@hotmail.com

Abstract. Opacity of the cornea is one of the important indices for estimating time of death. Now the work is done by forensic medical experts. An unbiased estimation method is needed. This paper proposed a method for finding the relationship between changes of cornea and postmortem intervals by processing and analyzing images. Firstly, a histogram based image segmentation method is proposed to extract corneal regions from pictures of rabbit's eye. Secondly, texture and color features are used to describe the extracted corneal regions. Those features are carefully chosen to represent the changes of cornea in different postmortem intervals. A KNN classifier is used to reveal the association of image features and postmortem intervals. The experimental results show that cornea image features can be used to automatically estimate postmortem interval.

Keywords: cornea, postmortem interval, death time, image analysis.

1 Introduction

In practical work, cornea opacity observing is one of the most important assistant means for estimating death time [1]. But this task is considered as a difficult one due to the lack of quantified method. The accuracy of the observation result largely depends on personal experience. So some biological and chemical methods are proposed to quantify cornea opacity [1-3]. However, those methods need to take sample from cornea to get indices. Those operations may influence the natural change of cell's chemical composition in some extent. This paper prefers to go back to the appearance observation method. But this time we use a quantified method. Image processing and analyzing are employed to find features which can represent the quantified relationship between changes of cornea and postmortem intervals. Table 1 shows the relationship given by forensic medical experts [1]. Figure 1 gives the selected rabbit's eye

* This work was partially supported by HTRDP (Hi-Tech Research and Development Program of China) 2007AA01Z161.

pictures taken within the four postmortem intervals given in Table 1. Our method takes them as the guideline of corneal image feature extraction.

Table 1. Corneal appearance within different postmortem intervals

Postmortem interval	Description of corneal appearance
0~6(hour)	Transparent. The surface is covered with a few white spot.
6~12(hour)	Transparent. The number of the white spot increases. (light opacity)
12~24(hour)	Semitransparent. The surface is covered with thin white cloud. (mild opacity)
24~48(hour)	Opaque. Cornea is shrunk. (high opacity)

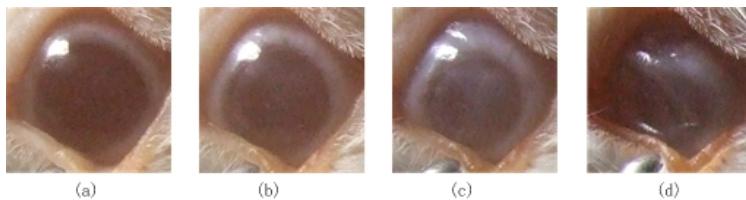


Fig. 1. Rabbit's left eye picture taken at different time after death. (a) 6:00 h. (b) 12:00 h. (c) 15:15 h. (d) 33:00 h.

In the research of cornea opacity quantification, only [4] presents an image processing based method. However, the researchers study diseased cornea of the living. The diseased cornea is covered by scar. But cornea of the dead is cloudy covered in the whole as shown in Figure 1. So the method in [4] is not applicable for us. The flow of our method is described as follows. (1) Cornea region segmentation. A histogram segmentation method is used in this step. (2) Feature selection. We test some features (including 1, 2, 3-order color moment based on HSV space; energy, inertia, entropy features based on co-occurrence Matrix), and two of them are carefully chosen to represent the changes of cornea in different postmortem intervals. One is gray-scale distance between corneal region and non-corneal region in one image. This feature can eliminate the influence of unsteady light source on images taken at different time. The other one is a texture feature --- energy ratio of low frequency. The extraction of texture feature is performed on Fourier Spectrum of images. (3) A KNN classifier is used to reveal the association of image features and time interval. Our goal is to prove some image features can effectively estimate the postmortem interval. So finding some other better classifier is our future work.

The rest of the paper is organized as follows. Section 2 gives the detail of cornea segmentation. Section 3 describes the feature extraction from corneal regions and the process of classify. Experimental results are showed in section 4. Conclusion remarks are given in last section.

2 Cornea Segmentation

Iris is covered with cornea. In a healthy eye, the color of iris is dark while the cornea is transparent. So the color of corneal region in image is dark. According to the fact, histogram segmentation is used to segment cornea region.

In this step, we need to extract corneal region from eye image by finding the uni-modal corresponds to cornea. The observation of all the test images shows that the highest peak of the gray-scale histogram corresponds to the corneal region. The unimodal with the highest peak provides us the gray-scale scope of the corneal region. A simple and efficient histogram segmentation method is used in our method to find the begin point and end point of a uni-modal [5].

Two morphological operations (open and close) are successively performed on the candidate region. Then a closed contour of cornea is detected by using the method in [6]. Corneal region can be extracted by the closed contour and original image. Figure 2 gives the whole process of cornea segmentation.

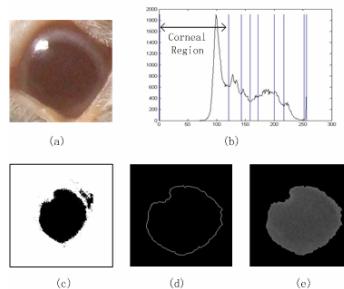


Fig. 2. The process of cornea segmentation. (a) image with rabbit's left eye, taken at 6 hours after death; (b) gray-scale histogram and its segmentation result of image shows in (a), the uni-modal containing highest peak refers to corneal region; (c) candidate corneal region extracted according to the result of histogram segmentation; (d) closed contour obtained by morphological operations and contour extraction; (e) gray-scale corneal region.

3 Feature Extraction

3.1 Color Feature

With the increasing of the postmortem time, the cornea becomes more and more opaque. The changing is also showed in gray-scale images of corneal region. The intensity of the corneal region is increasing slowly. Naturally gray-scale value of the corneal region is the best choice to reflect the corneal opacity. But unsteady light source influence the change of the gray-scale value. To avoid this, we will choose the difference of average gray-scale value between corneal region and non-corneal region as the feature. The gray-scale based feature GF is defined as Equation. (1).

$$GF = \left| u_c^2 - u_{nc}^2 \right| \quad (1)$$

where u_c and u_{nc} are the average gray-scale value of corneal region and non-corneal region respectively. Figure 3(a) shows the GF value of some samples which can reveal the change of GF value with time.

In the early stage of the postmortem time, the gray-scale difference is large between corneal region and non-corneal region. In the middle period of the time, the difference becomes less. And on the later stage, the gray-scale difference becomes large again. The reason is that the eye ball shrinks and sinks deeply in the orbit in the later period of the postmortem time. The region of eye ball becomes dark due to the lack of light. So the GF becomes large again.

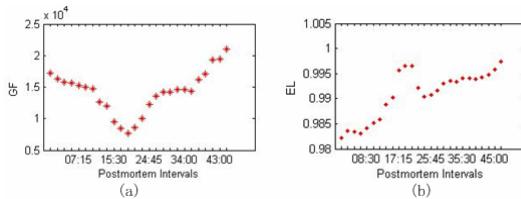


Fig. 3. (a) The change of GF value with the time. (b) The change of EL value with the time.

3.2 Texture Feature

Fourier transform is one of the techniques to describe the spatial characteristics of an image. We try to find the change of texture feature of corneal region by analyzing the power spectrum of corneal square region with time. However, the various sizes and the erratic edge of corneal regions make it difficult to compare the power spectrum of different images. Corneal square region with same size is needed to be segmented from original eye image. First, the geometric center of cornea is pointed from image with segmented cornea. The center is denoted as (x_c, y_c) . then taking (x_c, y_c) as center, a 78×78 square is obtained from the corresponding original eye image. Texture feature will be extracted from the square.

Images with smooth texture have large energy value on lower frequency intervals, while the images with coarse texture have large energy value on higher frequency intervals. To calculate energy on different frequency intervals, the power spectrum should be converted into polar coordinate system. In the system, origin of the coordinates is shifted to the center of the spectrum for easily observing. As we move away from the origin the low frequencies correspond to the slowly varying components of an image. As we move further away from the origin, the higher frequencies begin to correspond to faster and faster gray level changes in the image. Equation 2 gives the definition of energy $E(f_1, f_2)$ on frequency intervals $[f_1, f_2]$.

$$E(f_1, f_2) = \sum_{(u,v) \in cl(f_1, f_2)} P(u, v) \quad (2)$$

where $cl(f_1, f_2)$ refers to the circled area which inside radius is f_1 and the outside radius is f_2 , and $P(u, v)$ is the power spectrum of the region. The proportion of the energy E of the low frequency is chosen as the texture feature to describe the change of corneal texture. The texture feature EL of I_{rc} (image of corneal rectangle) is defined as equation 3.

$$EL = \frac{E(l, f_l)}{\sum_{(u,v) \in I_{fc}} P(u,v)} \quad (3)$$

where f_l is the upper boundary of the low frequency. In this paper, f_l is set as $2/3$ *value of the highest frequency in power spectrum.

Figure 3(b) shows the change of EL along with the increase of time. We can find that the curve has a rising trend, which means the energy of the Fourier spectrum is focusing on the low frequency more and more. It reflects the texture is becoming smooth with the time.

4 Experimental Results

The proposed method has been implemented in MATLAB script under Windows-XP on a PC with Celeron 2.5G Hz and 512M memory. The test dataset contain 188 images of Rabbit's left eye. The eye images were taken every 15 minutes after death. The total number of the images is 188.

First, each sample/image is labeled with a class ID according to the time after death. The class IDs are 1 (from 15 minutes to 12 hours and 30 minutes after death), 2 (from 12 hours and 45 minutes to 28 hours after death) and 3 (from 28 hours and 15 minutes to 47 hours after death). Then the 188 samples are divided into two parts. The size of training dataset is 127 samples. The rest 61 samples become test dataset. Table 2 gives the detail description of data set. KNN is performed five times on the randomly selected data sets. Figure 4 shows the best result of the 5 tests in $NEL-NGF$ space. NEL is the normalization of EL , and NGF is the normalization of GF . Table 3 gives the precision rates of total 5 tests.

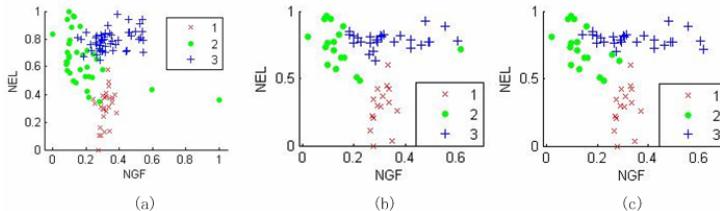


Fig. 4. Second Test. (a) The distribution of training dataset in $NEL-NGF$ space. (b) The distribution of test dataset in $NEL-NGF$ space. (c) The classification result.

Table 2. The description of data set

Class ID	Postmortem interval	Number of training samples	Number of test samples
1	00:15-12:30	32	15
2	12:45-28:00	42	20
3	28:15-47:00	53	26
Total	00:15-47:00	127	61

Table 3. Correction rates of total 5 tests. (KNN, k=3, euclidean distance)

Class ID	1	2	3	4	5
1	100%	100%	87%	93%	93%
2	85%	90%	95%	90%	80%
3	88%	92%	92%	77%	92%
Total	90%	93%	90%	85%	89%

5 Conclusion

The cornea becomes more and more opaque with the time after death. So the appearance of cornea can be used to estimate time of death. This paper proposes a quantified method for solving the problem. Image analysis and classifier are used to automatically find the relationship between changes of cornea and postmortem intervals. The encouraging experimental results shows features of cornea images can be used to estimate postmortem interval. Next, we want to find more effective features to improve the correct rate and the precision of time interval.

References

1. Chen, H., Zhu, S.H., Fang, D., et al.: An study on the relationship between changes of corneal AQP1 and postmortem interval. Forensic science and technology (2007)
2. Guolin, S., Jihai, S.: Study of comeal endothelium cell and its application in forensic medicine. Forensic science and technology (1), 36–37 (2001)
3. Changmiao, X., Haisong, Z., Songfu, Q.: The influence of temperature on the feature parameter of the endothelial cell living rate from corpse cornea [J]. Chinese Journal of Forensic Medicine 14(1), 33234 (1999)
4. Wang, X.W., Shen, L.S., Liu, D.H.: An Image Processing Based Method for Cornea Opacity Analysis. Journal of circuits and systems 7(2), 18–21 (2002)
5. Liu, F., Peng, X., Wang, T.: A Density-based Approach for Text Extraction in Images. In: Proc. of the 19th international conference on Pattern Recognition (December 2008) (to be published)
6. Fusheng, W., Guoqing, Q.: Boundary tracking algorithm of objects in binary image. Journal of Dalian Maritime University (2006)

Context-Based Term Frequency Assessment for Text Classification

Rey-Long Liu

Department of Medical Informatics
Tzu Chi University
Hualien, Taiwan, R.O.C.
rlliutcu@mail.tcu.edu.tw

Abstract. Automatic text classification (TC) is a fundamental component for information processing and management. To properly classify a document d , it is essential to identify semantics of each term t in d , while the semantics heavily depends on contexts (neighboring terms) of t in d . In this paper, we present a technique CTFA (Context-based Term Frequency Assessment) that improves text classifiers by considering term contexts in test documents. Results of the term context recognition are used to re-assess term frequencies, and hence CTFA may easily work with various kinds of text classifiers that base their TC decisions on term frequencies. Moreover, CTFA is efficient, and neither huge memory nor domain-specific knowledge is required. Experimental Results show that CTFA may successfully enhance performances of Rocchio and SVM (Support Vector Machine) classifiers on Reuters and Newsgroups data.

Keyword: Text classification, Term Contexts, Term Frequency Assessment.

1 Introduction

Automatic text classification (TC) is a fundamental component for information processing. For each document d , the classifier estimates the degree of acceptance (DOA) of d with respect to a category c (e.g., similarity between d and c , or probability of d belonging to c). Unfortunately, perfect DOA estimation could not be expected [2], since no classifiers may be perfectly trained and tuned.

In this paper, we explore how term context recognition (TCR) may improve DOA estimation so that performances of various kinds of text classifiers may be improved. We present a technique CTFA (Context based Term Frequency Assessment) to identify and encode term context information so that DOA estimation made by the text classifiers may be more proper. The idea is motivated by the fact that semantics of a term t in a document d often depends on context (neighboring) terms of t in d . By recognizing the term contexts, DOA estimation may be more proper.

The main challenge of TCR lies on identification of term contexts, including the locality, amount, and ordering of context terms. This was also the main focus of previous research efforts. For example, in text retrieval, the considerations of two adjacent terms [10] [11] and two nearby terms (e.g. in a locality distance smaller than 5 words [1]) were shown to be helpful in finding relevant documents. In text classification, typical

forms of term contexts included multiple consecutive terms (i.e. n-gram [3]), nearby terms in a fixed order (e.g. the sleeping experts, [5]), co-occurring terms in whatever order and location (e.g. RIPPER, [5]), and semantic features of nearby terms (recognized by information extraction and a dictionary [9]).

Unfortunately, the previous proposals have difficulties in tackling the challenges of mining for the term contexts and encoding of the term contexts. When mining for term contexts, they needed to consider a huge number of combinations of multiple terms, incurring a heavy computational loading and requiring a huge amount of memory. Moreover, they often also need to face the problem of data sparseness in the mining process, since meaningful term contexts do not always have enough number of occurrences in the training corpus. This is also the reason why dictionaries were employed in some previous studies (e.g. [9]). However, usage of a dictionary also limits the scope and the domain of the application.

Moreover, the previous proposals often did not encode term context information into a feature set, which is a set of selected terms or phrases on which text classifiers operate. An intelligent way to encode term context information into a feature set is essential, since various kinds of text classifiers may benefit from the encoding, without needing any modification to the classifiers.

2 Context-Based Term Frequency Assessment

CTFA aims to improve performances of text classifiers by TCR. In training, CTFA uses training documents to identify the correlation between each term and category. In testing, CTFA assesses frequencies of terms in the input document, and hence helps the underlying classifier to make more proper TC decisions for the document.

More specifically, in training, CTFA employs the χ^2 (chi-square) method to identify term-category correlation. For a term t and a category c , $\chi^2(t,c) = [N \times (A \times D - B \times C)^2] / [(A+B) \times (A+C) \times (B+D) \times (C+D)]$, where N is the total number of classifier-building documents, A is the number of classifier-building documents that are in c and contain t , B is the number of classifier-building documents that are not in c but contain t , C is the number of classifier-building documents that are in c but do not contain t , and D is the number of classifier-building documents that are not in c and do not contain t . Two types of correlation are identified: *positively correlated* and *negatively correlated* types. A term t is positively correlated to a category c , if $A \times D > B \times C$; otherwise it is negatively correlated to c .

In testing, upon receiving a test document, CTFA is invoked to assess the term frequency (TF) of each term, which is then passed to the underlying classifier to make TC decisions. Table 1 defines the algorithm for the TF assessment. For each category c , CTFA returns a TF vector for all distinct non-stop words in the test document (ref. Return). CTFA gives higher TF values to two kinds of terms in the test document d : (1) for positively-correlated terms, those that have many neighboring positively-correlated terms (ref. Step 4), and (2) for negatively-correlated terms, those that have many neighboring terms that are positively-correlated to some category other than c (ref. Step 5). The former helps c to accept d , while the latter helps c to reject d . Therefore, a term t would get a lower TF value if it just happens to appear in d with its neighboring terms *not* of the same correlation type. In that case, the occurrence of t should simply be a noise for text classification.

Table 1. Context-based term frequency assessment

```

Procedure CTFA( $d$ ), where  $d$  is a test document.
Return:  $\text{TF}_{n \times C}$  is a matrix, where  $n$  is the number of distinct terms in  $d$ ,  $C$  is the number of categories,
and  $tf_{i,c}$  is the assessed term frequency of term  $t_i$  with respect to category  $c$ .
Begin
  (1)  $S \leftarrow$  Sequence of terms in  $d$  with stop words removed;
  (2)  $n \leftarrow$  Number of distinct terms in  $S$ ;
  (3)  $\text{TF}_{n \times C} \leftarrow \mathbf{0}$ ;
  // Assess TF for each distinct positively-correlated term
  (4) For  $i = 1$  to  $n$ , do
    (4.1) For each term  $t_{i,j}$  that is the  $i^{\text{th}}$  distinct term occurring at the  $j^{\text{th}}$  position in  $S$ , do
      (4.1.1) For each category  $c$ ,
        (4.1.1.1) If  $t_{i,j}$  is positively correlated to  $c$ ,
          (4.1.1.1.1) PositiveTypeNum  $\leftarrow 0$ ;
          (4.1.1.1.2) For  $k = j - 1$  to 1, do
            (4.1.1.1.2.1) If the  $k^{\text{th}}$  term in  $S$  is positively correlated to  $c$ , PositiveTypeNum  $\leftarrow$ 
              PositiveTypeNum + 1;
            (4.1.1.1.2.2) Else exit for;
          (4.1.1.1.3) For  $k = j + 1$  to  $|S|$ , do
            (4.1.1.1.3.1) If the  $k^{\text{th}}$  term in  $S$  is positively correlated to  $c$ , PositiveTypeNum  $\leftarrow$ 
              PositiveTypeNum + 1;
            (4.1.1.1.3.2) Else exit for;
          (4.1.1.1.4)  $tf_{i,c} \leftarrow tf_{i,c} + \text{Log}_2(\text{PositiveTypeNum} + 2)$ ;
  // Assess TF for each distinct negatively-correlated term
  (5) For  $i = 1$  to  $n$ , do
    (5.1) For each category  $c$ , do
      (5.1.1) If  $tf_{i,c} = 0$ ,  $tf_{i,c} \leftarrow \text{Max}\{tf_{i,c}, \text{for all categories}\}$ ;
  (6) Return  $\text{TF}_{n \times C}$ ;
End.

```

More specifically, for a term t that is positively correlated to a category c , CTFA counts the number of left neighboring terms (ref. Step 4.1.1.1.2) and right neighboring terms (ref. Step 4.1.1.1.3) that are positively correlated to c . The higher the number is, the larger the TF value of t with respect to c will be, and if the number is 0, the TF value is 1.0 (ref. Step 4.1.1.4). For a term that occurs at multiple positions in the test document, its final TF value is the sum of its TF values at the positions (ref. Steps 4.1 and 4.1.1.4). On the other hand, for a term t that is negatively correlated to a category c , its TF value with respect to c is simply the maximum TF value among its TF values with respect to those categories to which t are positively correlated (ref. Step 5). The maximum TF value actually indicates the extent to which t suggests other categories to accept the test document (and hence suggests c to reject the test document as well).

3 Experiments

Experimental data includes Reuter-21578¹ and 20 Newsgroups², which were popular data collections. Reuter-21578 has 135 categories (topics). We follow [8] to set up the

¹ <http://www.daviddlewis.com/resources/testcollections/reuters21578>

² <http://people.csail.mit.edu/jrennie/20Newsgroups/>

training and testing data, leading to a training set with 7780 documents and a test set with 3022 documents. On the other hand, the 20 Newsgroups data contains 19997 documents of 20 topics (categories). As in [4], we employ 75% of the documents for training (14997), and the remaining 25% for testing (5000).

Precision (P), recall (R), and F_1 are employed as the criteria to evaluate the classifiers. P is [total number of correct classifications / total number of classifications

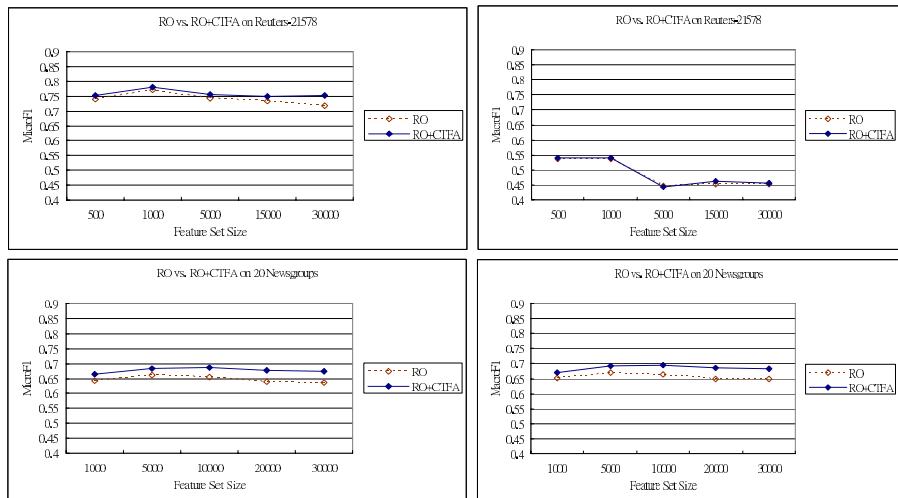


Fig. 1. Contributions of CTFA to RO

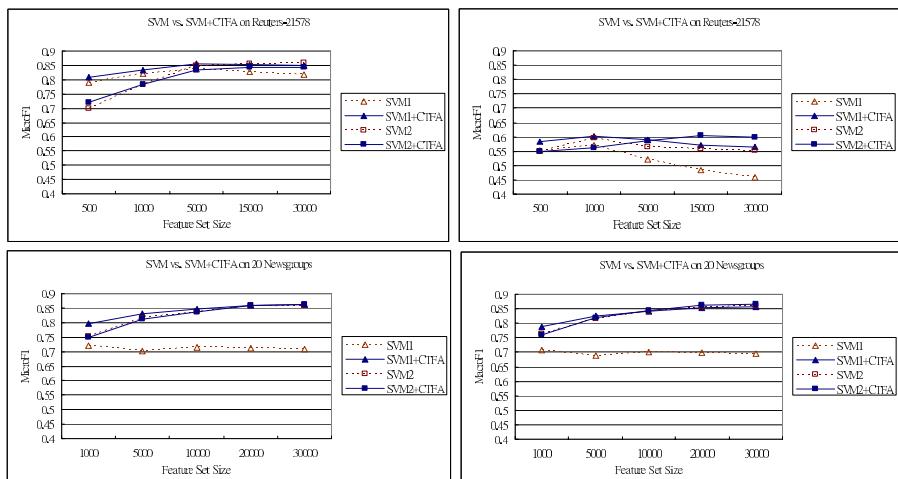


Fig. 2. Contributions of CTFA to SVM

made], R is [total number of correct classifications / total number of correct classifications that should be made], and F_1 is $2 \times P \times R / (P + R)$. As in many previous studies, we try two ways to compute average performances on P , R , and F_1 : micro-averaged and macro-averaged ways. Also noted that, in the experiment, P and R are set to 1.0 when their denominators are zero (and hence when P and R are incomputable, F_1 is 1.0 [7]). When both P and R are zero, F_1 is zero as well.

Each category c is associated with a classifier, which is based on the Rocchio method (RO) and the Support Vector Machine (SVM). We follow [8] to set RO, and employ SVM^{Light} to implement SVM³ [6]. We report results from two versions of SVM: (1) SVM1 that employs the default setting of SVM^{Light}, and (2) SVM2 that sets the “cost-factor” parameter to 10 (i.e. relative weights of errors on positive examples to errors on negative examples). Both RO and SVM required a fixed (predefined) feature set, which was built using the χ^2 (chi-square) weighting technique. CTFA is applied to enhancing RO, SVM1 and SVM2, and the resulting systems are named RO+CTFA, SVM1+CTFA, and SVM2+CTFA, respectively.

Fig. 1 shows the performances of RO and RO+CTFA. On Reuters-21578, CTFA helps RO to achieve better micro-averaged F_1 under all feature set sizes, especially when the feature set size becomes larger. Therefore, by considering term contexts, CTFA helps to reduce the effect of noises in test documents. Moreover, on 20 Newsgroups, CTFA helps RO to achieve better performances in both micro-averaged F_1 and macro-averaged F_1 .

Fig. 2 shows the results of applying CTFA to SVM. On 20 Newsgroups, SVM1 performs much worse than SVM2, while after introducing CTFA, SVM1 may achieve similar performance as SVM2, which is quite similar to the performance of well-tuned SVM reported in previous studies (e.g. [4]). Therefore, with the help of CTFA, even the default setting of SVM may achieve similar performances of best-tuned SVM. Moreover, on Reuters-21578, CTFA greatly improves macro-averaged F_1 of SVM1 when SVM1 achieves its best micro-averaged F_1 (i.e. when feature set size is 5000). Similarly, when SVM2 achieves its best micro-averaged F_1 (i.e. when feature set size is 30000), CTFA greatly improves its macro-averaged F_1 as well (improving from 0.5525 to 0.5973).

4 Conclusion

We propose a novel technique CTFA that identifies and encodes term contexts for text classification. The basic idea is: existence of a term t in a document d should not be a good evidence to classify d into a category c , if t happens to appear in d with its context term not suggesting the same correlation type to c . The idea relieves term context recognition from the problems of data sparseness and over-fitting, and there is no need for huge memory, expensive computation, and domain-specific knowledge. Moreover, CTFA encodes the term context information into term frequencies, which are common input for text classifiers, making CTFA able to improve various kinds of text classifiers without requiring any revisions to the classifiers. Empirical evaluation justifies the contributions of CTFA.

³ http://www.cs.cornell.edu/People/tj/svm%5Flight/old/svm_light_v5.00.html

Acknowledgments. This research was supported by the National Science Council of the Republic of China under the grant NSC 96-2221-E-320-001-MY3.

References

1. Alvarez, C., Langlais, P., Nie, J.-Y.: Word Pairs in Language Modeling for Information Retrieval. In: Proceedings of RIAO (Recherche d'Information Assistée par Ordinateur), University of Avignon (Vaucluse), France, pp. 686–705 (2004)
2. Arampatzis, A., Beney, J., Koster, C.H.A., van der Weide, T.P.: Incrementality, Half-life, and Threshold Optimization for Adaptive Document Filtering. In: Proceedings of the 9th Text Retrieval Conference, Gaithersburg, Maryland, pp. 589–600 (2000)
3. Caropreso, M.F., Matwin, S., Sebastiani, F.: Statistical phrases in automated text categorization. Technical Report IEI-B4-07-2000, Istituto di Elaborazione dell'Informazione, Pisa, IT (2000)
4. Chakrabarti, S., Roy, S., Soundalgekar, M.V.: Fast and accurate text classification via multiple linear discriminant projections. *The VLDB Journal* (2003)
5. Cohen, W.W., Singer, Y.: Context-Sensitive Mining Methods for Text Categorization. In: Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval, Zurich, Switzerland, pp. 307–315 (1996)
6. Joachims, T.: Making Large-Scale SVM Learning Practical. In: Schölkopf, B., Burges, C., Smola, A. (eds.) *Advances in Kernel Methods - Support Vector Learning*. MIT-Press, Cambridge (1999)
7. Lewis, D.D., Schapire, R.E., Callan, P., Papka, R.: Training Algorithms for Linear Text Classifiers. In: Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval, Zurich, Switzerland, pp. 298–306 (1996)
8. Liu, R.-L.: Dynamic Category Profiling for Text Filtering and Classification. *Information Processing & Management* 43(1), 154–168 (2007)
9. Riloff, E., Lehnert, W.: Information Extraction as a Basis for High-Precision Text Classification. *ACM Transactions on Information Systems*, 12(3) (1994)
10. Srikanth, M., Srihari, R.: Biterm Language Models for Document Retrieval. In: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval, Tampere, Finland (2002)
11. Wang, X., McCallum, A., Wei, X.: Topical N-grams: Phrase and Topic Discovery, with an Application to Information Retrieval. In: Proceedings of the IEEE 7th International Conference on Data Mining, Omaha NE, USA, pp. 697–702 (2007)
12. Yang, Y., Lin, X.: A Re-examination of Text Categorization Methods. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, Berkeley, California, pp. 42–49 (1999)

Outlier Mining on Multiple Time Series Data in Stock Market^{*}

Chao Luo, Yanchang Zhao, Longbing Cao, Yuming Ou, and Li Liu

Faculty of Engineering & IT, University of Technology, Sydney, Australia
`{chaoluo,yczhao,lbcao,yuming,liliu}@it.uts.edu.au`

Abstract. With the dramatic increase of stock market data, traditional outlier mining technologies have shown their limitations in efficiency and precision. In this paper, an outlier mining model on stock market data is proposed, which aims to detect the anomalies from multiple complex stock market data. This model is able to improve the precision of outlier mining on individual time series. The experiments on real-world stock market data show that the proposed outlier mining model is effective and outperforms traditional technologies.

Keywords: Outlier mining, time series, stock market.

1 Introduction

In stock market, the key surveillance function is identifying market anomalies, such as insider trading and market manipulation, to provide a fair and efficient trading platform [2,6]. Insider trading refers to the trades on privileged information unavailable to the public [8]. Market manipulation refers to the trade or action which aims to interfere with the demand or supply of a given stock to make the price increase or decrease in a particular way [3].

Recently, new intelligent technologies are required to deal with the challenges of the rapid increase of stock data. Outlier mining technologies have been used to detect market manipulation and insider trading . The objective of outlier mining is to find the data objects which are grossly different from or inconsistent with the majority of data. However, in stock market data, outliers are highly intermixed with normal data [4] and it is difficult to judge whether an object is an outlier or not. Therefore, a more effective and more efficient approach is in demand.

This paper presents a new technique for outlier detection on multiple time series data in stock market. At first, principal curve algorithm is used to detect the outliers from individual measurements of stock market. Then, the generated outliers are measured with the probability of being real alerts. To improve the accuracy and precision, these outliers are combined by some rules associated with the domain knowledge. The experimental results on real stock market data show that the proposed model is feasible in practice and achieves a higher accuracy and precision than traditional methods.

* This work was partly supported by the Australian Research Council (ARC) Linkage Project LP0775041 and Discovery Projects DP0667060 & DP0773412.

2 Related Work

A qualified surveillance function is expected to capture all the anomalies from a large amount of complex market records, while avoiding false alerts so as to reduce the waste of time and human resources [1]. The methods of generating alerts are typically rule-based approaches. Whenever an actual value is above a predetermined threshold, a specific alert will be triggered. In addition, statistical methods are also used to improve the effectiveness of surveillance by analysing the mean and standard deviation of values.

Recently, several researches made a valuable progress in theory to improve the effect of surveillance with information technologies. Palshikar et al. [9] studied collusion set detection using graph clustering. A set of traders is a candidate collusion set when they have heavy trading among themselves, as compared to their trading with others. They proposed a new graph clustering algorithms for the above problem. Lee and Yang [5] provided a prototype artificial immune abnormal trading detecting System (AIAS), which aims to detect the abnormal trading in stock markets. An effective method to evaluate the outlier is a variance-based outlier mining model (VOMM) proposed by Qi and Wang [10]. In their model, outliers are viewed as the top k samples holding maximal abnormal information in a dataset. The VOMM is executed by using principle curve algorithm. Their experiments on real-world dataset show that it performs better than the Gaussian model and GARCH model. VOMM detects outliers based on stock price only, which has limited information about stock market. We will present a new technique to improve VOMM for more effective outlier detection on multiple time series data in stock market.

3 Outlier Mining on Multiple Time Series (OMM)

There are multiple measures in stock market, which are price, volume, volatility, and so on, and each measure makes a time series. In order to combine multiple time series in stock market efficiently, it is necessary to first choose appropriate data based on financial knowledge in stock market and then define a quantitative measurement of outliers in stock market. The price movement and trading amount are regarded as good measurements for anomalies [7]. The price movement can be measured by price return and price fluctuation range during one day. Price fluctuation range is presented by the difference between the highest price and the lowest price in one day.

Voting-based OMM. Our first model is *Voting-based OMM (Voting-based Outlier Mining on Multiple time series)*. Let $D \subseteq R^n$ be the sample space of stock market. Let $T \subseteq D(\text{Card}(T) = n)$ be a set of samples drawn from D . A simple way to find the optimal function is majority voting. That is, every time series will be used to detect outliers individually, and then a day will be outputted as an outlier if an outlier was found in the day in the majority of all time series. In the voting-based OMM, the top k outliers are detected from individual time

series with the principal curve algorithm. It produces n candidate outlier sets $V_i, i = 1, 2, \dots, n$. Let the function $Counter(X)$ count the times X appear in $V_i, i = 1, 2, \dots, n$. Hence, $Counter(X) \in 1, 2, \dots, n, X \in (V_1 \cup V_2 \cup \dots \cup V_n)$. Let the function $f(X)$ to evaluate whether X is an outlier. If $Counter(X)$ indicate that X is the majority voting, $f(X) = 1$; otherwise, $f(X) = 0$.

Probability-based OMM. Our second model is *Probability-based OMM* (*Probability-based Outlier Mining on Multiple time series*). In probability-based OMM, we define a quantitative measurement of outliers. It is similar with the Dixon Ratio Test. Let HV be the test samples, AV be the average value of all samples which are less than the test samples and LV be the lowest value. Our test ratio R is calculated as

$$R = (HV - AV) / (HV - LV). \quad (1)$$

In probability-based OMM method, the top k outliers are chosen from individual time series with the principal curve algorithm. This produces n candidate outlier sets $V_i, i = 1, 2, \dots, n$. Then we calculate the outlier ratio R based on Formula (1). Let $Dis(X_i)$ be the distance between X_i and the generated principal curve, where $X_i \in T, i = 1, 2, \dots, k$. Let $LD = \min(Dis(X_i))$ be the lowest distance for all the X in V_i . Let $P(X_i) \in (0, 1)$ be the probability of X_i being an outlier, and we can get

$$P(X_i) = \frac{Dis(X_i) - \frac{1}{n-i} \sum_{j=i+1}^n Dis(X_j)}{Dis(X_i) - LD}, \quad (2)$$

where $i = 1, 2, \dots, k$.

For each $X \in (V_1 \cup V_2 \cup \dots \cup V_n)$, the outlier test ratios $P_1(X), P_2(X), \dots, P_n(X)$ are calculated corresponding to the n individual dimensions. Let the final $P(X)$ be the maximum value.

$$P(X) = \max(P_1(X), P_2(X), \dots, P_n(X)), X \in (V_1 \cup V_2 \cup \dots \cup V_n) \quad (3)$$

The final step is to sort descendingly the candidate outlier sets T according to $P(X)$, and then choose the top k samples as the outliers.

4 Experiments

Experimental Method. The experimental data are daily transaction records from Shanghai Stock Exchange in 425 trading days from 1 June 2004 to 3 Mar 2006. The attributes of the data sets include the daily highest price, the daily lowest price, the daily closing price and daily trade amounts. Daily price return and daily price fluctuation range are calculated based on the financial domain knowledge. We choose the real alerts generated by China stock exchange during the above timeframe as a benchmark for our experiments. By taking a day as abnormal if there are one or more alters during the day, the alerts are converted into 21 abnormal trading days. Hence, the trading days when alerts were found

Table 1. Comparison on the Number of Correctly Detected Outliers

Method	k=60	k=50	k=40	k=30	k=20	k=10
VOMM on Price Return	18	17	16	15	13	9
VOMM on Price Range	17	17	16	16	15	9
VOMM on Trade Amount	15	15	13	13	13	9
Voting-based OMM	20/52	20/45	17/32	17/27	16/19	9/9
Probability-based OMM	21	20	20	20	16	10

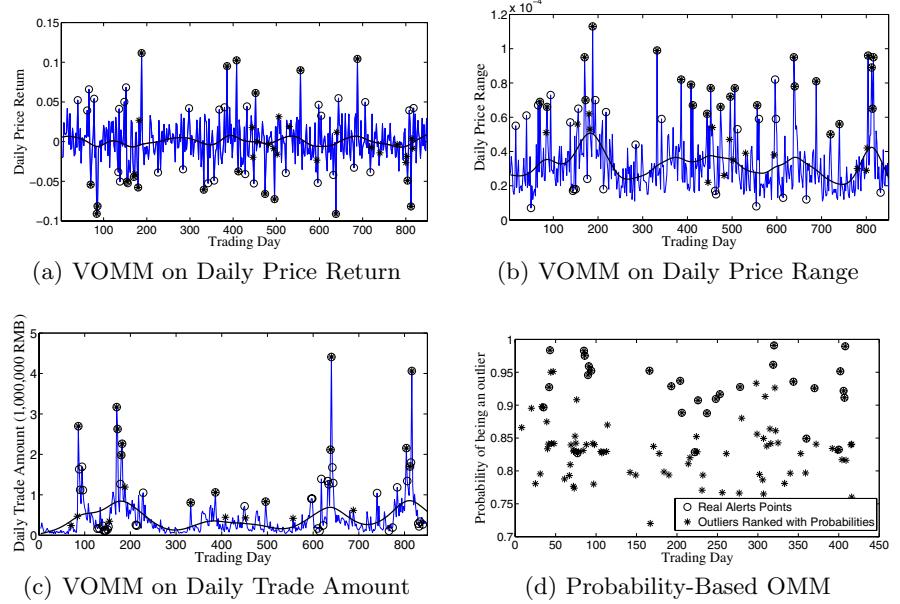
are regarded as exceptional days. Time Constraint Principal Component Analysis (TCP PCA) approach is used to preserve the temporal sequence information under the condition of the principal curve algorithm [11]. In our experiments, we set the scale factor $\lambda = 2$ according to the observations in experiments.

Experimental Results. The experimental results are shown in Table 1. The columns stand for the factor k in the above experiment, which indicates the expected number of outliers. For example, $k = 20$ means that the top 20 samples are regarded as outliers, while the rest of the samples are regarded as normal samples. The observation in each row stands for the number of alerts which are correctly identified by corresponding methods. For example, the value 16 on row 1 and column 4 means that 16 alerts are identified by VOMM methods on price return measures. One special case is the observation of Voting-based OMM, where the left value stands for the number of real alerts detected, while the right value stands for the calculated number of outliers.

Fig. 1(a), Fig. 1(b) and Fig. 1(c) show the results of VOMM on daily price return, daily price range and daily trade amount. The smooth curve passing through the middle of data sets is the generated principal curve. The X axis is the temporal sequence, which is tuned by the scale factor $\lambda = 2$. The Y axis is the value of individual measure. The samples marked only by “o” are expected outliers but not alerts, and the samples marked only by “*” are real alerts but not identified. The samples marked by both of “o” and “*” are identified real alerts. Fig. 1(d) shows the experimental results of Probability-based OMM. The X axis indicates the trading day, while the Y axis shows the probabilities of being an outlier. The samples marked by “o” are detected with $k = 60$. The samples marked by “*” show real alerts.

Experimental Result Analysis. The experimental results are analyzed based on four variables: True Positive (TP) stands for the number of detected outliers those are real alerts; False Positive (FP) stands for the number of detected which are not actual alerts; False Negative (FN) represents the number of the identified normal days which are real alerts and True Negative (TN) stands for the number of identified normal days which are not alerts. In order to compare the results in an intuitive way, we calculate the accuracy, specificity, precision and recall based on the following formulae:

$$\text{Accuracy} = (TP + TN) / (TP + FN + FP + TN) \quad (4)$$

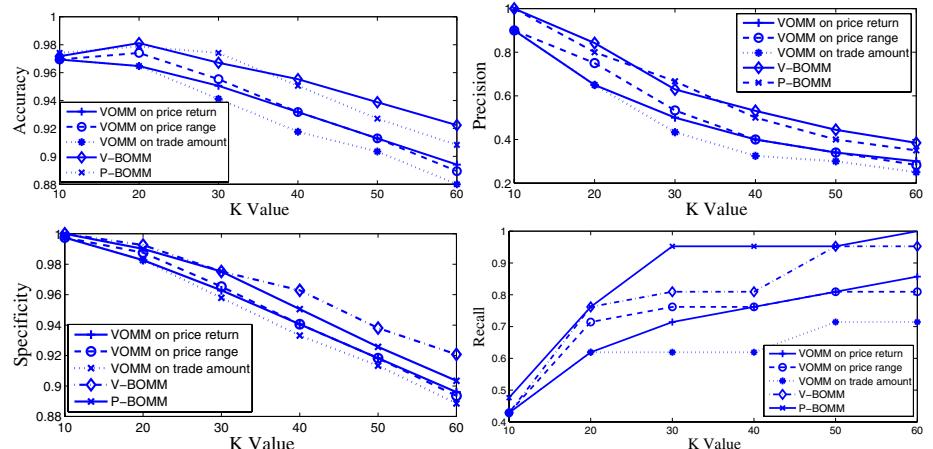
**Fig. 1.** Outliers Detected

$$\text{Specificity} = TN/(FP + TN) \quad (5)$$

$$\text{Precision} = TP/(TP + FP) \quad (6)$$

$$\text{Recall} = TP/(TP + FN) \quad (7)$$

Fig. 2 shows the accuracy, specificity, precision and recall of the models, where the X axis stands for k and the Y axis indicates the values of four measures. We

**Fig. 2.** Comparison on Accuracy, Specificity, Precision and Recall of Different Models

can see that the Voting-based OMM (V-BOMM) and Probability-based OMM (P-BOMM) have better performance than VOMM on all the four measures, no matter what value k is. Another finding is that the accuracy have the optimal results when $k=20$. With the increase of k , the precision and specificity decrease and the recall increases.

5 Conclusion

In this paper, we have studied outlier mining on multiple time series in stock market, and proposed two models for outlier mining on multiple time series (OMM). The experimental results show that our proposed models perform better in all the four measures than the previous outlier mining model VOMM on single time series. In future work, we will research on improving OMM to detect exceptional patterns on multiple time series, especially in stock market. Another potential future research is using microstructure knowledge to assist stock market surveillance.

References

1. Brown, P., GoldSchmidt, P.: Alcod idss: Assisting the Australian stock market surveillance teams review process. *Applied Artificial Intelligence* 10, 625–641 (1996)
2. Cheng, L., Firth, M., Leung, T., Rui, O.: The effects of insider trading on liquidity. *Pacific-Basin Finance Journal* 14, 467–483 (2006)
3. Dobson, M., Felixson, K., Pelli, A.: Day end returns-Stock price manipulation. *Journal of Multinational Financial Management* 9, 95–127 (1999)
4. Han, J., Kamber, M.: *Data Mining: concepts and techniques*. Morgan Kaufmann Publishers, San Francisco (2001)
5. Lee, V.C.S., Yang, X.J.: Development and test of an artificial-immune- abnormal-trading-detection system for financial markets. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 410–419. Springer, Heidelberg (2005)
6. Lucas, H.C.: Market expert surveillance system. *Communications of the ACM* 36, 27–34 (1993)
7. Meulbroek, L.K.: An emrirical analysis of illegal insider trading. *The Journal of Finance* 47, 1661–1699 (1992)
8. Minenna, M.: Insider trading abnormal return and preferential information: Supervising through a probabilistic model. *Journal of Banking and Finance* 27, 59–86 (2003)
9. Palshikar, G.K., Apte, M.M.: Collusion set detection using graph clustering. *Data Mining and Knowledge Discovery* 16, 135–164 (2008)
10. Qi, H., Wang, J.: A model for mining outliers from complex data sets. In: The 2004 ACM symposium on Applied computing, pp. 595–599. ACM, New York (2004)
11. Reinhard, K., Niranjan, M.: Parametric subspace modeling of speech transitions. *Speech Communication* 27, 19–42 (1999)

Generating Interactive Facial Expression of Communication Robots Using Simple Recurrent Network

Yuki Matsui¹, Masayoshi Kanoh², Shohei Kato¹, and Hidenori Itoh¹

¹ Nagoya Institute of Technology,
Gokiso-cho, Showa-ku, Nagoya, 466-8555, Japan

matsui@juno.ics.nitech.ac.jp

² Chukyo University,
101 Tokodachi, Kaizu-cho, Toyota, 470-0393, Japan
mkanoh@sist.chukyo-u.ac.jp

Abstract. To improve face-to-face interaction with robots, we developed a model for generating interactive facial expressions by using a simple recurrent network (SRN). Conventional models for robot facial expression use predefined expressions, so only a limited number of expressions can be presented. This means that the expression may not match the interaction and that the person may find the expressions monotonous. Both problems can be overcome by generating expressions dynamically. We tested this model by incorporating it into a robot and comparing the expressions generated with those of a conventional model. The results demonstrated that using our model increases the diversity of face-to-face interaction with robots.

1 Introduction

For robots to communicate smoothly with people, they need not only the ability to handle ordinary physical interactions but also to use *kansei* (*sensibility*). We have been working on various ways for robots to generate facial expressions as a component of their communication [1,2]. In general, robot facial expressions are generated by the intricate, coordinated movement of motors located in the robot's eyes, neck, and other areas. Since this requires much time and effort, the variety of facial expressions is limited. Moreover, the expressions lack the variety found in human expressions. When a person expresses an emotion, he or she makes a facial expression with a certain pattern and features. However, the expression is always slightly different. If a robot's facial expression is always the same for a particular emotion and lacks variety, the person interacting with the robot will find it unnatural. For robots to express a variety of natural facial expressions, they need a way to dynamically generate expressions corresponding to emotions by synthesizing facial expressions using predefined expressions.

When people express an emotion, the emotion is based on a past state. A person has the flexibility to be stimulated by external agents, and emotions are

generated by the transitions of the internal state. This leads to dynamic facial expression generation. To enable robots to exhibit dynamic facial expressions, we developed a model based on a simple recurrent network (SRN) [3] for generating facial expressions. An SRN generates output in accordance with previous state transitions. We implemented this model in the Ifbot [4] robot and experimentally investigated its ability to improve human-robot communication.

2 Interactive Facial Expression Model Using SRN

In general, a robot's facial expression is controlled by a motion file corresponding to the desired emotion. This conventional facial expression is static. Our interactive model, which enables a robot to generate a unique facial expression in response to each stimulus, is based on a neural network. We used a simple recurrent network [3] as the basis of our expression-generation model because it can generate expressions on the basis of past state transitions. Our proposed model is illustrated in Figure 1. The input is a stimulation event, and the output is a facial expression. A context layer and hidden layer pair is an emotion for which the robot outputs a facial expression considering changes in past stimulations. The hidden layer contains an internal representation for mapping a facial expression to the stimulation event. The relationship between a stimulation event and the facial expression is mapped by learning the network. The robot can thus express a facial expression corresponding to the stimulation event.

This model of learned time-oriented relationships between basic stimulation events and facial expressions can generate various facial expressions using the generalization capability of the neural network. When a stimulation event is input, the resulting facial expression is automatically generated because the network can learn and generate temporal patterns. Moreover, various mixed expressions can be synthesized corresponding to the timing of the stimulation event because the network has internal feedback. Without considering a specific interpolation method, the network can dynamically generate the facial expression.

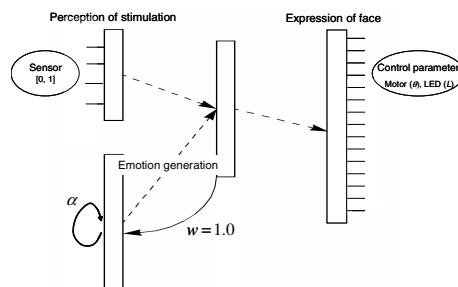
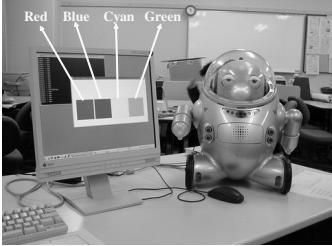
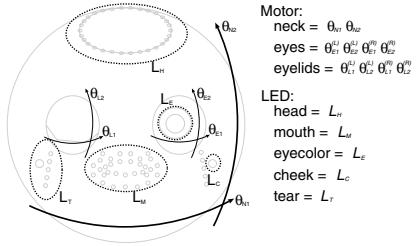


Fig. 1. Proposed interactive facial expression model

**Fig. 2.** Evaluation system using Ifbot**Fig. 3.** Facial expression mechanisms

3 Evaluation System

The system used for the evaluation is illustrated in Figure 2. It is based on the assumption that a stimulation event can occur at any time. An emotion is identified for the event, and the robot, Ifbot, generates an appropriate facial expression. A person controls the interactions with the robot by using a display screen. As shown in Figure 2, red, blue, cyan, and green buttons are displayed. They represent four emotions (anger, sadness, surprise, and happiness) adopted from six basic emotions [5]. The system inputs a 1 to the unit in the input layer corresponding to the button pressed inputs a 0 when there is no stimulation event. The values of the facial expression control parameters in the model are used to control the robot. The 15 control parameters are expressed as

$$S = (\theta_{N1}, \theta_{N2}, \theta_{E1}^{(L)}, \theta_{E2}^{(L)}, \theta_{E1}^{(R)}, \theta_{E2}^{(R)}, \theta_{L1}^{(L)}, \theta_{L2}^{(L)}, \theta_{L1}^{(R)}, \theta_{L2}^{(R)}, L_H, L_M, L_E, L_C, L_T)^T,$$

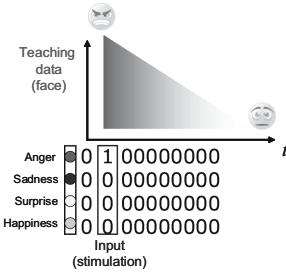
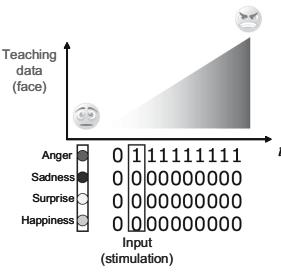
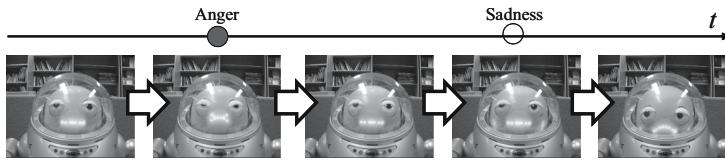
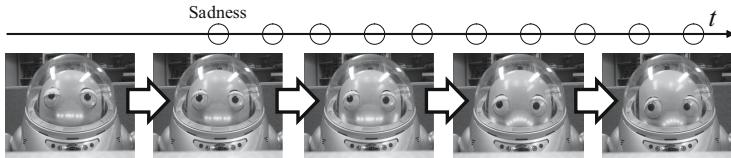
where $\theta^{(\cdot)}$ are motor outputs corresponding to $\theta^{(\cdot)}$, and $L.$ are patterns output from the LEDs of each part in Figure 3.

We used 4, 25, and 15 units in the input, hidden, and output layer, respectively (with 25 context units). The input units correspond to the four emotions. To train the network, we used four of Ifbot's facial expressions as teaching data.

3.1 Characterization by Learning

We created four robots (two proposed and two conventional) that can generate expressions corresponding to the four emotions.

Proposed A: *Proposed A* generated a facial expression for the appropriate emotion at the instant a stimulation event was input. The method used for training this robot (e.g. learning anger) is shown in Figure 4. For each input sequence, in which four bits were presented at a time, the correct output at the corresponding point in time is shown. When a stimulation event was input, the leftmost facial expression was generated. At time $t = 1$, the input unit corresponding to the stimulation event was input as 1.0, and the network trained the robot to generate the appropriate facial expression. For $t = 2 - 10$, the network gradually trained the robot to generate the default facial expression for when there was no stimulation event.

**Fig. 4.** Proposed A**Fig. 5.** Proposed B**Fig. 6.** Facial expressions using *Proposed A***Fig. 7.** Facial expressions using *Proposed B*

Proposed B: Proposed B gradually generated a facial expression for the appropriate emotion as the stimulation event proceeded. The method used for training this robot is shown in Figure 5. At time $t = 1$, the input unit corresponding to the stimulation event as continuously input as 1.0, and the network gradually trained the robot to generate the appropriate facial expression.

Default A and **B** operated the same as the proposed robot, except that it only generated facial expressions taken from the training data.

Figure 6 and Figure 7 show examples of face changes using our proposed model. You can see that face changes become more natural because each facial expression corresponds with stimulation.

4 Evaluation

We subjectively evaluated the impression that each robot made on a person interacting with them in comparison with that made by the conventional system. The participants were 28 college students. They interacted freely with the four robots one robot at a time for as long as they wanted. The evaluation was based

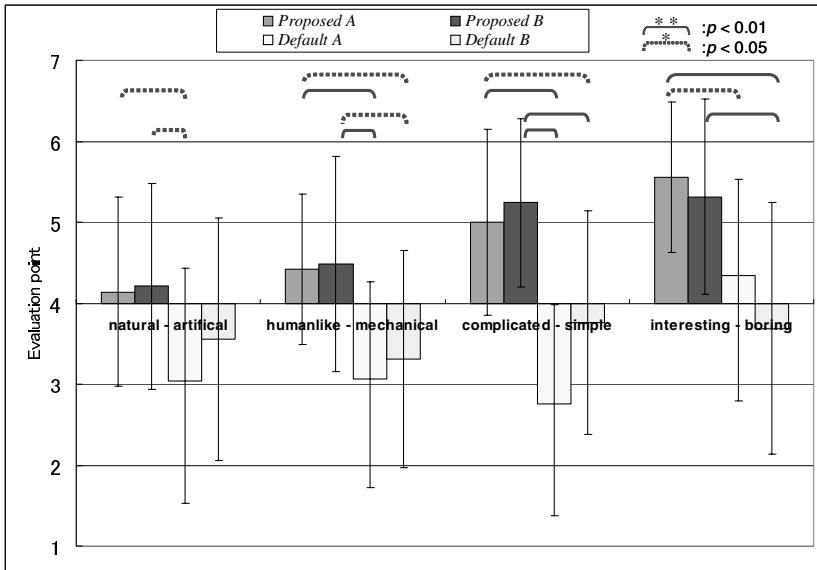


Fig. 8. Experiment result for (1) natural - artificial, (2) humanlike - mechanical, (3) complicated - simple, (4) interesting - boring

on a semantic differential (SD) technique in which values on a 7-point scale were assigned for six pairs of opposing evaluative adjectives.

4.1 Results

We tested the questionnaire using the Kruskal-Wallis test, a nonparametric one-way analysis of variance, and a Scheffe test, a test of statistical significance. The mean values for the evaluated pairs are shown in Figure 8.

natural - artificial: The difference in evaluations between the two proposed robots and *Default A* was significant. Both proposed robots received a neutral evaluation, while the two conventional robots tended to receive an artificial evaluation. This is attributed to the proposed method always changing the facial expression in accordance with the stimulation events and to the timing of the stimulation events. The proposed method can reduce the amount of artificial feeling in the interactions compared with the conventional method.

humanlike - mechanical: Both proposed robots were evaluated as humanlike, and both conventional ones were evaluated as mechanical; the difference was significant. We attribute this to the differences in the generated facial expressions between the proposed and conventional robots. The proposed robots displayed a variety of facial expressions because the expressions changed in accordance with the timing of the stimulation events. The proposed robots thus tended to create a less mechanical feeling. This was reflected in the descriptive impressions.

complicated - simple: The significant difference was the highest for this evaluation pair. Both proposed robots were evaluated as complicated, while both

conventional ones tended to be evaluated as simple. Several opinions were expressed in the descriptive impressions: “The actions of the conventional robots felt monotonous”; “Since a mixture of expressions was expressed, the actions of the proposed robots felt complex”. In short, the proposed robots were better able to generate expressions conveying mixed emotions.

interesting - boring: Although the Scheffe test did not reveal a significant difference between *Proposed B* and *Default A* ($p = 0.0507$), the differences were significant for the other combinations of proposed and conventional robots. Both proposed robots were evaluated as interesting, while both conventional robots were evaluated as neutral. These results suggest that the proposed model can increase the diversity of human-robot interaction.

5 Conclusion

We have developed an interactive facial expression model that uses the feedback and generalization capabilities of a simple recurrent network. Only basic expressions are made and trained; the network uses them to automatically generate similar facial expressions. We implement the proposed model in a robot and evaluated the effectiveness of the dynamic facial expression during interaction between a person and the robot. The results suggest that dynamically generating facial expression using the proposed method gives the person interacting with the robot a better impression than that using a conventional method. We showed that using the proposed model

1. reduces the artificial and mechanical feeling created by the facial expressions of a robot,
2. a robot can express interest and complicated (mixed) feelings, and
3. a robot can be characterized by using facial expressions.

Acknowledgment

This work was supported in part by Grant-in-Aid for Young Scientists (A) #20680014 of the Ministry of Education, Culture, Sports, Science and Technology, and Artificial Intelligence Research Promotion Foundation.

References

1. Kato, S., Ohshiro, S., Itoh, H., Kimura, K.: Development of a communication robot Ifbot. In: The 2004 IEEE International Conference on Robotics and Automation (ICRA), pp. 697–702 (2004)
2. Kanoh, M., Iwata, S., Kato, S., Itoh, H.: Emotive Facial Expressions of Sensitivity Communication Robot Ifbot. Kansei Engineering International 5(3), 35–42 (2005)
3. Elman, J.L.: Finding structure in time. Cognitive Science 14, 179–211 (1990)
4. Business Design Laboratory Co. Ltd. Communication Robot ifbot,
<http://www.ifbot.net>
5. Ekman, P.: Unmasking the Face. Prentice-Hall, Englewood Cliffs (1975)

Effects of Repair Support Agent for Accurate Multilingual Communication

Mai Miyabe¹, Takashi Yoshino², and Tomohiro Shigenobu³

¹ Graduate School of Systems Engineering, Wakayama University,
930 Sakaedani, Wakayama, Japan
s085051@sys.wakayama-u.ac.jp

² Faculty of Systems Engineering, Wakayama University,
930 Sakaedani, Wakayama, Japan
yoshino@sys.wakayama-u.ac.jp
<http://www.wakayama-u.ac.jp/~yoshino/>

³ Language Grid Project,
National Institute of Information and Communications Technology,
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan
shigenobu@nict.go.jp

Abstract. Translation repair plays an important role in intercultural communication using machine translation. It can be used to create messages that have very few translation mistakes. However, translation repair is a laborious task. It is important to carry out translation repair efficiently. Therefore, we propose a repair support agent that provides the segments that have not been translated accurately. We perform experiments on the translation repair efficiency to evaluate the effectiveness of the repair support agent. The results of these experiments are as follows. (1) Providing inaccurately translated segments improves the ability to detect inaccurate segments. (2) The inaccurate-judgment rate can affect the improvement of the efficiency of translation repair.

Keywords: Multilingual communication, machine translation, back translation, translation repair.

1 Introduction

Opportunities for multilingual communication have increased due to the development of the Internet. However, communication using nonnative language is difficult. When people communicate in their nonnative language, the language barrier prevents mutual understanding among communicating individuals[1]. To overcome the language barrier in communication, machine translation is used for communication using native language[2].

Despite recent advances in machine translation technology, it is still very difficult to obtain highly accurate translations. In machine translation, the probability of a translation mistake increases as the translation sentence lengthens. Translation mistakes prevent mutual understanding among communicating individuals. Therefore, for smooth communication, users have to create messages

that have very few translation mistakes. Translation repair plays an important role in multilingual communication using machine translation. It can be used to create messages that have very few translation mistakes.

Translation repair work is difficult when a user is presented with the translation of text in a nonnative language. Therefore, the use of back translation for translation repair is considered. However, translation repair is a laborious task. The support of efficient translation repair is required. Therefore, we propose a repair support agent that can reduce repair cost and provide inaccurately translated segments.

In this study, we evaluate the cost of translation repair by using the repair support agent. We verify the following from experiments.

1. Improvement in the efficiency of translation repair by using the repair support agent.
2. Factors that influence translation repair.

From the verification of the above points, we present the effectiveness of the repair support agent and the necessary support of adequate translation repair.

2 Structure of Repair Support Agent

Users find it difficult to carry out adequate translation repair if they cannot accurately detect inaccurately translated segments. If users repair the irrelevant parts, translation repair efficiency is down. Specifically, inadequate translation repair increases the time required for translation repair and delays the start of translation repair.

We propose a repair support agent that provides segments that have not been translated accurately. The repair support agent carries out morphological analysis.

In translation repair, the main technique for repairing sentences is the paraphrasing of words or expressions. Inaccurate back-translated sentences do not contain words that exist in the input sentence. We consider that providing the inaccurately translated segments improves the translation repair efficiency. The proposed repair support agent provides inaccurately translated segments by carrying out morphological analysis. This agent estimates inaccurately translated segments by comparing the words contained in an input sentence with the words contained in its back-translated sentence. We compare only nouns and verbs for estimation purposes.

Figure 1 shows the procedure of providing inaccurately translated segments. The procedure can be summarized as follows:

1. Obtain the back-translated sentence of an input sentence.
2. Carry out the morphological analysis of the input sentence and its back-translated sentence using the morphological analyzer MeCab[3].
3. Extract nouns and verbs that exist in the input sentence and do not exist in the back-translated sentence.
4. Consider the extracted words as inaccurately translated segments and highlight them in red.

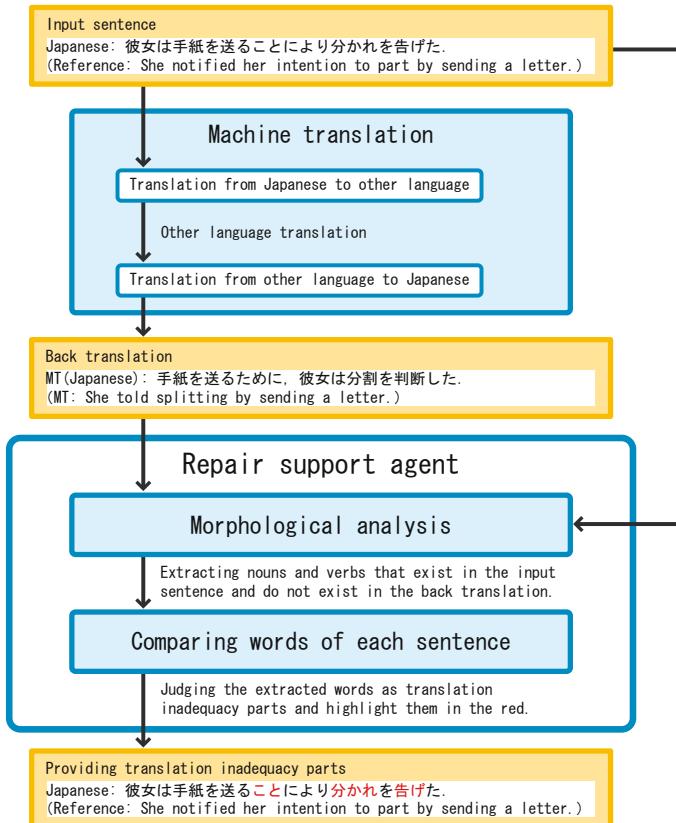


Fig. 1. Procedure of providing inaccurately translated segments

3 Experiment for Verification of Translation Repair Efficiency

3.1 Experimental Outline

We performed a translation repair experiment using the repair support agent in order to examine the translation repair efficiency. The subjects were 30 students from Wakayama University. The experimental tool used the J-Server developed by Kodensha[4], with Language Grid[5] as the machine translation system.

In the experiment, the subjects performed the following tasks.

Task A: Translation repair work with repair support agent.

Task B: Translation repair work without repair support agent.

In each task, subjects repaired 100 sentences. Fifteen subjects performed two tasks – Task A and Task B – in that order. The remaining subjects performed tasks in the reverse order. Hereafter, the first batch of 15 subjects is referred to as Group 1, and the second batch is referred to as Group 2.

Table 1. Repair time and delay in the start of the translation repair in repair work

		First task		Second task	
		Group1 Without re- pair support agent (s)	Group2 With repair support agent (s)	Group1 With repair support agent (s)	Group2 Without re- pair support agent (s)
Repair time	Average	108	103	87	83
	Standard deviation	96	82	74	68
	Significance probability	0.988		0.462	
Delay in the start of the translation repair	Average	23	20	20	21
	Standard deviation	20	29	19	24
	Significance probability	0.000*		0.584	

In the experiment, we considered only those sentences that contained 20-30 letters. We selected 200 sentences that required translation repair from material provided by NTT[6].

We allowed the subjects to terminate the translation repair of a sentence without translation repair when they concluded that the meaning of an unrepairsed back-translated sentence was the same as that of the original sentence. When the subjects were unable to repair a sentence for ensuring that the meaning of the back-translated sentence conformed to that of the original sentence after 5 min, we allowed them to abandon the translation repair halfway.

3.2 Results

We discuss the results on the basis of the data of 26 subjects because the data of 4 subjects was incorrect. The number of subjects in groups 1 and 2 was 12 and 14, respectively.

Repair Time and The Delay in the Start of the Translation Repair. Table 1 shows the averages, standard deviations, and significance probabilities of each measure of efficiency in each task.

1. Repair time

As shown in table 1, there is no significant difference between the repair time for Group 1 and the repair time for Group 2 in both tasks. Therefore, the repair support agent may not affect the reduction in the repair time.

2. Delay in the start of the translation repair

As shown in table 1, there is a significant difference between the delay in the start of the translation repair for groups 1 and 2 in the first task. In the second task, subjects started translation repair at almost the same time,

Table 2. Correlation between the inaccurate-judgment rate and each measure of repair efficiency

	Correlation coefficient	Significance probability
Repair time(first task)	-0.580	0.002*
Repair time(second task)	-0.656	0.000*
Delay in the start of the translation repair(first task)	-0.271	0.181
Delay in the start of the translation repair(second task)	-0.072	0.725

irrespective of the presence or absence of the repair support agent. The repair support agent reduces the delay in the start of the translation repair when inexperienced users start carrying out translation repair. Therefore, we consider that the repair support agent improves the ability to detect inaccurate segments.

Inaccurate-Judgment Rate of Subjects. In the experiment, the subjects do not have to carry out repair if the back translations are highly accurate. The number of accurate back-translated sentences is ten. If the subjects did not repair sentences other than the above-mentioned ten back-translated sentences, they were considered to have inaccurately detected the inaccurate parts.

The number of times subjects judge an inaccurate sentence to be an accurate sentence, is termed the “inaccurate-judgment rate.” It is calculated using the following equation:

$$R = \frac{N_u - N_{high'}}{N_{total} - N_{high} - N_g}$$

Here,

N_{total} is the total number of experimental sentences.

N_{high} is the number of accurate sentences.

N_g is the number of sentences that the subjects were unable to repair.

N_u is the number of unrepairs sentences.

$N_{high'}$ is the number of unrepairs and accurate sentences.

Table 2 shows the correlation coefficient between the inaccurate-judgment rate and each measure of repair efficiency. As shown in this table, the repair time is negatively correlated with the inaccurate-judgment rate.

Moreover, we have evaluated repaired translations by the subjective evaluation [7]. The correlation coefficients between the evaluated value and the inaccurate-judgment rate are -0.510 and -0.595 in first task and second task. The significance probabilities of the correlation coefficient are 0.008 and 0.001 in each task. Therefore, it can be observed that there is a negative correlation between the evaluated value and the inaccurate-judgment rate.

These results show that a high inaccurate-judgment rate may reduce the repair time and cause the debasement of the translation accuracy.

4 Conclusion

Translation repair plays an important role in intercultural communication using machine translation.

In this study, we have proposed a repair support agent that provides inaccurately translated segments by carrying out morphological analysis, as a tool that assists translation repair. We have performed experiments on the translation repair efficiency in order to evaluate the effectiveness of the repair support agent. In the experiment, we have discussed the experimental results on the basis of the following two measures of repair efficiency: repair time and delay in the start of the translation repair.

The results of these experiments are as follows.

1. Providing inaccurately translated segments improves the ability to detect inaccurate segments.
2. The inaccurate-judgment rate can affect the improvement of the efficiency of translation repair.

Acknowledgments. The authors express their sincere thanks to Ph.D. Naomi Yamashita for her comments on the data analysis in our experiment. This work was partly supported by Grant-in-Aid for Scientific Research (B), No. 19300036, 2007-2009.

References

1. Aiken, M.: Multilingual Communication in Electronic Meetings. ACM SIGGROUP, Bulletin 23(1), 18–19 (2002)
2. Inaba, R.: Usability of Multilingual Communication Tools. In: Aykin, N. (ed.) HCII 2007. LNCS, vol. 4560, pp. 91–97. Springer, Heidelberg (2007)
3. MeCab, <http://mecab.sourceforge.jp/>
4. KODENSHA, <http://www.kodensha.jp/>
5. Ishida, T.: Language Grid: An Infrastructure for Intercultural Collaboration. In: IEEE/IPSJ Symposium on Applications and the Internet (SAINT 2006), pp. 96–100 (2006)
6. NTT Natural Language Research Group, <http://www.kecl.ntt.co.jp/icl/mtg/resources/index.php>
7. Walker, K., Bamba, M., Miller, D., Ma, X., Cieri, C., Doddington, G.: Multiple-Translation Arabic (MTA) Part 1, Linguistic Data Consortium (LDC) catalog number LDC2003T18 and ISBN 1-58563-276-7

Towards Adapting XCS for Imbalance Problems

Thach Huy Nguyen¹, Sombut Foitong², Phaitoon Srinil³, and Ouen Pinngern⁴

Department of Computer Engineering, Faculty of Engineering,
Research Center for Communication and Information Technology (ReCCIT)
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
{s9060654¹, s8060058², s8060020³, kpouen⁴}@kmitl.ac.th

Abstract. The class imbalance problem has been recognized as a crucial problem in machine learning and data mining. Learning systems tend to be biased towards the majority class and thus have poor performance in classifying the minority class instances. This paper analyzes the imbalance problem in accuracy-based learning classifier system XCS. XCS has shown excellent performance on some data mining tasks, but as other classifiers, it also performs poorly on imbalance data problems. We analyze XCS's behavior on various imbalance levels and propose an appropriate parameter tuning to improve performance of the system. Particularly, XCS is adapted to eliminate *over-general* classifiers¹ and protect accurate classifiers of minority class. Experimental results in Boolean function problems show that, with proposal adaptations, XCS is robust to class imbalance.

Keywords: Classification, Learning Classifier System, XCS, Genetic Algorithm, Imbalance problem.

1 Introduction

Introduced in 1995 by Wilson, accuracy-based learning XCS is one of the most successful learning classifier systems [1]. XCS classifier system has recently shown a high degree of competence on a variety of data mining problems. There are some performance comparisons of strength-based fitness systems and accuracy-based fitness systems showing that accuracy-based fitness systems, including XCS, have a significant impact on data mining field. In [3], Mansilla and Garrel-Guiu studied performances of a set of well-known machine learning algorithms demonstrating that XCS exceeds the performance of most current Machine Learning techniques. In this paper, we study XCS in imbalance data problem.

In classification, data imbalance occurs when the number of patterns of a class is much larger than that of the other class [8]. Many real world data mining applications encounter how to deal with the imbalance problem, such as fraudulent credit card transactions identification, fault detection, target detection oil spill detection or medical diagnosis [8]. In classical machine learning or data mining setting, the classifiers that are designed to optimize overall predictive accuracy tend to produce high predictive

¹ Over-general classifier is a classifier or a rule covering more than one class.

accuracy the majority class but ignore the minority class. This comes from two major causes. The first cause comes from the distribution of the classes. Since the number of majority class patterns exceeds that of the minority class, the majority class is likely to invade the territory the minority class so that the class boundary becomes vulnerable to be distorted. Second, the simple accuracy as an objective function used in most classification tasks is inadequate for the task having data imbalance. From that, there are two classes of methods proposed to solve imbalance problems: at the sampling level and at the classifier level. Methods at the sampling level aim at balancing the a priori probabilities of classes, while methods at the classifier level try to adapt the classifier to class imbalances, e.g. parameter adapting and cost sensitive learning.

This paper studies in the former approach by extending XCS to able to handle imbalance problems. Our study analysis effects of parameters to performance of XCS and provides guidelines to set them appropriately to solve imbalance problems. We restrict our analysis to Boolean classification problems because its characteristics could be controlled carefully rather than on real-world domains whose results would be difficult to decipher. The aim of this study is twofold: first, to make the XCS classifier system more robust on imbalance problems and second, to contribute to the understanding of the functioning of XCS in general to enable further improvement of XCS as well as to understand its limitations.

The rest of paper is organized as follows: XCS classifier system is described briefly in Section 2. In section 3, the proposed parameter adaptation, experimental results and comparison with original XCS will be presented to verify the feasibility of our research. Section 4 provides conclusions and future works.

2 Brief Description of XCS

As other reinforcement learning systems, after XCS sends an action to environment, a reward r is returned, which is then used to update the parameters of classifiers following order [2]: prediction p - an estimate of the payoff for that classifier if its condition matches to the input and its action is selected, prediction error ε - estimates the average error made in the predictions, and finally fitness F - an estimate of the accuracy of the payoff prediction given by p . Prediction p is updated with learning rate as follows:

$$p \leftarrow p + \beta(r - p) \quad (1)$$

where $\beta (0 \leq \beta \leq 1)$ is the learning rate. Next, the prediction error is updated as:

$$\varepsilon \leftarrow \varepsilon + \beta(|r - p| - \varepsilon) \quad (2)$$

Then, the classifier's accuracy is computed as an inverse function of the classifier's prediction error.

In equation (3) and Fig. 1, ε_0 determines the threshold error under which a classifier is considered to be accurate, i.e., it has an accuracy of 1; otherwise, the classifier accuracy κ drops off. $\alpha (0 < \alpha < 1)$ and $v (v > 0)$ control the degree of decline in accuracy if the classifier is inaccurate [1]. Then, the relative accuracy over the action set is computed:

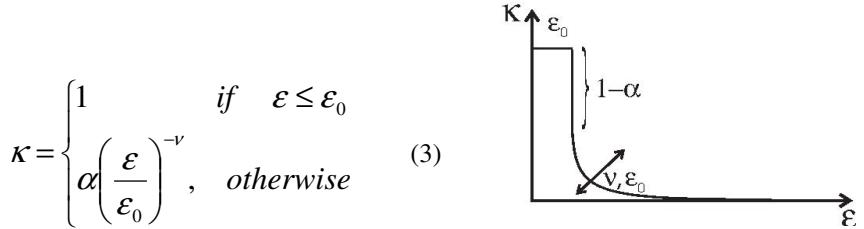


Fig. 1. The scaling of accuracy κ is crucial for successful selection of genetic algorithm

$$\kappa' = \frac{\kappa}{\sum_{cl \in [A]} \kappa_{cl}} \quad (4)$$

where $[A]$ is action set that consists of all the classifiers advocating the action sent to environment. And finally, classifier fitness is updated according to the classifier's relative accuracy:

$$F \leftarrow F + \beta(\kappa' - F) \quad (5)$$

3 Adaptations of XCS for Imbalance Problems

In this section, we analysis effects of imbalance problems and our proposed parameter adaptation to performance of XCS. For this purpose, XCS was applied to version of the well-known multiplexer tasks.

3.1 The Imbalance Multiplexer Dataset Generation

The idea of multiplexer problem is from digital circuit. It is introduced to apply in classifier system by Wilson [1]. There are many versions of the multiplexer problem, each with a different size of the form $k + 2^k$, for $k \geq 1$. For example, in 11-Multiplexer problem, there are 11 inputs and one output (11 is of the form $3 + 2^3$). The first three inputs, a_0 , a_1 , and a_2 , can be considered as address lines. They describe the binary representation of an integer between 0 and 7. This integer chooses one of the 7 remaining inputs, which are labeled d_0 , d_1 , d_2 , d_3 , d_4 , d_5 , d_6 , d_7 . The correct output for the multiplexer is the input on the line specified by the address lines. For instance, the value for the input string 01111110000 is 1.

The imbalance complexity is controlled by means of the probability of sampling an instance of the minority class P_{smin} . At each learning interaction, the environment chooses an instance randomly. If it is a minority class instance, it is passed to XCS with probability P_{smin} . If it is not accepted, a new instance is randomly sampled which undergoes the same decision process. In the remainder of this paper, we use the imbalance ratio ir to denote the ratio between the number of the majority and minority class instances that are given to XCS. Specially, the minority class is sampled with probability $P_{min}=1/(1+ir)$ and the majority class is sampled with probability $P_{maj}=ir/(1+ir)$. Another definition is also used that is imbalance level i where $i=\log_2(ir)$.

3.2 Parameter ε_0 , v and α Dependence in Imbalance Problems

From Fig. 1, we can see that parameters ε_0 , v and α are very important to decide whether a classifier is accurate or not. In imbalance problem, these parameters should be adapted approximately then *over-general* classifiers are considered inaccurate and accurate classifiers are protected. The idea behind adapting these parameters is that in highly imbalanced datasets, an *over-general* classifier will often be correct more often than wrong. Any change in the classifier's condition will be reflected in a change in its statistical correctness – thus its error ε – and this will tend to guide the system toward accurate classifiers.

Parameter adaptation can be approached mathematically. Let us assume a two-level $R_{\max}/0$ payoff landscape, where R_{\max} is provided if the prediction is correct, 0, otherwise. According to [4] and [7], an *over-general* classifier would be considered inaccurate or its accuracy will fall down on the inaccurate part in Fig.1 as long as:

$$\frac{\varepsilon_0}{R_{\max}} * ir^2 + 2 \left(\frac{\varepsilon_0}{R_{\max}} - 1 \right) * ir + \frac{\varepsilon_0}{R_{\max}} \leq 0 \quad (6)$$

Set $t = \varepsilon_0/R_{\max}$, where ε_0 is quite small when compared with R_{\max} , so we only need consider $0 \leq t \leq 1/2$.

From (6), we obtain the boundary of ir respected to changing in ε_0/R_{\max} :

$$\frac{1-t-\sqrt{1-2t}}{t} \leq ir \leq \frac{1-t+\sqrt{1-2t}}{t} \quad (7)$$

Because:

$$\begin{cases} \lim_{t \rightarrow 0^+} \frac{1-t-\sqrt{1-2t}}{t} = \lim_{t \rightarrow 0^+} \frac{t}{1-t+\sqrt{1-2t}} = 0 \\ \lim_{t \rightarrow 0^+} \frac{1-t+\sqrt{1-2t}}{t} = +\infty \end{cases} \quad (8)$$

So, if we decrease the value of $t = \varepsilon_0/R_{\max}$, the boundary of parameter ir increases or XCS can solve problems with higher imbalance levels. Our proposal is to set ε_0/R_{\max} based on imbalance ratio ir , $\varepsilon_0=1/ir$, other parameters are set as with standard XCS.

Additionally, in Fig.1, α causes a strong distinction between accurate and not quite accurate classifiers. The steepness of the succeeding slope is influenced by v , so we should set α low enough and v high enough. Moreover, population size should assure that no niche will be lost as suggested elsewhere [6], it should be increased proportionally with the imbalance rate ir , β and θ_{GA} should set with an appropriate window size as suggested in [7]. First, we run XCS on imbalanced multiplexer problems with imbalance level from $i=0$ to $i=10$ with the standard parameters setting (as introduced in [2]): $N=1000$, $\beta=0.2$, $\alpha=0.1$, $\varepsilon_0=1$, $v=5$, $\theta_{GA}=25$, $\chi=0.8$, $\mu=0.04$, $\theta_{del}=20$, $\delta=0.1$, $\theta_{sub}=200$, $P_\#=0.6$. Second, the result of our proposed method is gotten by setting: $\varepsilon_0=1/ir$.

Table 1 shows that the performance is improved significantly if appropriate ϵ_0 , v and α values are chosen. With standard parameters setting, XCS performs well up to imbalance level $i = 4$, it begins to encounter problem with $i = 5$ and $i = 6$. For higher imbalance levels, from $i = 7$ to $i = 10$, the systems classifies all the input instances as if they belonged to the majority class. In while, adaptive XCS can reach to 100% performance with $i \leq 7$; 95% with $i = 8$; 80% with $i = 9$ which is a notable improvement with respect to the performance of standard XCS. For the highest imbalance level, $i = 10$ or $ir = 2^{10} = 1024$, XCS can only classify correctly about 5% minority instances. Variant values of ϵ_0 , v and α were tested as $v = 10$ and $\alpha = 0.05$ show a better performance speed and stability in the imbalance problems. If we set $v \geq 10$ and $\alpha \leq 0.05$, we can get the results as same as Table 1 (not shown for brevity), but XCS needs a longer time in the beginning steps to evolve a population with accurate classifiers.

Table 1. Performance of Standard XCS and our proposal system in imbalance 11-MUX problem

Imbalance level	Standard XCS	Our proposal
$i = 0$	100%	100%
$i = 1$	100%	100%
$i = 2$	100%	100%
$i = 3$	100%	100%
$i = 4$	100%	100%
$i = 5$	70%	100%
$i = 6$	20%	100%
$i = 7$	0%	100%
$i = 8$	0%	95%
$i = 9$	0%	80%
$i = 10$	0%	5%

4 Conclusions

We have introduced an XCS with adapting parameters to solve imbalance problems. We first analyzed effects of imbalance multiplexer problems to performances of XCS. The results show that with standard parameter settings, XCS is quite robust to imbalance problems up to imbalance ratio $ir = 16$ ($i = 4$) of multiplexer problem. For higher imbalance levels, XCS's parameters are needed to adapt to improve its performance. The parameter setting that we proposed makes it possible for XCS to address imbalance problems. It attempts to eliminate *over-general* classifiers and protect accuracy classifiers of minority class, so the performance of XCS is improved. The value of ϵ_0 guides the selection pressure within genetic algorithm since it represents the steepness of the fitness curve. Therefore, choosing the correct value of ϵ_0 gives the system the ability to create the required amount of pressure that can solve the problems used by [4] including the imbalanced ones. It also controls the GA's ability to cope with the low values for other parameters.

As future works, we would like to study effects of injecting various degrees and types of noises to XCS's performance. Another important consideration has to do that is studying sampling techniques [9] in conjunction with XCS to enhance quality

performance. A real world problem that we are investigating is intrusion detection where the problem is to realize decision boundaries between attacks and normal activities and highly imbalanced attack class distribution [10]. With the on-line learning probability, XCS is shown as a potential tool to solve intrusion detection.

References

1. Wilson, S.W.: Classifier Fitness Based on Accuracy. *Evolutionary Computation* 3(2), 149–175 (1995)
2. Butz, M., Wilson, S.: An algorithmic description of XCS. *Journal of Soft Computing* 6, 144–153 (2002)
3. Bernadó-Mansilla, E., Garrell Guiu, J.M.: Accuracy-Based Learning Classifier Systems: Models, Analysis and Applications to Classification Tasks. *Evolutionary Computation* 11 3, 209–238 (2003)
4. Butz, M., Kovacs, T., Lanzi, P.L., Stewart, W.: Toward a theory of generalization and learning in XCS. *IEEE Trans. Evolutionary Computation* 8(1), 28–46 (2004)
5. Kovacs, T.: XCS classifier system reliably evolves accurate, complete, and minimal representations for boolean functions. In: *Soft Computing in Engineering Design and Manufacturing*, London, U.K, pp. 59–68. Springer, Heidelberg (1997)
6. Orriols-Puig, A., Goldberg, D.E., Sastry, K., Bernadó-Mansilla, E.: Modeling XCS in Class Imbalances: Population Size and Parameter Settings. Technical report, IliGAI Report No. 2007001, USA (February 2007)
7. Orriols-Puig, A., Bernadó-Mansilla, E.: The class imbalance problem in learning classifier systems: a preliminary study. In: *GECCO 2005*, Washington, D.C, USA, pp. 74–78 (2005)
8. Chris, S., Taghi, M.K., Jason, V.H., Andres, F.: An Empirical Study of the Classification Performance of Learners on Imbalanced and Noisy Software Quality Data. In: *Information Reuse and Integration*, IEEE International Conference, pp. 651–658 (August 2007)
9. Nitesh, V.C., Kevin, W.B., Lawrence, O.H., Philip, W.K.: SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16, 321–357 (2002)
10. Lee, W., Stolfo, S., Mok, K.: A data mining framework for building intrusion detection models. In: *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, Oakland, CA, pp. 120–132 (May 1999)

Personalized Summarization Agent Using Non-negative Matrix Factorization

Sun Park

Department of Computer Engineering, Honam University, Gwangju, Korea
sunpark@honam.ac.kr

Abstract. This paper proposes personalized summarization agent using non-negative matrix factorization (NMF) to extract sentences relevant to a user interesting for a generic and query based summary. The proposed agent uses NMF to summarize generic summary so that it can extract sentences covering the major topics of the search results with respect to user interesting. Besides, it can improve the quality of query based summaries because the inherent semantics of the search results are well reflected by using NMF and the sentences most relevant to the given query. The experimental results demonstrate that the proposed method achieves better performance the other methods.

1 Introduction

The automatic summarization is the process of reducing the sizes of documents while maintaining their basic outlines. That is, it should distill the most important information from the document. It can be either generic summaries or query based (or topic based) summaries [6]. A generic summary [2, 9] distills an overall sense of the documents' contents whereas a query based [6, 7, 8] summary only distills the contents of the document relevant to the user's query.

With the fast growth of the Internet access by personal user, it has increased the necessity of the personalized information seeking and personalized summaries. If the summary is personalized according to user interests, the user can save time not only in deciding whether it is interesting or not, but also in finding the information without having to read the full text. The personalized summarization is the process of summarization that preserves the specific information that is relevant for a given user profile rather than information that truly summarizes the content of the search results [1]. To build a personalized or user-adapted summary a representation of the interests of the corresponding user is studied [1, 10, 12].

In this paper, we propose a personalized summarization agent using a generic and query based summary by NMF to summarize a personalized summarization with regard to a given query. The NMF can find a parts representation of the data because non-negative constraints of the NMF are compatible with the intuitive notions of combining parts to form a whole, which is how the NMF learns a parts-based representation [4, 5]. The non-negative semantic feature matrix (NSFM) and the non-negative semantic variable matrix (NSVM) are calculated from NMF.

The propose method has the following advantages: First, it can select sentences covering the major topics of the search results with respect to user interesting by using NMF and relevance score with respect to user query. Second, it can improve the quality of summarization since extracting sentences to reflect the inherent semantics of the search results by using NMF.

2 Non-negative Matrix Factorization

In this paper, we define the matrix notation as follows: Let X_{*j} be j 'th column vector of matrix X , X_{i*} be i 'th row vector, and X_{ij} be the element of i 'th row and j 'th column.

Non-negative matrix factorization (NMF) is to decompose a given $m \times n$ matrix A into a non-negative semantic feature matrix W and a non-negative semantic variable matrix H as shown in Equation (1).

$$A \approx WH \quad (1)$$

Where W is $m \times r$ non-negative matrix, H is $r \times n$ non-negative matrix, r is the number of semantic feature vectors. Usually r is chosen to be smaller than m or n , so that the total sizes of W and H are smaller than that of the original matrix.

The update rules for NMF are as follows [4, 5]:

$$H_{\alpha\mu} \leftarrow H_{\alpha\mu} \frac{(W^T A)_{\alpha\mu}}{(W^T W H)_{\alpha\mu}}, W_{i\alpha} \leftarrow W_{i\alpha} \frac{(A H^T)_{i\alpha}}{(W H H^T)_{i\alpha}} \quad (2)$$

3 Personalized Summarization Agent

In this paper, we propose a personalized summarization agent using NMF. The proposed agent consists of the preprocessing phase, the sentence ranking phase, the summary generation phase.

3.1 Preprocessing

In the preprocessing phase, after given search results are decomposed into individual sentences, we remove stop-words and perform words stemming [3]. Then we construct the weighted term-frequency vector for each sentence in search results using Equation (3) [11]. Let A be $m \times n$ matrix, where m is the number of terms and n is the number of sentences in the whole search results. Let element A_{ji} be the weighted term-frequency of term j in sentences i .

$$A_{ji} = L_{ji} \log(N/N(j)) \quad (3)$$

Where L_{ji} is the local weight (term frequency) for term j in the sentence i , N is the total number of sentences in the whole search results, and $N(j)$ is the number of sentences that contain term j [11].

3.2 Sentence Ranking

The sentence ranking procedure consists of the generic phase, the query based phase.

In the generic phase, we construct the candidate sentence set and calculate the relevance score for summary generation. We modify our previous generic summarization method using non-negative semantic variable by NMF for constructing the candidate sentence set [9].

We define the generic weight of a sentence $gweight()$ as follows:

$$gweight(H_{j^*}) = \sum_{q=1}^n H_{jq} \quad (4)$$

The $gweight$ (H_{j^*}) means the relative relevance of j 'th semantic feature (W_{*l}) among all semantic features. It also means how much the sentence reflects major topic which is represented as semantic features.

We compute the relevance score of each selected sentence with a user query by Equation (5). The relevance score means how much the selected sentence reflects user query which are represented as the semantic weight of a sentence.

$$r_i = gweight(H_{j^*}) \times \vec{sim}(q, A_{*i}) \quad (5)$$

Where r_i is a relevance score of i 'th sentence, $sim()$ is a cosine similarity function, \vec{q} is a query for user interesting, A_{*i} is a i 'th sentence.

The cosine similarity function between the vector of a 'th sentence A_{*a} and the vector of b 'th sentence A_{*b} is computed as follows [11].

$$sim(A_{*a}, A_{*b}) = \frac{A_{*a} \cdot A_b}{|A_{*a}| \times |A_{*b}|} \quad (6)$$

In the query based phase, we construct the candidate sentence set and calculate the query score for summary generation. We modify our previous query based document summarization method [7, 8] using NMF for constructing the candidate sentence set. The query based phase is described as follows. To evaluate the degree of similarity of the semantic feature vector W_{*l} with regard to the query \vec{q} as the correlation between the vector W_{*l} and \vec{q} used Equation (6). To select those sentences of the search results that is most relevant to a given query and reflecting major topic in search results, query score t , is defined as follows.

$$t_i = sim(W_{*l}, q) \times \vec{gweight}(H_{l^*}) \quad (7)$$

Where t_i is a query score of i 'th sentence, k is the number of extract sentences, and $gweight()$ is a generic relevance weight.

3.3 Summary Generation

Summary generation phase extracts top k ranked sentences with respect to user interesting from the candidate sentences set for generic or query based summaries. This phase is described as follows: We normalize the relevance scores r and the query scores t . We then calculate the ranking score of the candidate sentences by using Equation (8).

$$rs_i = \alpha \cdot r_i + \beta \cdot t_i \quad (8)$$

Where rs_i is a ranking score of i 'th sentence, parameters α and β is adjustment between the generic summary and the query based summary.

4 Experimental Results

As an experimental data, we used Yahoo-Korea News¹. We gave a query to retrieve news from Yahoo-Korea News. Three independent evaluators were employed to manually create summarization on the 200 news from the retrieved Yahoo Korea news with respect to 10 queries.

In this paper, we used the recall (R), precision (P), and F -measure to evaluate the performance of the proposed method. Let S_{man} , S_{sum} be the set of sentences selected by the human evaluators, and the summarizer, respectively. The standard definitions of recall (R), precision (P), and F -measure is defined as follows [11]:

$$R = \frac{|S_{man} \cap S_{sum}|}{|S_{man}|}, \quad P = \frac{|S_{man} \cap S_{sum}|}{|S_{sum}|}, \quad F = \frac{2RP}{R+P} \quad (9)$$

Experiment 1. We evaluate 3 different generic summarization methods such as the RM, the LSA, and the GNMF. The RM denotes Gong's method using Relevance Measure [2]. The LSA denotes Gong's method using Latent Semantic Analysis [2]. The GNMF denotes the proposed generic method. Let β be zero. The average recall of GNMF is approximately 26.3% higher than that of RM, 10.5% higher than that of LSA. The average precision of GNMF is approximately 22.9% higher than that of RM, 20.0% higher than that of LSA. The average F -measure of GNMF is approximately 10.5% higher than that of RM, 5.3% higher than that of LSA. Experimental results show that the proposed method surpasses RM and LSA. Our generic method uses NMF and relevance score with respect to user interesting to find the semantic feature with non-negative values for meaningful summarization [8, 9].

Experiment 2. We evaluated 3 different summarization methods such as the UPS, the PSFV, and the TNMF. The UPS denotes Diaz's method [1] using user-model based personalized summarization. The PSFV also denotes our previous method using NMF for personalized summary [10]. The TNMF denotes the proposed query based method. Let α be zero. The average recall of TNMF is approximately 15.8% higher than that of UPS, 4.4% higher than that of PSFV. The average precision of TNMF is approximately 18.0% higher than that of UPS, 7.6% higher than that of PSFV. The average F -measure of TNMF is approximately 20.0% higher than that of UPS, 7.9% higher than that of PSFV. The result shows that recall, precision, and F -measure of

¹ <http://kr.news.yahoo.com> (2008)

PSFV are better than those of the UPS because the PSFV generates more meaningful summary by reflecting the inherent semantics of the search results with respect to generic summary. The TNMF shows best performance. The proposed method generates meaningful personalized summary by means of the semantic feature and semantic variable reflecting the inherent structure in the search results for user interesting.

5 Conclusion

In this paper, we propose the automatic personalized text summarization agent based on generic and query based summarization method using NMF. The proposed method can select sentences covering the major topics of the search results with respect to user interesting. Besides, it can improve the quality of summarization since extracting sentences to reflect the inherent semantics of the search results with respect to a given query. It can select sentences that are highly relevant to a user interesting because it can choose the sentences related to the user's query relevant semantic features that well represent the structure of the search results. Experimental results show that the proposed method outperforms other generic and query based summarization methods.

References

1. Diaz, A., Gervas, P.: User-model based personalized summarization. *Information Processing and Management* 43, 1715–1734 (2007)
2. Gong, Y., Liu, X.: Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In: Proceeding of ACM SIGIR, pp. 19–25 (2007)
3. Kang, S.S.: *Information Retrieval and Morpheme Analysis*. HongReung Science Publishing Company (2002)
4. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 788–791 (1999)
5. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562 (2000)
6. Mani, I.: *Automatic Summarization*. John Benjamins Publishing Company, Amsterdam (2001)
7. Park, S., Lee, J.H., Ahn, C.M., Hong, J.S., Chun, S.J.: Query Based Summarization using Non-negative Matrix Factorization. In: Proceeding of the International Conference on Knowledge-Based & Intelligent Information & Engineering Systems, pp. 84–87 (2006)
8. Park, S., Lee, J.H.: Topic-based Multi-document Summarization Using Non-negative Matrix Factorization and K-means. *Journal of KIISE: Software and Applications* 35(4), 255–264 (2008)
9. Park, S.: Generic Summarization using Non-negative Semantic Variable. In: Huang, D.-S., Wunsch II, D.C., Levine, D.S., Jo, K.-H. (eds.) *ICIC 2008. LNCS*, vol. 5226, pp. 1025–1031. Springer, Heidelberg (2008)
10. Park, S.: Personalized Document Summarization using Non-negative Semantic Feature and Non-negative Semantic Variable. In: Proceeding of the 9th International Conference on Intelligent Data Engineering and Automated Learning (2008)
11. Ricardo, B.Y., Berthier, R.N.: *Moden Information Retrieval*. ACM Press, New York (1999)
12. Sanderson, M.: Accurate user directed summarization from existing tools. In: Proceeding of the international conference on information and knowledge management, pp. 45–51 (1998)

Interactive Knowledge Acquisition and Scenario Authoring

Debbie Richards

Computing Department, Macquarie University
North Ryde, Australia, 2109
richards@ics.mq.edu.au

Abstract. Encoding knowledge and authoring content in virtual reality systems typically requires trained specialists. This project, concerned with training people to identify risky situations, aims to reduce the knowledge acquisition bottleneck by allowing trainers to enter their knowledge and generate alternative scenarios whilst interacting with existing training sessions. The goal is not only to speed up the capture of knowledge and content but also to tap into (tacit) knowledge through this knowledge-in-action approach and to allow the trainee to experience and query the underlying domain knowledge. Ripple Down Rules is proposed as the knowledge representation and method as it supports incremental acquisition of rules performed by the domain expert as new situations arise. In this project we seek to capture concurrently new knowledge and cases.

Keywords: knowledge based systems, ripple down rules, virtual reality, training simulation, narrative intelligence, adaptive/intelligent user interfaces.

1 Introduction

Acquiring knowledge has been a bottleneck in the development of knowledge based systems (KBS) [11]. Similar issues are faced in the authoring of drama or narrative-based systems (NBS) [1]. These issues include the need for intensive effort and specialist involvement and the need to find a representation that is sufficiently expressive. On the KBS side, further major issues concern validation and maintenance. On the NBS side, control and flow are key challenges. In our current project involving the development of an immersive training system we need to manage the relevant knowledge and the storyline. Thus we seek a solution which allows us to acquire, maintain and validate knowledge in the context of different training scenarios that supports flexible and varied interactions between the user and system. The focus is on experiential learning to potentially achieve deep learning that will assist the trainee to handle potential threats in real situations. The particular domain in which our studies are being conducted is the training of customs/immigrations officers at airports.

To allow the domain expert to directly enter their knowledge into the system and to motivate and contextualize the knowledge capture event, we use a technique known as Ripple Down Rules (RDR) [3] which captures knowledge (in the form of production rules) associated with cases which naturally occur in the domain. The cases provide the context in which the rule/s apply and the rules provide the indexes to the

cases. The type of knowledge captured is knowledge-in-action, which is a combination of codified/explicit knowledge and tacit/practical knowledge.

In the following sections, we describe ripple down rules (section 2) and our use of RDR to acquire knowledge while interacting with a scenario (section 3). Finally, we present related work and conclusions.

2 MCRDR (Multiple Classification Ripple Down Rules)

Kang [7] developed the MCRDR algorithm. MCRDR overcomes a major limitation in Ripple Down Rules (RDR), which only permitted single classification of a set of data. That is MCRDR allows multiple independent classifications. An MCRDR knowledge base is represented by an n-ary tree [7]. The tree consists of a set of exception production rules in the form “If Condition Then Conclusion”.

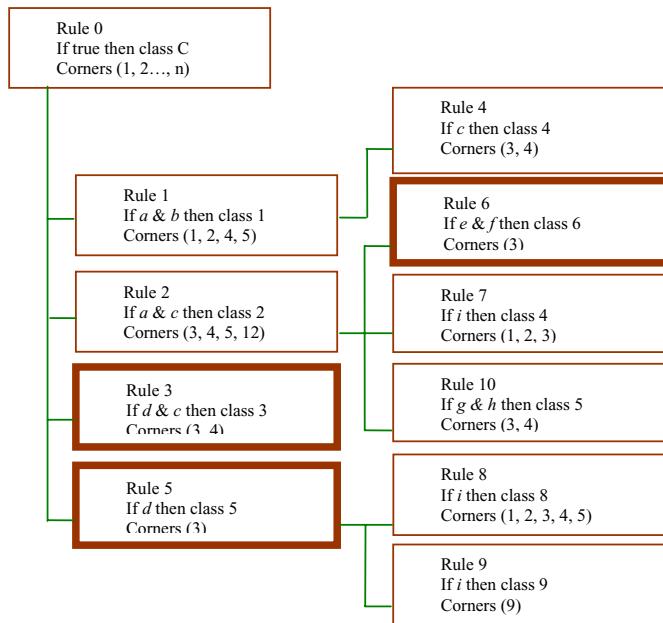


Fig. 1. MCRDR Knowledge Base and inference results for the case {a, c, d, e, f, h, k}

The inference process of MCRDR allows for multiple independent conclusions with the validation and verification of multiple paths [7]. This can be achieved by validating the children of all rules which evaluate to true. An example of the MCRDR inference process is illustrated in Figure 1. In this example, a case has attributes {a, c, d, e, f, h, k} and three classifications (conclusion 3, 5 and 6) are produced by the inference. Rule 1 does not fire. Rule 2 is validated as true as both ‘a’ and ‘c’ are found in our case. Now we should consider the children (rules 6, 7, and 10) of rule 2. From

comparison of the conditions in children rules with our case attributes, only rule 6 is evaluated as true. Hence, rule 6 would fire to get a conclusion 6 which is our case classification. This process is applied to the complete MCRDR rule structure in Figure 3. As a result, rule 3 and 5 can also fire, so that conclusion 3 and conclusion 5 are also our case classifications.

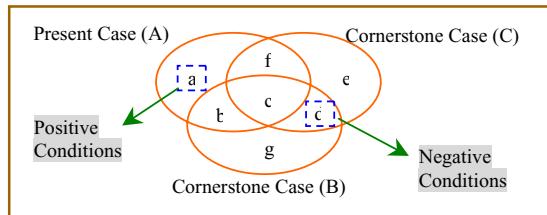


Fig. 2. Difference list {a, not d} distinguishes the Present Case (A) from two Cornerstones Cases (B) and (C)

This technique is based on a situated view of knowledge where knowledge is “made-up” to fit the particular context. In RDR knowledge is patched by adding new knowledge which non-monotonically overrides previous knowledge. Knowledge is patched in the local context of a rule that misfires producing decision lists of exceptions. Context is provided by cases. In our training simulation the current state of the world will be treated as the current case.

When a new situation/case arises there is an opportunity for new knowledge to be acquired. We consider a new case (present case) A and two cornerstone cases B and cornerstone cases C, see Figure 2. The cornerstone case is the case that prompted the rule being modified to be originally added. That is, the present case has been run and a rule has fired but the domain expert does not agree with the conclusion given. There must be some features in the present case which are different to the cornerstone case which merit a different conclusion. The present case will become the cornerstone case for the new (exception) rule. To generate conditions for the new rule, the system has to look up the cornerstone cases in the parent rule. When a case is misclassified, the rule giving the wrong conclusion must be modified. The system will add an exception rule at this location and use the cornerstone cases in the parent rule to determine what is different between the previously seen cases and the present case. These differences will form the rule condition and may include positive and negative conditions (see Formula (1)).

Positive Condition:

$$\text{Present Case (A)} - (\text{Cornerstone Case (B)} \cup \text{Cornerstone Case (C)}) \quad (1)$$

Negative Condition:

$$(\text{Cornerstone Case (B)} \cap \text{Cornerstone Case (C)}) - \text{Present Case (A)}$$

3 MCRDR and Interactive Authoring

In accordance with the RDR approach, knowledge acquisition and maintenance will occur in response to a domain expert executing the knowledge, that is, performing an inference over the rule base for a given a case. In our project the context is running a training scenario. The type of knowledge to be captured during training will be exception knowledge, knowledge of the user (their actions and preferences) and the kinds of risks being encountered. To enter knowledge or make changes to the environment (e.g. existence of a piece of furniture, level of lighting, tone of voice) the domain expert will be able to interrupt the game and add a rule which captures the current situation, and allows a rule to be added which then changes something in the current situation. When a session is paused the current state of the world will be displayed. In figure 3 we see a “case” popup over the Vizard window which contains the salient features pertaining to the current state/case. At this point, the trainer who is playing the dual roles of domain expert and scenario author, is able to change the world and thereby create a new variation to the existing scenario. For example, the trainer may select a different character, this will result in a change to the personal details (as they are attached to each avatar). While age, gender and race can not be individually changed as they are key elements of the character, it will be possible to change the clothing they are wearing by applying a different skin. Other features that can change include: luggage; the purpose of the visit; nationality; employment status; and criminal history. Additional features could be added as deemed necessary for the domain. The trainer may the save the changes and the scenario can continue with the new changes integrated into the scenarios. This poses some challenges that we are still working on.

Beyond simply creating alternative scenarios we want to use these interruptions/episodes to capture knowledge under the assumption that at least in some cases the

trainer wants to change the scenario so that something different can be experienced and some new knowledge be learnt by the trainee. Thus when a new case is created, the trainer is asked “why” they made the change. The answer to this question becomes the new knowledge. For example, if the character is



Fig. 3. Vizard Screen Shot of Airport World showing case popup overlaying the scenario

changed from a middle-aged man to a 25 year old male it may be because the latter is more likely to be guilty of some illegal behaviour. This could be represented, for example, in an exception rule {if age=25-35 then risk=risk*2} which overrides the rule {if gender=male then risk=3}.

The use of RDR for risk detection and manage a story line is novel. Further, the need to integrate and update the new knowledge with the scenarios is also novel and further research is needed to determine what modifications are needed to the standard RDR algorithm and method of knowledge acquisition. However, from our experience with RDR for other purposes (e.g. pathology report interpretation [4], help desk [8]), domain experts find the acquisition of knowledge through the incremental comparison of a current situation (case) with a previous situation as it arises more manageable than having to specify upfront global rules and anticipate the range of situations the system will need to deal with. Particularly when we are dealing with experiential and tacit knowledge, we think that it is only via interaction and a sense of immersion that this knowledge can come to the surface.

While at this stage from the trainee's point of view the system will primarily comprise scenarios to experience, the trainee will also be able to interrupt the scenario to test their own knowledge and also to access the underlying knowledge. Using the knowledge in the RDR KBS, the trainee will be able at any point in the scenario to query the system regarding the knowledge related to the scenario on hand. Similar to the trainer's interface, the trainee can interrupt the system and run the current state of the scenario to see what conclusions are given. As an alternative to consultation mode, the trainee can select critiquing mode and enter what they believe are the appropriate actions and conclusions, such as the degree of risk. Also, as a rule based system, the trainee will be able to ask for an explanation of why a certain recommendation (e.g check the passenger's bag, frisk the passenger, ask where they are staying) was made or conclusion reached (e.g. risk level is low, passenger is highly suspicious, passport is valid). The system will be able to return a rule trace to allow the trainee to follow the expert's line of reasoning.

4 Related Work and Conclusions

This work is multidisciplinary and multi-faceted. On the one hand we are concerned with a particular application domain and problem, which could in general be seen as the identification of risk or more specifically the detection of suspicious or deceptive behaviour. Much of the research in this area is conducted by psychologists, such as the work of well-known psychologist Paul Ekmann [6]. Ekmann currently plays a leading role in training customs/immigrations officers in the US to detect when passengers are lying. Also relevant are the finding of DePaulo et al. [5] who have gathered together the results of more than 1,300 estimates of 158 cues to deception from 120 independent samples. We are encoding many of the identified behaviours into our knowledge base, dialogues and behaviours. Work concerning deception and also using virtual environments has been done by Rehm and André [12]. Similar to the work of Pelachaud et al. [9] and Prendinger et al. [10], the work by [12] concerns

displaying and detecting deception via facial expression. Our focus currently is limited to language, dialogues and gross actions rather than fine actions such as body gestures and facial expressions. Trust related to agent based systems has been explored by many. Specifically related to deception and lying is the work of Castelfranchi and Poggi [2]. This body of work is valuable to us particularly in building up a body of knowledge to be encoded in our RDR KBS. None of these approaches focus on how to acquire knowledge or author scenarios interactively, which are the major goals of the work described in this paper.

The closest application to our problem domain using RDR is the work conducted by Wang et al. [13] for the Australian Health Insurance Commission to identify general practitioners who are involved in fraudulent practice. Single classification RDR was used to classify 1,500 practice profiles as fraudulent or not fraudulent. Through this incremental classification process the rules were simultaneously acquired and the results compared against manual assignment of classifications by humans. Our work differs not only by using MCRDR but the interactive acquisition of knowledge as a narrative unfolds makes the challenges and issues quite different. We are currently working on implementation of the concepts. It is not yet clear if any modifications may be needed to standard RDR.

Acknowledgements. This project is sponsored by the Australian Research Council Discovery Grant (DP0558852). Thanks to Meredith Taylor and John Porte for their contribution.

References

1. Brisson, A., Dias, J., Paiva, A.: From Chinese shadows to interactive shadows: building a storytelling application with autonomous shadows. In: Proceedings of the Workshop on Agent-Based Systems for Human Learning and Entertainment (ABSHLE), AAMAS 2007. ACM Press, New York (2007)
2. Castelfranchi, C., Poggi, I.: Lying as pretending to give information. In: Parret, H. (ed.) *Pretending to communicate*, de Gruyter, Berlin, New York, pp. 276–291 (1993)
3. Compton, P., Jansen, R.: A Philosophical Basis for Knowledge Acquisition. *Knowledge Acquisition* 2, 241–257 (1990)
4. Compton, P., Peters, L., Edwards, G., Lavers, T.G.: Experience with Ripple-Down Rules. *Knowledge-Based Systems* 19(5), 356–362 (2006)
5. DePaulo, B.M., Lindsay, J.J., Malone, B.E., Muhlenbruck, L., Charlton, K., Cooper, H.: Cues to deception. *Psychological Bulletin* 129, 74–118 (2003)
6. Ekman, P.: *Telling Lies — Clues to Deceit in the Marketplace, Politics, and Marriage*, 3rd edn. Norton and Co. Ltd., New York (1992)
7. Kang, B.H.: *Validating Knowledge Acquisition: Multiple Classification Ripple Down Rules*, PhD dissertation, Computer Science, University of New South Wales (1995)
8. Kang, B., Yoshida, K., Motoda, H., Compton, P.: A help desk system with intelligence interface. *Applied Artificial Intelligence* 11, 611–631 (1997)
9. Pelachaud, C., Poggi, I.: Subtleties of facial expressions in embodied agents. *Journal of Visualization and Computer Animation* 31, 301–312 (2002)

10. Prendinger, H., Ishizuka, M.: Social Role Awareness in Animated Agents. In: Proceedings of Agents 2001, Montreal, Canada, pp. 270–277 (2001)
11. Quinlan, J.R.: Discovering rules by induction from large collections of examples. In: Mitchie, D.E. (ed.) Expert systems in the micro-electronic age. Edinburgh Uni. Press, Edinburgh (1979)
12. Rehm, M., André, E.: Catch me if you can: exploring lying agents in social settings. In: Proceedings of the Fourth international Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2005, The Netherlands, pp. 937–944. ACM, New York (2005)
13. Wang, J.C., Boland, M., Graco, W., He, H.: Classifying general practitioner practice profiles. In: Compton, P., Mizoguchi, R., Motoda, H., Menzies, T. (eds.) PKAW 1996: Pacific Knowledge Acquisition Workshop, Coogee, Sydney, Australia, pp. 333–345 (October 1996)

Reconstructing Hard Problems in a Human-Readable and Machine-Processable Way

Rolf Schwitter

Centre for Language Technology
Macquarie University
Sydney NSW 2109, Australia
schwitt@ics.mq.edu.au

Abstract. This paper shows how a controlled natural language can help to reconstruct a logic puzzle in a well-defined subset of natural language and discusses how this puzzle can then be processed and solved using a state of the art model generator. Our approach relies on a collaboration between humans and machines and bridges the gap between a (seemingly informal) problem description and an executable formal specification.

1 Introduction

Recent work in Language Technology has mainly focused on broad-coverage language processing techniques using statistical methods [2]. Hard questions about the role of structural semantics in combination with conceptual knowledge and inference for high-quality natural language understanding have been largely ignored. For example, most methods that are used in the Pascal Recognizing Text Entailment (RTE) challenge are rather shallow than deep, apart from a few exceptions that are doing quite well [3]. Without doubt many NLP applications could benefit from more complete and precise understanding of texts. Solving logic puzzles is an interesting target domain for research in structural semantics and automated reasoning since the solutions must be logically inferred and this requires a formal representation of the problem descriptions. In contrast to the RTE challenge, logic puzzles consist of more than two or three sentences, are clearly situated in a specific context, and have a clear evaluation metric since there is no disagreement about their correct solution [4]. The transition of a puzzle's informal problem description into its formal specification is rarely straightforward and requires very often major reconstructions of the original text on a case by case basis and the addition of inference-supporting background knowledge. For many people, authoring knowledge in formal logic is difficult, and I believe that providing a high-level interface language is of benefit for this user group. The interface language that I promote is a machine-oriented controlled natural language¹ that offers a number of attractive features: the language looks like English and is easy to understand by humans and easy to process by a machine; furthermore, the language is precisely defined so that it can be translated unambiguously into first-order logic and thus is in fact a formal language.

¹ See <http://www.ics.mq.edu.au/~rolfs/controlled-natural-languages>

2 Einstein's Riddle

Einstein's Riddle is a logic puzzle which is said to have been invented by Albert Einstein when he was a boy in the late 1800s. Some sources claim that Einstein said that only 2 percent of the world's population can solve this puzzle. There exist several versions of the puzzle which is also known as the Zebra Puzzle but there is no hard evidence for Einstein's authorship – nevertheless the puzzle is a nice one. The version below is quoted from the first known publication [5]:

1. There are five houses.
2. The Englishman lives in the red house.
3. The Spaniard owns the dog.
4. Coffee is drunk in the green house.
5. The Ukrainian drinks tea.
6. The green house is immediately to the right of the ivory house.
7. The Old Gold smoker owns snails.
8. Kools are smoked in the yellow house.
9. Milk is drunk in the middle house.
10. The Norwegian lives in the first house.
11. The man who smokes Chesterfields lives in the house next to the man with the fox.
12. Kools are smoked in the house next to the house where the horse is kept.
13. The Lucky Strike smoker drinks orange juice.
14. The Japanese smokes Parliaments.
15. The Norwegian lives next to the blue house.

Now, who drinks water? Who owns the zebra?

In the interest of clarity, it must be added that each of the five houses is painted a different color, and their inhabitants are of different national extractions, own different pets, drink different beverages and smoke different brands of American cigarettes. One other thing: In Statement 6, *right* means *your right*.

There are many **formal** solutions to this puzzle available in popular programming languages (Prolog, Lisp, C/C++) and the puzzle has also been used as a benchmark for testing automated theorem provers [7]. It is usually straightforward for a computer to solve a formalised problem description of a puzzle but it is usually hard for a human to correctly formalise a puzzle, and it is even harder for a machine to take the original natural language description as input and generate the formal representation automatically. To the best of my knowledge there exists **no language processing and reasoning tool** that takes Einstein's original text as input and produces the correct solution.

3 Logical Reconstruction of the Puzzle

Let us try to reconstruct the puzzle using a controlled natural language that supports the writing of production rules, and let us further assume that the writing process is backed up by a **predictive authoring tool** [6] that enforces the constraints of the controlled natural language. For this reconstruction process, we use the following simple grammar rules as guidelines:

$S_2 \rightarrow [if], S_1, [then], S_1.$	$N_1 \rightarrow Adj, Noun \mid Noun.$
$S_2 \rightarrow [it,is,false,that], S_1.$	$VP \rightarrow V1, Conj, V1.$
$S_1 \rightarrow S, Conj, S.$	$VP \rightarrow V1.$
$S_1 \rightarrow S.$	$V1 \rightarrow TV, NP.$
$S \rightarrow NP, VP.$	$V1 \rightarrow Cop, RAdj, NP.$
$S \rightarrow [there,is], NP.$	$V1 \rightarrow Cop, [not], RAdj, NP.$
$NP \rightarrow Spec, N1, \{NVar\}.$	$V1 \rightarrow Cop, Adj.$
$NP \rightarrow MNoun \mid PN \mid NVar \mid Adj.$	$Conj \rightarrow [and] \mid [or].$

This gives us just the necessary power to reconstruct the puzzle in a way that makes the logical content of the puzzle explicit and machine-processable. Of course, current controlled natural languages have many more grammar rules but this set of rules is sufficient for our purpose.

3.1 Specifying the Background Information

People who solve this puzzle by hand apply automatically and unconsciously a surprisingly large amount of background knowledge. We explicitly specify this background knowledge by rules that convey the intended meaning. We structure this information around the two classes *person* and *house* and describe the relevant activities and properties with the help of five conditional sentences:

- 1b. If there is a person then the person lives in the first house or lives in the second house or lives in the third house or lives in the fourth house or lives in the fifth house.
- 2b. If there is a person then the person owns the dog or owns the snail or owns the horse or owns the fox or owns the zebra.
- 3b. If there is a person then the person drinks orange juice or drinks coffee or drinks tea or drinks milk or drinks water.
- 4b. If there is a person then the person smokes Kools or smokes Chester-fields or smokes Old Gold or smokes Lucky Strike or smokes Parliaments.
- 5b. If there is a house then the house is red or is yellow or is blue or is green or is ivory.

In the next step, we make sure that all elements are distinct since it is, for example, not possible that the Englishman and the Spaniard live in the same house or that the first house and the third house have the same colour. We can express these uniqueness conditions as follows in controlled natural language:

- 6b. It is false that a person X1 lives in a house and a person X2 lives in that house and X1 is not equal to X2.

This can be done in a similar way for [7b.-10b.] with *person/owns/animal*, *person/drinks/beverage*, *person/smokes/brand*, and *house/has/colour*.

So far, we did not specify any class axioms or declare any class membership. That means the theorem prover would, for example, not recognise those classes that are subsumed by the superclass *person* or those individuals that belong to the class *brand*. Subclass relationships such as 11b, 12b and 13b provide the necessary class axioms and statements such as 14b and 15b express class membership of specific individuals:

- 11b. Every Englishman is a person.
- 12b. Every dog is an animal.
- 13b. Tea is a beverage.
- 14b. Kools is a brand.
- 15b. Red is a colour.

This kind of ontological information is sometimes available in a linguistic resource but there is no guarantee that this resource is complete for any application, therefore the user needs to be able to add this background information – if necessary. The original problem description uses the relations *next to* (see 11, 12 and 15) and *right of* (see 6) but does not specify that the houses are in a row. This information is missing and needs to be added: the relation *next to* is symmetric and can be defined with the help of the *right of* relation:

- 16b. If X is right of Y then X is next to Y.
- 17b. If Y is right of X then X is next to Y.
- 18b. If X is next to Y then X is right of Y or Y is right of X.

In principle, it does not matter whether the houses are counted from left to right or from right to left, what is important is the order but not the direction. Therefore, we assume that the fifth house is the rightmost one and specify:

- 19b. The fifth house is right of the fourth house.
- 20b. The fourth house is right of the third house.
- 21b. The third house is right of the second house.
- 22b. The second house is right of the first house.

As we can see, an automatic solution of the puzzle needs a substantial amount of nontrivial background information (1b–22b) that is not available in textual form from the original problem description.

3.2 Specifying the Premises

Once we have specified the required background information, we can consider the premises of the original puzzle in more detail and start with those sentences which provide factual information (see 3, 5, 10 and 14). These sentences do not require any reconstruction and are simply repeated here:

- 3. The Spaniard owns the dog.
- 5. The Ukrainian drinks tea.
- 10. The Norwegian lives in the first house.
- 14. The Japanese smokes Parliaments.

Now, we take all pair wise clues of the original puzzle into consideration (see 2, 4, 7, 8, 9 and 13) and reconstruct them as conditional sentences since they are in essence rules:

- 2r. If the Englishman lives in a house then the house is red.
- 4r. If a person drinks coffee and lives in a house then the house is green.

- 7r. If a person smokes Old Gold then the person owns a snail.
- 8r. If a person smokes Kools and the person lives in a house then the house is yellow.
- 9r. If a person drinks milk then the person lives in the third house.
- 13r. If a person smokes Lucky Strike then the person drinks orange juice.

In the next step, we consider the neighborhood clues (see 6, 11, 12, and 15) and express them with the help of the following conditional sentences:

- 6r. If a house X1 is green and a house X2 is ivory then X1 is right of X2.
- 11r. If a person X1 owns the fox and lives in a house Y1 and a person X2 smokes Chesterfields and lives in a house Y2 then Y1 is next to Y2.
- 12r. If a person X1 smokes Kools and lives in a house Y1 and a person X2 owns the horse and lives in a house Y2 then Y1 is next to Y2.
- 15r. If the Norwegian lives in a house Y1 and Y1 is next to a house Y2 then Y2 is blue.

As these examples show, variables are used instead of pronouns and provide a precise and ambiguity-free replacement for them. These variables are always introduced together with a noun and can be used afterwards like personal pronouns. Finally, we specify the two questions: *Who drinks water?* and *Who owns the zebra?*. These are simple questions but we can query all aspects of the puzzle.

4 Solving the Puzzle

The puzzle is translated with the help of a logic grammar into TPTP notation [7] and then into the input format of E-KRHyper [1]. E-KRHyper is a model generator for first-order logic with equality and uses clauses of the form:

```
HEAD :- BODY.
```

The head is either false or consists of a conjunction or a disjunction of positive literals. The body is either true (in the case of facts) or consists of a conjunction of body literals. Body literals are composed of negative literals and a number of special logical operators, among them stratified negation as failure (\+). Here are a few clauses which illustrate the input format to E-KRHyper. For example, the translation of sentence 1b results in a disjunctive clause of the form:

```
live_in(P,1) ; live_in(P,2) ; live_in(P,3) ; live_in(P,4) ;
live_in(P,5) :- person(P).
```

and five facts for the definite noun phrases:

```
house(1). house(2). house(3). house(4). house(5).
```

The translation of sentence 6b leads to an integrity constraint where the head is false and the body consists of a number of negative literals with a negation as failure operator for testing the inequality of the two variables P1 and P2:

```
false :- person(P1), house(H), live_in(P1,H), person(P2),
live_in(P2,H), \+(P1=P2).
```

The translation of sentence $\mathcal{Z}r$ which represents a pair wise clue results in the following definite clause with one positive literal in the head:

```
prop(H,red) :- house(H), live_in(E,H), englishman(E).
```

Finally, the translation of the two questions results in two conjunctive queries that can be added directly to the theory:

```
answer(Who) :- drink(Who,W), water(W).
answer(Who) :- own(Who,Z), zebra(Z).
```

E-KRHyper takes the entire theory in clausal form as input and generates a finite model consisting of a set of ground atoms that satisfy the clauses, for example:

```
own(j,z). own(n,f). zebra(z). water(w). coffee(c).
drink(j,c). drink(n,w). japanese(j). norwegian(n).
```

During the model generation process the variables of the answer literals are instantiated and form answers to the questions. Thus we find out that the Norwegian drinks water and that the Japanese owns the zebra.

5 Conclusions

There exists **no** natural language processing system that can take the original version of Einstein's Riddle as input and find the correct solution automatically. Without doubt most natural language processing applications which aim at high-quality understanding of texts would benefit from more precise structural semantic knowledge, conceptual knowledge and inference. But as long as it is not possible to solve these kinds of puzzles fully automatically, we will not see significant progress in axiom-based natural language understanding. It might be wiser to shift the focus and bring the user back into the loop and to use a controlled natural language together with an authoring tool that allows the user to work **in cooperation** with a machine in order to solve hard problems.

Acknowledgments

I am grateful for the constructive comments made by Norbert E. Fuchs on earlier versions of this paper and for the feedback from four anonymous reviewers.

References

1. Baumgartner, P., Furbach, U., Pelzer, B.: Hyper Tableaux with Equality. In: Fachberichte Informatik, 12-2007, Universität Koblenz-Landau (2007)
2. Collins, M.: Head-Driven Statistical Models for Natural Language Parsing. Computational Linguistics 29(4), 589–637 (2003)

3. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the First PASCAL Challenges on Recognising Textual Entailment, pp. 1–8 (2005)
4. Lev, I., MacCartney, B., Manning, C.D., Levy, R.: Solving Logic Puzzles: From Robust Processing to Precise Semantics. In: Proceedings of the 2nd Workshop on Text Meaning and Interpretation at ACL 2004, pp. 9–16 (2004)
5. Life International: Who Owns the Zebra? Life International magazine 17, 95 (December 1962)
6. Schwitter, R., Ljungberg, A., Hood, D.: ECOLE – A Look-ahead Editor for a Controlled Language. In: Proceedings of EAMT-CLAW 2003, pp. 141–150 (2003)
7. Sutcliffe, G., Suttner, C.B.: The TPTP Problem Library: CNF Release v1.2.1. Journal of Automated Reasoning 21(2), 177–203 (1998)

Evolving Intrusion Detection Rules on Mobile Ad Hoc Networks

Sevil Sen and John A. Clark

Department of Computer Science, University of York, YO10 5DD, UK
`{ssen,jac}@cs.york.ac.uk`

Abstract. Intrusion detection on mobile Ad Hoc Networks (MANETs) is in its early stages. In this paper, we show how grammatical evolution can be used to evolve detection programs for dropping attacks, a particularly important attack type for such networks.

Keywords: Grammatical evolution, intrusion detection, MANETs.

1 Introduction

A mobile ad hoc network (MANET) is a self-configuring network of mobile nodes connected by wireless links. MANETs do not have any fixed and pre-established infrastructure such as centralised management or base stations in wireless networks. The union of nodes forms an arbitrary network topology that changes frequently due to the mobility of the nodes. In addition, the nodes must cooperate with each other to provide essential networking. Mobile nodes that are within each other's radio range can communicate directly via wireless links, while those that are far apart must rely on other nodes to forward their messages. Since they provide communication even in the absence of a fixed infrastructure, they are very attractive for many applications such as rescue operations, tactical operations, environmental monitoring, conferences, and the like.

Attempts to design intrusion detection systems for MANETs to date are typically carried out entirely by the designer. However, humans are not particularly adept at selecting good choices when complex tradeoffs have to be made. Accordingly we propose to investigate the use of an artificial intelligence based learning technique to explore this difficult design space. In this paper, grammatical evolution (GE) is explored as a technique to detect known attacks on MANETs. Detection rules are evolved to detect a known type of attack on MANETs (dropping attack) and evaluated on networks with varying mobility and traffic patterns.

2 Grammatical Evolution in Intrusion Detection on MANETs

2.1 The Problem

In this paper, we use grammatical evolution [1] to evolve detection rules for dropping attacks on MANETs. In the dropping attack scenario malicious node(s)

drop data packets not destined for themselves to disrupt the network connection. Since malicious nodes need to be on a routing path to drop data packets, they have little reason to drop routing protocol control packets such as RREQ, RREP, and RERR messages used in route discovery and maintenance mechanisms of AODV. So, it is assumed that malicious nodes do not drop routing protocol control packets.

While packet losses usually occur due to congestion in wired networks, there can be other causes on MANETs. Major causes of packet losses on MANETs are given as wireless link transmission errors, mobility and congestion in [2].

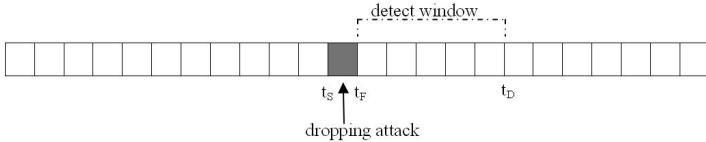
Transmission errors depend on the physical characteristics of the channel and the terrain, and they can not be eliminated or reduced by improving routing protocols [2]. Packet losses due to mobility are the result of one of the main characteristics of MANETs. Mobility of the nodes changes network topology and frequently makes existing routes inactive. Situations like buffer overflows, broken links, and no route to the destination can occur due to mobility and cause packets to be dropped. Lastly, packet losses due to congestion occur when the demands exceeds the capacity of a communication link [2].

Mobility is given as the major cause of packet losses on AODV [2]. It is shown that more than 60% of total packet loss on AODV is due to mobility. We mainly aim to differentiate packet dropping due to malicious behaviour from packet dropping due to mobility in this paper.

2.2 The Method

We evolve a program to detect dropping attacks on MANETs. The evolved program is distributed to each node on the network. We assume that dropping attacks can be detected by the neighbours of the malicious node who sent/forwarded packets to the malicious node, but has not received any acknowledgement from it for a while. Moreover, an attack is assumed to be detected in a time interval Δ after it has occurred. Since features are gathered every time interval by each node locally, a sliding window mechanism, which includes all features in Δ , is applied for training and testing the evolved program.

The *detect window* shown in Figure 1 below is defined as the window that consists of the network features available during the period of length Δ after a dropping attack has occurred. For training purposes we assume that an ideal evolved IDS program should flag the occurrence of an attack precisely Δ seconds after the attack has finished (i.e. using the feature data available during the detect window), at the *detect point* t_D . It should flag "no attack" at or before the attack begins and also after more than Δ seconds after it has finished (i.e. after the detect point). At all other times we do not care whether the evolved program flags "attack" or "no attack". This means that in the training process programs that can detect some attacks earlier than Δ units after they have finished are not punished for doing so. Additionally our training assumes that a network "returns to normality" at $\Delta+1$ seconds after an attack has finished. This is a very straightforward evaluation approach for experimental purposes. Other choices for desired flagging profiles are clearly possible.

**Fig. 1.** Detect Window

Grammar. The grammar used to evolve a program to detect dropping attacks on MANETs and raise an alarm is defined in Table 1:

Table 1. BNF grammar used for the problem

S	$=$	$<\text{code}>$
$<\text{code}>$	$::=$	$\text{if}(<\text{cond}>) \text{ raise_alarm}()$
$<\text{cond}>$	$::=$	$<\text{cond}> <\text{set-op}> <\text{cond}> \mid <\text{expr}> <\text{rel-op}> <\text{expr}>$
$<\text{expr}>$	$::=$	$<\text{expr}> <\text{op}> <\text{expr}> \mid (<\text{expr}> <\text{op}> <\text{expr}>) \mid <\text{pre-op}> (<\text{expr}>) \mid <\text{pre-op2}> (<\text{expr}>, <\text{expr}>) \mid <\text{var}>$
$<\text{op}>$	$::=$	$+ \mid - \mid / \mid *$
$<\text{pre-op}>$	$::=$	$\text{sin} \mid \text{cos} \mid \text{log} \mid \text{ln} \mid \text{sqrt} \mid \text{abs} \mid \text{exp} \mid \text{ceil} \mid \text{floor}$
$<\text{pre-op2}>$	$::=$	$\text{max} \mid \text{min} \mid \text{pow} \mid \text{percent}$
$<\text{rel-op}>$	$::=$	$< \mid \leq \mid > \mid \geq \mid == \mid !=$
$<\text{set-op}>$	$::=$	$\text{and} \mid \text{or}$
$<\text{var}>$	$::=$	feature set in Appendix A

Features used in the grammar are given in Appendix A. We use both mobility-related features as well as packet-related features as input to the evolution system. While some of these features give information about mobility directly (such as changes in the number of neighbours), some of them can be the result of mobility (such as added routes in the last period). Packet-related features include routing protocol control packets and transport protocol packets. AODV[5], which is one of the most commonly used on-demand routing protocols on MANETs, is used in this paper. Because TCP expects acknowledgement packets from the destination and lack of acknowledgements may indicate dropping attacks, it is chosen as a transport layer protocol. All features are gathered periodically at every second by each node.

Fitness Function. As evaluation measures we use detection rate (the ratio of correctly detected intrusions to the total intrusions on the network) and false positives rate (the ratio of normal activities which are incorrectly marked as intrusions to the total normal activities on the network). Low false positive rate is as important as high detection rate for a good intrusion detection system. That's why the constant k ($=4$) in the fitness function is used for decreasing the false positive rate.

$$\text{Fitness} = \text{detection rate} - k * \text{false positive rate} \quad (1)$$

GE Parameters. The parameters used in GE are given in Table 2.

Table 2. GE Tableau for detecting dropping attacks on MANETs

Objective:	Find a program to detect dropping attacks on MANETs
Non-Terminal Operators:	The binary operators +,-,* , /, pow, min, max, percent The unary operators sin, cos, log, ln, sqrt, abs, exp, ceil
Terminal Operators:	The feature set in Appendix A
A Fitness cases:	The given sample of network data marked malicious or non-malicious
Raw Fitness:	The detection rate over the fitness cases subtract the false positive rate over the fitness cases
Standardised Fitness:	Same as raw fitness
Parameters	Populations Size = 100 Termination when Generations= 1000 Prob. Mutation = 0.02, Prob. Crossover = 0.9 Steady State

2.3 Experiment and Results

The evolved program is evaluated on the networks simulated by ns-2 [3]. Mobility of the nodes is simulated by the Random Waypoint model which is created using BonnMotion [4]. In the Random Waypoint model, each node moves from its current location to a random new location with random speed and pause time in determined speed/pause time limits [6]. Different network scenarios are created with different mobility levels and traffic loads. The parameters of the network simulation are given in Table 3.

The algorithm is evolved using the training data collected from a network under medium mobility with 30 TCP connections. The same network with dropping attacks and without attacks are used for training to reduce false positives. Evolved programs are evaluated on different network scenarios and the results are presented in Table 4. There are two different networks under medium mobility in Table, which are simulated with different mobility and traffic patterns, since one of them is used for evolution. In the results, false positives increase in proportion to the mobility (as expected). False positives also increase under high traffic loads (which can be a source of non-malicious packet loss).

Table 3. The parameters of the network simulation

network dimensions	1000x500
number of nodes	50
packet traffic	TCP with 20 and 30 connections
speed	0-20 m/sec
pause time	40,20,5 sec (low, medium and high mobility)
routing protocol	AODV
radio propagation	two-way ground model with 250m transmission range
local link connectivity	AODV periodic hello messages
simulation time	1000 sec (testing), 2000 sec (training)

Table 4. The experiment results

Scenarios	Detection Rate	False Positive Rate
Low mobility, 20 TCP connections	79.59%	3.81%
Low mobility, 30 TCP connections	93.85%	5.25%
Medium mobility, 20 TCP connections	92.45%	3.95%
Medium mobility, 30 TCP connections	87.04%	6.30%
Medium mobility, 20 TCP connections	90.48%	4.07%
Medium mobility, 30 TCP connections (training)	82.64%	5.53%
High mobility, 20 TCP connections	83.33%	5.05%
High mobility, 30 TCP connections	84.38%	6.22%

3 Conclusion

Grammatical evolution essentially "grows" intrusion detection programs by evaluating populations of potential programs and subjecting them to a variety of genetically inspired operators. The results show that our grammatical evolution technique has significant potential for evolving efficient detectors from such discrimination attacks. The approach has good chances of generalizing: we aim now to simulate a variety of attacks and employ the same grammatical evolution technique to evolve detectors for those attacks.

One interesting feature of the approach is that we can evolve detectors that can be evaluated with respect to how well they detect a range of attacks rather than a single specific attack. Though we have not done so here, it should also be possible to employ multi-objective evaluation mechanisms to explore optimal tradeoffs between resources consumed by programs (e.g. memory and power) and detection efficacy. This is very difficult for a human designer to address.

References

1. Ryan, C., Colline, J.J., O'Neill, M.: Grammatical Evolution: Evolving Programs for an Arbitrary Language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) EuroGP 1998. LNCS, vol. 1391, pp. 83–95. Springer, Heidelberg (1998)
2. Lu, Y., Zhong, Y., Bhargava, B.: Packet Loss in Mobile Ad Hoc Networks. Technical Report, Dept. of Comp. Sci, Purdue Uni., TR 03-009 (2003)
3. The Network Simulator: ns-2, <http://www.isi.edu/nsnam/ns>
4. BonnMotion: A Mobility Scenario Generation and Analysis Tool, <http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/>
5. Perkins, C.E., Royer, E.M.: Ad-hoc On-Demand Distance Vector Routing. In: Proc. of the 2nd IEEE Workshop on Mobile Computer Systems and Applications, pp. 90–100 (1999)
6. Camp, T., Boleng, J., Davies, V.: A Survey of Mobility Models for Ad Hoc Network Research. Wireless Communication and Mobile Computing, 483–502 (2002)

Appendix A. The Features

neighbours	number of neighbours
added_neighbours	number of added neighbours
removed_neighbours	number of removed neighbours
active_routes	number of active routes
repaired_routes	number of routes under repair
invalidated_routes	number of invalidated routes
addedroutes_disc	number of added routes by route discovery mechanism
addedroutes_notice	number of added routes by overhearing
updated_routes	number of updated routes (modifying hop count, sequence number)
addedroutes_repaired	number of added routes under repair
invroutes_timeout	number of invalidated routes due to expiry
invroutes_other	number of invalidated routes due to other reasons
recv_rreqPs	number of received route request packets destined to this node
recvF_rreqPs	number of received route request packets to be forwarded by this node
send_rreqPs	number of broadcasted route request packets from this node
frw_rreqPs	number of forwarded route request packets from this node
recv_rrepPs	number of received route reply packets destined to this node
recvF_rrepPs	number of received route reply packets to be forwarded by this node
send_rrepPs	number of initiated route reply packets from this node
frw_rrepPs	number of forwarded route reply packets from this node
recvB_rerrPs	number of received broadcast route error packets (to be forwarded or not)
send_rerrPs	number of broadcasted route error packets from this node
recv_aodvPs	number of received total routing protocol packets
recvF_aodvPs	number of received total routing protocol packets to be forwarded
send_aodvPs	number of initiated total routing protocol packets from this node
frw_aodvPs	number of forwarded total routing protocol packets by this node
recvF_dataPs	number of TCP data packets forwarded by this node (not acknowledged)
send_dataPs	number of TCP data packets initiated by this node (not acknowledged)

On the Usefulness of Interactive Computer Game Logs for Agent Modelling

Matthew Sheehan and Ian Watson

Dept. of Computer Science, University of Auckland

Private Bag 92019, Auckland, New Zealand

`mshe065@ec.auckland.ac.nz, ian@cs.auckland.ac.nz`

Abstract. Interactive computer game logs show the potential for use as replacement for time-consuming supervisory learning processes for embodied, situated agents. However, due to the inherent nature of the data in the logs themselves, for the time being this promise cannot be fulfilled. An unsuccessful attempt to use largely rule-based data mining processes to learn behaviours from game logs led to finding the inherently top-down nature of such processes was fundamentally at odds with the unsupervised bottom-up learning requirement of the problem. Innate issues with game logs toward the goal of unsupervised agent learning are discussed. Possible approaches to the problem subsuming successful applications of various methods in narrower fields are presented for both symbolic and sub-symbolic advocates.

1 Introduction

This paper primarily addresses the use of interactive computer game logs for embodied, task-generalised, goal oriented agent learning. In particular, in our view, is the potential of game logs to supplement or even replace time-consuming supervisory learning methods. While our own research was largely unsuccessful in achieving this goal, what was learnt is considered useful for future researchers and is presented here.

A computer game log can be recognised as a file that is used to store captured (usually live) game data, often in a sequential or time-stamped manner. A game log file can be human text readable, or require a parsing application (for an example see [1]). Whether it is the sequential moves taken in a turn based game or a FPS ‘demo’ file that captures all server-client network traffic down to timeframes of a tenth of a second, the defining and encompassing attribute of the term’s use in this paper is the storage of game activity in an abridged manner.

2 Motivation and Related Work

The inherent opportunities and motivation for artificial intelligence research in computer game genres where the player (and or opponents) are situated within

the game environment is well documented by Laird and van Lent [2]. The First Person Shooter (FPS) genre in particular has been popular with researchers for its complexities and availability of software. A number of internet sites house large, freely accessible game log repositories built using game logs uploaded by game enthusiasts.

Some of the best known uses of game logs to model the behaviour of simulated agents is the Robocup Coach and Simulated League Competitions (e.g. [3], [4]). Robocup examples can be differentiated to the current research by the general level of complexity of data within the game logs themselves, due to the relatively simple nature of the agents, game rules and two-dimensional environment the Robocup games are situated in.

More pertinently, Gorman et al [5] have used a three dimensional, FPS environment to implement a two tier (reactive/strategic) situated agent using Quake II game demo files. Tactical tier implementations were investigated by the two research groups separately but were never integrated.

The majority of other implementations of situated/embedded agents in FPS environments use a largely symbolic, top-down approach [6,7]. However, none of the symbolic examples listed above use game logs as a means of learning new rules. The closest examples are perhaps the annotated expert decisions that have been used by König and Laird [8] in an adventure game setting and the expert traces used by Nejati et al [9] to learn hierarchical tasks, currently a feature of Choi et al's FPS agent implementation [7].

3 Approach

Quake II demos, though not able to be directly read using a text-editor, given the correct map the in-game console are able to be used to playback the entire game. With the API, it is possible to extract game information/attributes from the world state at each tenth of a second frame. Because of the inherent rule-based nature of games and previous successful rule-based agent implementations, a largely rule-based data mining approach was taken, forming flat files from extracted information and running them on data mining software, using both algorithmic and brute force techniques. However, it was found that this approach was not only largely unsuccessful, but more interestingly, perhaps deeply flawed, discussed in the next section.

4 The Usefulness of Interactive Game Logs in Agent Modeling

The symbol grounding problem is a key issue for symbolic approaches to agent modelling with FPS demo game logs in particular due to the sub-symbolic (i.e. quantitative) nature of the data. Using an example from our own research in attempting to find aiming rules, one time-frame's captured world state might contain the positional coordinates and angles of player and opponent but no labelled links to what these positions and angles represent. Unlike [8] it is infeasible to elucidate every action in a FPS setting.

In attempting to model item-collection behaviour from the game logs it was found that some actions are hierarchical and interruptible. For example, should an agent be walking towards an ammunition pack when an opponent appears, an action to protect itself against attack should (generally) override the collection of items.

Though the quality of behavioural data seems evident when watched as a video, as current interactive game log attributes are recorded in a global positioning system, these behaviours cannot be explicitly extracted. For instance, aiming information that is agent-relative for learning general rather than situation specific behaviours can be calculated from these attributes, but it requires the use of complex trigonometric processes to be applied to the data (perspective projection).

Deriving basic lessons by watching experts means the ability to break down and slow down complex maneuvers into their constituent smaller manoeuvres, an exercise that effectively requires expert knowledge to begin with. An expert manoeuvre has implicit within it tactical weapon control (which weapon to use, when), modified circle/strafing behaviour, map knowledge and ambushing behaviour, with each constituent part seamlessly blended into the others parts, unlikely to be repeated exactly again, even in the same part of the map.

The game environment is three dimensional, but environment items and agent movements and decisions make the problem space of a much higher dimensionality. A means of filtering and lowering the dimensionality of this data (addressed in the next section) could help mitigate this issue.

5 Possible Approaches

This section shows possible approaches to implementing a means of using interactive game logs for embodied agent learning by suggesting general means of approach coupled with techniques that have overcome some of the inherent difficulties outlined above in more specific domains.

5.1 Symbolic/Rule-Based Approach

The largest difficulty in learning declarative rules automatically from interactive game logs is in the sub-symbolic and procedural nature of the game log data. One answer (not without its own difficulties, admittedly) to the problem of the symbol grounding problem would be to make the contents of the game logs declarative, rather than procedural. For this to occur, it would be necessary that the game itself to work in declarative terms or at least have a declarative layer built into the game engine (something akin to the ‘Symbolic Environmental Representation’ in [8]). From this game layer symbolic states and transitions could be collected for logs, to be used by symbolic cognitive architectures or integrated for implementation with artificial agent game interfaces such as Berndt and Watson’s [10] Open AI Standard Interface Specification (OASIS).

This layer might be built using a standardized game ontology perhaps using qualitative reasoning techniques, with particular regard to qualitative spatial

reasoning [11,12,13] (using abstractions to model physical space) and qualitative physics [12] (modelling the physics of a world using rules rather than exact quantitative methods). Zagal et al [14] have already begun to outline a game ontological language for game analysis that could possibly be extended to the standardised qualitative programming of some types of game, while Zhou and Ting [15] have used qualitative methods in a three-dimensional FPS environment to model moveable objects. While these two examples, in themselves, do not come close to a complete solution to modelling a physical three dimensional simulation symbolically, they are indicative of approaches that may be successful in the future.

Nejati et al's [9] hierarchical learning approach is a beginning to confronting hierarchical learning processes. However, like the annotated expert decisions in [8], their top down approach means that it requires positive (i.e. successful) examples with a set goal and start point are required for each behaviour to be learnt, not possible from recorded game demos without large scale trainer intervention.

It would seem that using data mining techniques would be useful, especially in regard to the large amounts of data that are present, plus current techniques providing means to bring quantitative, inherently time-based data to a symbolic level using knowledge based temporal abstraction [16]. Unfortunately, rule learning from even qualitative data time series seems to still require *top-down* methods of knowing the patterns and relationships you are looking for, before being able to find the sort of relationships between each sort of pattern [17]. Our own discouraging results in using data mining techniques to find rules in demo log data emphasise the inherent fundamental discord between the necessity of a bottom-up learning approach and one of the golden rules of data mining being already knowing what you are looking for [18]. Even when using brute force techniques, rules found must be recognised as being valid.

It is believed a bottom up approach could work better, bottom up approaches also useful in not requiring agent sophistication before learning begins.

5.2 Sub-Symbolic Approaches

A sub-symbolic approach to modelling an agent would seem to be the best way to approach the problem given the nature of the data in the current game logs. Perhaps more efforts have not been made in this area by sub-symbolic and nouvelle AI advocates due to a generally held belief championed by Brooks [19] condemning simulated environments.

It is interesting to note that after beginning with completely reactive, strictly bottom up approaches using neural nets and self organising maps to model agents' movement [20], later strategic layered approaches took on more top-down attributes including setting nodal positional points (formed by clustering the entire set of player positions) which the agent would travel between and the making of goals in the form of item pickup points [5].

To be forced to create a sub-cognitive means to deal with data that is much closer to the kind of data a real-world agent would face could be seen as a effort

to model not only a real world environment, but the kind of sensory inputs that would be received also. However, a large amount of preprocessing of the data is still required, even if specific logs are created (instead of using expert logs from the online repositories) before it can be used for learning. While it could be noted that the important part of the logs is the behavioural data they contain, not the form they are in, the ability to ‘cheat’ (whether consciously or unconsciously) to produce circumstances favourable to the agent’s learning circumstances is possible. Means to extract important features or bring levels of dimensionality down with large amounts of data include self-organised maps, principle component analysis (both used in methods by Thurau et al [21]) and independent component analysis. The success of Floyd et al [4] in largely automating the learning process, even in a less complex environment and producing effectively reactive agents, shows CBR methods could provide a means to sidestep this process.

6 Conclusions

Interactive game logs show promise as a source for expert behaviour to supplement or even replace time-consuming supervised expert tracing schemes. However, there are still a number of issues that need to be addressed before their full potential can be utilised. Both the content and means to extract that content automatically has a large bearing on their usefulness: the larger the supervisory aspect required by the method of agent training, the less useful they become, as it is easier to produce single logs and elucidate every action separately in (basic) movements for agents to learn than to extract behaviours from the thousands of game logs created by expert players by hand. Recommendations to increase this usefulness can be generalised into two parts: firstly, the improvement of means of information representation in the logs themselves; and secondly, the choice of methods by the researcher that lead to largely automatic information extraction.

References

1. Gorman, B., Fredriksson, M., Humphrys, M.: Qase - an integrated api for imitation and general ai research in commercial computer games. In: Proceedings of the IEEE 7th International Conference on Computer Games, CGAMES, pp. 207–214 (2005)
2. Laird, J.E., van Lent, M.: Human-level ais killer application: Interactive computer games. In: Proceedings of the AAAI Fall Symposium, pp. 80–97. AAAI Press, Menlo Park (2000)
3. Kuhlmann, G., Knox, W.B., Stone, P.: Know thine enemy: A champion robocup coach agent. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI 2006), pp. 1463–68 (2006)
4. Floyd, M.W., Esfandiari, B., Lam, K.: A case based reasoning approach to imitating robocup players. In: Proceedings of the Twenty-First International FLAIRS Conference. AAAI Press, Menlo Park (2008)
5. Gorman, B., Thurau, C., Bauckhage, C., Humphrys, M.: Bayesian imitation of human behavior in interactive computer games. In: Proceedings of the 18th International Conference on Pattern Recognition, ICPR, vol. 1, pp. 1244–1247 (2006)

6. Laird, J.E.: It knows what you're going to do: adding anticipation to a quakebot. In: AGENTS 2001: Proceedings of the fifth international conference on Autonomous agents, pp. 385–392. ACM, New York (2001)
7. Choi, D., Konik, T., Nejati, N., Park, C., Langley, P.: A believable agent for first-person perspective games. In: Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment International Conference (AIIDE 2007). AAAI Press, Menlo Park (2007)
8. Konik, T., Laird, J.: Learning goal hierarchies from structured observations and expert annotations. In: Proceedings of the Fourteenth International Conference on Inductive Logic Programming, pp. 198–215. Springer, Heidelberg (2004)
9. Nejati, N., Langley, P., Konik, T.: Learning hierarchical task networks by observation. In: Proceedings of the 23 rd International Conference on Machine Learning, vol. 148, pp. 665–672. ACM Press, New York (2006)
10. Berndt, C.N., Watson, I., Guesgen, H.: Oasis: An open ai standard interface specification to support reasoning, representation and learning in computer games. In: Aha, D., Muñoz-Avila, H., van Lent, M. (eds.) Proceedings of the 2005 IJCAI Workshop on Reasoning, Representation, and Learning in Computer Games, pp. 19–24 (2005)
11. Freksa, C.: Qualitative spatial reasoning. In: Mark, D.M., Frank, A. (eds.) Cognitive and Linguistic Aspects of Geographic Space, pp. 361–372. Kluwer Academic Publishers, Dordrecht (1991)
12. Forbus, K.D.: Qualitative reasoning. In: Tucker, J.A. (ed.) The Computer Science and Engineering Handbook, pp. 715–733. CRC Press, Boca Raton (1997)
13. Forbus, K.D., Mahoney, J.V., Dill, K.: How qualitative spatial reasoning can improve strategy game ais. IEEE Intelligent Systems 17, 25–30 (2002)
14. Zagal, J., Mateas, M., Fernandez-Vara, C., Hochhalter, B., Lichti, N.: Towards an ontological language for game analysis. In: Proceedings of the Digital Interactive Games Research Association Conference, DiGRA 2005 (2005)
15. Zhou, S., Ting, S.P.: Qualitative physics for movable objects in mout. In: ANSS 2006: Proceedings of the 39th annual Symposium on Simulation, Washington, DC, USA, pp. 320–325. IEEE Computer Society, Los Alamitos (2006)
16. Shahar, Y.: A framework for knowledge-based temporal abstraction. Artificial Intelligence 90, 79–133 (1997)
17. Sacchi, L., Larizza, C., Combi, C., Bellazzi, R.: Data mining with temporal abstractions: learning rules from time series. Data Mining Knowledge Discovery 15, 217–247 (2007)
18. Pyle, D.: Data Preparation for Data Mining. Morgan Kaufmann, San Francisco (1999)
19. Brooks, R.A.: Elephants don't play chess. Robotics and Autonomous Systems 6, 3–15 (1990)
20. Thurau, C., Bauckhage, C., Sagerer, G.: Combining self organizing maps and multilayer perceptrons to learn bot-behavior for a commercial game. In: Proceedings of GAME-ON 2003, pp. 119–123 (2003)
21. Thurau, C., Paczian, T., Bauckhage, C.: Is bayesian imitation learning the route to believable gamebots? In: GAME-ON North America, pp. 3–9 (2005)

An Empirical Study on the Effect of Different Similarity Measures on User-Based Collaborative Filtering Algorithms

Ashish Sureka and Pranav Prabhakar Mirajkar

Software Engineering and Technology Labs (SETLabs),
Infosys Technologies Limited, India
{Ashish_Sureka,Pranav_Mirajkar}@infosys.com

Abstract. Computation of similarity between user profiles (user rating vectors) is one of the core components of user-based (k -nearest-neighborhood based) collaborative filtering algorithms. Present techniques work by identifying or selecting a similarity function by the designer of the recommendation engine and keeping it fixed throughout the collaborative filtering process and using the same function to compute the neighborhood of every user. However, we found that there is no single similarity measure that gives best predictive accuracy for all users. We see this as a limitation of current systems. For the same user, applying different similarity functions results in different predictive accuracy. We propose that the accuracy of user-based collaborative filtering recommendation engines can be further increased by learning an optimal similarity function for a particular user and by applying different similarity measure for different users. We present an empirical study on the effect of eleven different similarity measures on the predictive accuracy of user-based collaborative filtering algorithms.

Keywords: Collaborative Filtering, Similarity Measures, Recommendation Systems, Data Mining.

1 Introduction

Collaborative filtering based recommendation systems aims at predicting a users' interest or rating for an item based on other users' interest and ratings. The main idea behind collaborative filtering algorithms is based on the intuition that the interest of an item to an individual can be predicted based on the interests and likings of other like-minded individuals. An overview of the field of recommender system and a survey of the state-of-the-art methods covering content-based, collaborative and hybrid approaches can be found in [1].

User-based collaborative filtering using k -nearest neighbor ($k - NN$) algorithm is amongst the initial and one of the most widely accepted technique. The first step of user-based collaborative filtering algorithm is to compute similarity between user profiles (user rating vectors). In its simplest form, the similarity between two users is computed by taking into account only those items that

have been rated by both the users. Once the top-N similar users are identified, the prediction for an item for the target user (also called as the active user) is determined as the weighted average of the ratings of similar users for the particular item. The weights quantify how similar the two rating vectors are. Thus $k - NN$ based collaborative filtering can be divided into two steps or phases: the first of neighborhood determination and the second phase of prediction or recommendations.

The manner in which similarity between two users is computed can have a significant impact on the overall performance of the system and the output it generates. Present techniques work by identifying or selecting a similarity function by the designer of the recommendation engine and keeping it fixed throughout the collaborating filtering process and using the same function to compute the neighborhood of every user. However, we found that there is no single similarity measure that gives best predictive accuracy for all users. We see this as a limitation of current systems. For the same user, applying different similarity functions results in different predictive accuracy. We propose that the accuracy of user-based collaborative filtering recommendation engines can be further increased by learning an optimal similarity function for a particular user and by applying different similarity measure for different users. Few references [3], [6] and [8] to the literature argues that there is no common consensus on the appropriateness of collaborative filtering design parameters such as similarity metrics, identify fallacies in the calculation of commonly used similarity metrics and the effectiveness of such measures in the recommendation problem have been seldom questioned. Few references [2] and [4] to the literature proves the point that there is no single similarity measure that performs superior to other measures in all scenarios. Few references [5] and [7] point to empirical studies done on examining the effectiveness of different similarity measures on domains other than the one addressed in this paper. The eleven similarity measures that we used for our study are Euclidean Distance, Manhattan Distance, Chebyshev Distance, Canberra Distance, Simple Matching, Jaccard, Russell, Hamann, Dice, Cosine measure, Pearson correlation.

The purpose of this work is to get a sense of the degree of variability in the result (in terms of the set of neighboring users for a particular user and the magnitude of similarity) across different similarity measures. An empirical study on 11 different similarity measures on a publicly available dataset can shed more light on this. We wanted to test the possibility of certain similarity measure consistently performing well for certain types of users. Our motivation behind testing this hypothesis is that if there is a case where a particular similarity measure performs well for a certain set of user types where another similarity measure performs well for another set of user types then instead of keeping a single similarity measure fixed for the entire user population. Determination of the most appropriate similarity measure for each user will be an offline activity, the result of which will be fed into the production system. Output of the offline activity is a mapping between users and similarity measure. The mapping will be hard coded in production system. Process of computing the mapping will

Fig. 1. Application of different similarity measures for two users

be triggered after a specific time interval (or event based) and the production system will be updated accordingly. The idea is to make the system dynamic and flexible in the sense that different users can be associated to different similarity measures and this mapping is also performed periodically so that the system is continuously adapting to the changed environment.

2 Experimental Data, Procedure and Result

For our experiments, we used the publicly available MovieLens dataset which can be downloaded from the GroupLens Research Lab Website. We used the MovieLens Data Sets having 100,000 ratings. The first experiment we performed was to see the impact of different similarity measures on the first phase of user-based collaborative filtering i.e. neighborhood determination. The experiment was performed using the entire 100,000 ratings. We computed the similarity score between all the users using 11 similarity measures. For example, the similarity between user 1 and a total of 943 users (including self) were computed for all the 11 similarity measures. We then computed the top 20 and 50 neighbors for each user and for each similarity measure.

Figure 2 shows some of the graphs that indicate the variability in the output of neighborhood determination across different similarity measures.

Graph A in Figure 1 shows the frequency of the top 20 neighbors across 11 similarity measures for User ID 88. For each similarity measure, top 20 neighbors with respect to User ID 88 are selected. After that frequency of each User ID appearing in top 20 is computed across all 11 similarity measures. The X-Axis of Graph A represents all the User IDs occurring in top 20 neighborhood list for all similarity measures. The Y-Axis represents their respective frequencies. Similarly, Graph B of Figure 2 shows the frequency distribution of top 20 neighboring users for User ID 935. User IDs 88 and 935 are randomly selected from set of 943 users. Similar graphs can be plotted for rest of the users also and we selected two users randomly for validation of our hypothesis. We observed that there are 59 and 53 users appearing on X axis of Graph A and B respectively (for top 20 neighbors). This shows the variability in the Set consisting of top 20 User IDs nearest to User ID 88 and 935 for different similarity measures. In

Fig. 2. Mean absolute Error for different similarity measure for two selected users

Graph A, we found that there is not even a single user which appeared in the top-20 neighborhood list of all the similarity measures. Referring to Graph A, there are less than 10 users amongst 59 users who appeared in at-least 9 of the top-20 neighborhood list.

To compute Mean Absolute Error (MAE), we used test data provided by MovieLens dataset. The dataset contains rating for selected (80%) set of items from original dataset. The remaining 20% are used as test cases. There are five such different test datasets. The training dataset and its corresponding test dataset are disjoint. The rating by a user for an item for a given similarity is predicted based on the user's top 20 neighbors for that similarity measure. In these top 20 neighbors we consider only those users which have rated the item under consideration.

Table 1 shows predicted and actual ratings for user 257 for various items in the test dataset and for various similarity measures.

Figure 2 shows MAE for 2 randomly selected users from set of 943 users. In each of the graph (A, B), X-Axis represents 11 similarity measures, while Y - Axis represents corresponding MAE. We observe that for a user each similarity measure has different MAE values. Also there is no specific trend between the MAE values of all the similarity measure of each user. As shown in Graph A, for User 57 we can see that Euclidean and Manhattan measures have the highest MAE. The data plotted in Figure 2 support the hypothesis that different similarity measure can result in different predictive accuracies and so far we did not observe any specific trend in these results. We also observe that there is no single similarity measure (amongst the ones we chose for our experiments) that outperforms all other similarity measures consistently. In our future work, we want to see if there is a correlation between a similarity measure and user type.

In the next phase, we computed best similarity measure (i.e. similarity measure which has minimum MAE) and worst similarity measure (i.e. similarity measure which has maximum MAE) for a set of 100 users. In Figure 3, for Graphs A and B, X-Axis represents the similarity measures while Y Axis represents the number of times that particular similarity measure occurred as best and worst similarity measure respectively. For a given user there can more than one similarity measure which occurs as a best similarity measure or worst similarity measure. In such a case, the best or worst count is attributed to all the similarity measures which occurred as best or worst. Hence, due to the repetition

Fig. 3. Count of each similarity measure as a best and worst for set of 100 Users**Fig. 4.** Mean Absolute Error for Different datasets

the sum of the count of best or similarity measure is more than 100. In Graph A we can see that, Pearson similarity measure occurred as best similarity measure for more than 20 times. But at the same time it can be seen from Graph B that it also occurred as worst similarity measure for 25 times, which is the highest. Also in both the graphs, we can observe that there is no single similarity measure which clearly stands out as the best or the worst.

We performed some more experiments where we considered set of 100 users but used different test datasets. We computed similarity distance between all the 100 users for all the similarity measure. Then we calculated top 20 neighbors for each of the 100 users for all the similarity measures. The next step was to compute predicted ratings for all the users for all possible items. After calculation of predicted rating we calculated absolute error for each user for each item for all the similarity measure. Then we found out MAE for each user. Next, we calculated mean of MAE for each similarity measure across all the users. The same steps of procedure were followed for the other dataset.

Figure 4 shows the plot of different similarity measures and their corresponding MAE values for two different datasets. In Graph A and B of Figure 4, X - Axis represents all the similarity measures, while Y - Axis represents corresponding MAE. The Graphs have been plotted for the same set of 100 users from two different datasets. The users in two datasets have rated different items i.e. there might be some items for which some users in one dataset have rated while some of those in other dataset have not and vice versa. As seen in Graph A of Figure 4, Russell similarity measure provides the least MAE, while in Graph

B it has second highest MAE. From the two graphs we can see that the similarity measures which have lesser MAE might or might not have lesser MAE in Graph B. The same holds true for higher similarity measures with higher MAE. This indicates that similarity measure results are dependent on set of users. In our future work, we want to extend our study by finding any possibility of correlation between similarity measures and properties of the dataset. For example, a study on the performance of a similarity measure on the type of dataset like sparse dataset or dense dataset or studying the correlation between similarity measures and rating scale, ratio of total number of users to the total number of items in the rating matrix etc.

3 Conclusion

In this paper, we explored the possibility of building a user-based collaborative-filtering system wherein multiple similarity measures are used at the back-end depending on their suitability for each user. We found that there is no single similarity measure that gives best predictive accuracy for all users. We performed an empirical study on a publicly available dataset and presented results and our observations on the variability in the predictive accuracies obtained as a result of applying different similarity measures on different users and test datasets.

References

1. Adomavicius, G., Tuzhilin, A.: Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering* 17(6), 734–749 (2005)
2. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Improving the effectiveness of collaborative filtering on anonymous Web usage data. In: Proceedings of the IJCAI 2001 Workshop on Intelligent Techniques for Web Personalization (ITWP 2001), Seattle (August 2001)
3. Herlocker, J., Konstan, J.A., Riedl, J.: An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. *Information Retrieval* 5(4), 287–310 (2002)
4. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing* 7(1), 76–80 (2003)
5. Spertus, E., Sahami, M., Buyukkokten, O.: Evaluating Similarity Measures: A Large-Scale Study in the Orkut Social Network. In: The Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL (August 2005)
6. Symeonidis, P., Nanopoulos, A., Papadopoulos, A., Manolopoulos, Y.: Collaborative Filtering: Fallacies and Insights in Measuring Similarity. In: Proceedings of the PKDD Workshop on Web Mining (WebMine 2006), Berlin, Germany (2006)
7. Lin, F., Goh, H.L.D., Foo, S.: The effect of similarity measure on the quality of query clusters. *Journal of Information Science* 30(5), 396–407 (2004)
8. Ahn, H.J.: A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences: an International Journal* 178(1) (January 2008)

Using Self-Organizing Maps with Learning Classifier System for Intrusion Detection

Kreangsak Tamee¹, Pornthep Rojanavasu¹, Sonchai Udomthanapong¹,
and Ouen Pinngern²

¹ Department of Computer Engineering, Faculty of Engineering,
Research Center for Communication and Information Technology (ReCCIT),
King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10200, Thailand
² Department of Computer Science, Faculty of Science, Ramkhamhaeng University,
Bangkok 10240, Thailand
{s6060207, s8060022, s9060705}@kmitl.ac.th, ouen@ru.ac.th

Abstract. Learning Classifier Systems (LCS) have previously been shown to have application in Intrusion Detection. This paper extends work in the area by applying the Self-Organizing Map (SOM) for creating the new input string by 2-bit encoding rely on degree of deviation of normal behaviour. The performance of systems is investigated under an FTP-only dataset. It is shown that the proposed system is able to perform significantly better than the conventional XCS, modified XCS and twelve ML algorithms.

Keywords: Intrusion Detection, LCS, Self-Organizing Map.

1 Introduction

As interconnections among computer systems grow rapidly, Intrusion Detection Systems (IDSs) is an important part of network security. A detailed survey and taxonomy of practical IDSs may be found in the literature [1]. Some are anomaly based and others are signature based. However, no detection system can catch all types of intrusions. Each model has its strengths and weaknesses in detecting different violations in networked computer systems. At the moment most of the researchers are interested in improving intrusion detection which includes artificial intelligencer [2].

Intrusion detection can be considered as a classification problem, where a bad or illegitimate activity in a computer system must be distinguished from normal activity. One of the classification methods by using both reinforcement learning and genetic algorithms to model a dataset is through the use of production system rules. In these system, known as Learning Classifier System (LCS). XCS was introduced by Wilson [3] as an enhanced version of the traditional LCS proposed by Holland [4] has demonstrated excellent performance on a number of data mining tasks [5]. Applying XCS and develop some modification on the XCS [6] to intrusion detection have proposed by using KDD Cup 99 data set [7]. Their studies showed that XCS outperform other classification algorithms, especially the modified XCS.

The main contribution of this paper is to improve XCS's application in the intrusion detection. We investigate a Self-Organizing Map (SOM) for clustering each feature separately to obtain relatively of normal behaviour patterns. Each feature of incoming data is quantized into four levels with 2-bit encoding rely on degree of deviation of normal behaviour. After that, create the input string for XCS by bit combination of each feature, let's the XCS learn and classify the data. Our findings suggest that the accuracy as well or better than other methods.

The paper is structured as follows. Section 2 will provide the system architecture and brief description of XCS. Section 3 describes the 1999 KDD data set. The experimental is explained in section 4. Section 5 provides the conclusion and future works.

2 System Architecture

The proposed system architecture consists of three main parts: abnormal quantization, combination parameters and XCS as shown in figure 1. The idea is, firstly, we calculate the level of abnormal behavior in each feature from SOM which is trained by normal behavior. Secondly, create the input string for XCS by bit combination of each feature. Lastly, let's the XCS learn and classify the data. The detail of each part is explained in the following sections.

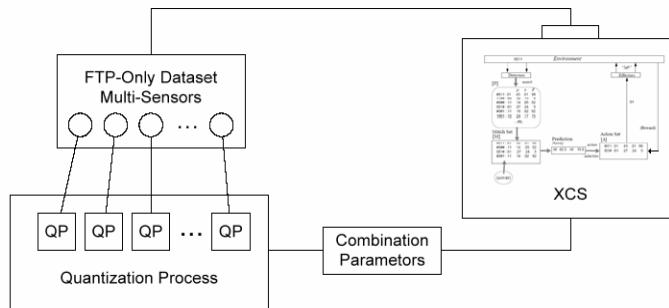


Fig. 1. The proposed IDS architecture

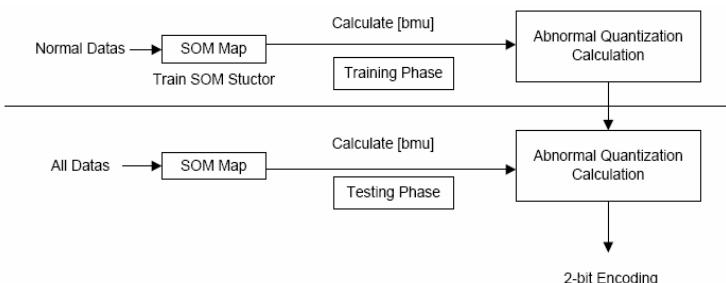


Fig. 2. Flow of data in SOM encoding phase

2.1 Quantization Process

Each module based on building model of normal data. The quantization error is calculated from the normal model in observed data. If the quantization error is grater than the threshold then the observed data is reported as an attack. In our approach, we hypothesize that each feature of data is independent and has one more group of normal behavior on each feature. Therefore, in training phase we create SOM model for clustering each feature separately. After that, in testing phase, we use these models quantize each feature of incoming data into four levels with 2-bit encoding. The overall quantization process can show you as figure 2.

2.1.1 Abnormal Quantization Calculation

In our architecture, we use 29 SOM models for cluster feature of data. Each SOM has size 5×1 and trains only normal behavior data.

The learning process of an SOM [8] can be thought in terms of continuous adaptation of nodes for input vectors. We summarize the learning process as a repetition of following basic tasks:

1. The input vector $x(t)$ is fed into every node in the map to identify the output vector's winning node. It is common to use Euclidean distance as the basis to measure similarities.
2. The weight of the winning node c is tuned by the difference between the input vector and the weight vector. Not only the winning node is learning but also its neighborhood nodes are learning as well. After finish the learning process of SOM, we calculate the mean and the standard deviation of each node based on the clustered data. The map of SOM can be represented several normal distribution curves as in figure 3.



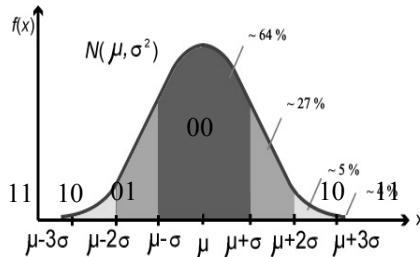
Fig. 3. The normal distribution curve of the SOM

We quantize each cluster (normal distribution curve) into four levels by using 2-bit encoding. Figure 4 shows how to break up the total area under the curve for encoding the input data. Table 1 shows the summarization of 2-bit encoding data. ‘00’ means the data is absolutely normal, ‘01’ means the data is almost normal, ‘10’ means the data is significantly abnormal and ‘11’ means the data is absolutely dangerous to the system.

2.1.2 Abnormal Quantization

Once the SOM is trained and abnormal quantization calculation is obtained, the incoming data can be quantized as follow:

1. For every incoming input, find the winning node of the SOM.
2. Encoding to 2-bit by using mean and standard deviation of winning node.

**Fig. 4.** The total area under the normal distribution curve**Table 1.** The 2-bit encoding data

The level of Abnormal	2 bit - encoding	The range of data
0 - Normal	00	($\mu-1\sigma \leq x \leq \mu+1\sigma$)
1- Minimal	01	($\mu-2\sigma \leq x < \mu-1\sigma$ or $\mu+1\sigma < x \leq \mu+2\sigma$)
2 - Significant	10	($\mu-3\sigma \leq x < \mu-2\sigma$ or $\mu+2\sigma < x \leq \mu+3\sigma$)
3 - Dangerous	11	($x < \mu-3\sigma$ or $x > \mu+3\sigma$)

2.2 Combination Parameters

After the data is encoded every features, we can create the input for XCS by combining every feature into an input string as shown in table 2.

Table 2. Combination of parameters as input for XCS

Parameters	P1	P2	P3	P4	P5	P6	...	P29
Input string	00	01	00	10	00	11	...	00

2.3 XCS

XCS is a Learning Classifier System without internal memory, where the rule-based consists of a number (N) of condition/action rules. The condition is the ternary alphabet: {0, 1, #} and the action is coded as an integer. Associated with each rule are prediction payoff (p), prediction error (ε), fitness parameters (F) and niche size estimate (σ).

On receipt of an input data, the rule-based is scanned, and any rule whose condition matches the message at each position is tagged as a member of the current match set [M]. An action is then chosen from those proposed by the members of the match set and all rules proposing the selected action form an action set [A]. A version of XCS's explore/exploit action selection scheme will be used here. That is, on one cycle an action is chosen at random and on the following the action with highest total payoff is chosen deterministically.

Once the action is selected, the environment returns a reward (R), which is used to update the prediction payoff (p), the prediction error (ε), the niche size estimate and fitness parameters (F) of each member of the current [A] using the Widrow-Hoff delta rule with learning rate β .

XCS employs two discovery mechanisms, a niche genetic algorithm (GA) and a covering operator. XCS uses a time-based mechanism under which each rule maintains a time-stamp of the last system cycle upon which it was consider by the GA. The GA is applied within the [A] when the average number of system cycles since the last GA in the set is over a threshold θ_{GA} . The reader is referred to [9] for a full algorithmic description of XCS.

3 FTP-Only Dataset

We use the 1999 KDD cup intrusion detection dataset [7] has been extensively used in ID research. Four categories of simulated attacks were injected among the normal traffic. In [6] extracted a small subset consisting of FTP control records (port 21 only) from these training and test datasets, which amounted to 798 training instances and 837 test instances. They call it the FTP-only dataset. The datasets are available from the ALAR LAB website (<http://www.itee.adfa.edu.au/~alar/>).

4 Experiments and Discussion

In our experiment, we test our system compare with the XCSR (XCS with real-number) and XCS(FC)0.3 [6] which is the extension of XCSR for IDS. We experiment with 5×1 SOM. The parameter setting of XCS similar to use in Wilson [3] as follows: $\beta=0.2$; $\alpha=0.1$; $\varepsilon_0=10$; $v=5$; $\chi=0.8$, $\mu=0.04$, $\theta_{del}=25$; $\theta_{GA}=25$; $\delta=0.1$ and the maximum population size is $N=2000$.

Table 5. The mean and standard deviation of the accuracy of various ML algorithm with XCS, XCS(FC)0.3 and The proposed system

ML	Normal	Probe	DOS	U2R	R2L	Overall
C4.5	98.36	100.0	89.47	33.33	0.3121	21.51
RF	99.10(0.01)	100.0(0.00)	99.59(0.01)	31.78(0.25)	15.54(0.13)	33.94(0.10)
RT	94.51(0.03)	98.33(0.09)	97.95(0.02)	37.56(0.32)	24.92(0.20)	40.44(0.15)
LMT	97.54	100.0	100.0	86.67	14.04	33.57
NB	92.62	100.0	98.25	60.0	8.89	28.32
BN	94.26	100.0	100.0	93.33	29.02	44.68
Logit	95.90	100.0	100.0	66.67	17.94	35.96
MLP	97.81(0.01)	83.33(0.37)	100.0(0.00)	44.00(0.38)	35.34(0.48)	49.12(0.01)
RBF	94.81(0.03)	100.0(0.00)	96.61(0.01)	86.67(0.00)	10.48(0.04)	30.22(0.03)
SMO	98.36	100.0	100.0	100.0	3.28	25.69
IB1	99.18	100.0	100.0	100.0	11.23	31.90
KSTAR	100.0	100.0	100.0	40.0	18.25	36.32
XCS	95.17(0.01)	35.00(0.26)	99.67(0.01)	91.77(0.11)	35.33(0.15)	49.27(0.11)
XCS (FC)0.3	94.37(0.01)	38.33(0.31)	94.53(0.03)	74.87(0.15)	59.13(0.14)	67.03(0.11)
The proposed system	Pro-94.09(0.02)	80.00(0.40)	98.05(0.40)	48.00(0.09)	83.15(0.15)	85.16(0.11)

We then challenge the proposed system with other twelve ML algorithms. The average accuracy of twelve ML algorithms reported in [6] for FTP-only data set. Table 5 illustrates the result of twelve ML algorithm, two versions of XCS and the proposed system. All twelve algorithms perform better on Probe. Consistent with the previous research [6], the result showed XCS perform poor when dealing with the imbalanced problem. In other influence class R2L, however, the proposed system performs well compared to twelve algorithms and the two versions of XCS.

5 Conclusion and Future Works

In this paper, we introduced the use of SOM improve the performance of XCS for intrusion detection. The proposed system uses SOM as input quantizer by training SOM separately for each feature on normal behavior data and uses their mean and standard deviation quantizes the input.

The proposed system was tested on FTP-only dataset which is a sub-set of 1999 KDD cup intrusion detection dataset. It is shown that the proposed system can improve the accuracy of XCS and also use less than memory in term of macro classifiers. We also compare with twelve ML algorithms and found that the proposed system outperform the accuracy obtained by twelve ML algorithms. We are currently our proposed system to the other intrusion detection domain.

References

1. Debar, H., Dacier, M., Wepshi, A.: A revised taxonomy for intrusion 590 detection systems. Technical report, Computer Science/Ma- 591 thematics (1999)
2. Frank, J.: Artificial Intelligence and Intrusion Detection: Current and future directions. In: Proceedings of the 17th National Computer Security Conference (October 1994)
3. Wilson, S.W.: Classifier Fitness Based on Accuracy. Evolutionary Computation 3(2), 149–176 (1995)
4. Holland, J.H.: Adaptation. In: Rosen, Snell (eds.) Progress in Theoretical Biology (1976)
5. Bull, L. (ed.): Applications of Learning Classifier Systems. Springer, Heidelberg (2004)
6. Shafi, K., Kovacs, T., Abbass, H.A., Zhu, W.: Intrusion detection with evolutionary learning classifier systems. Natural Computing (December 2007)
7. Hettich, S., Bay, S.D.: The UCI KDD Archive (1999), <http://www.kdd.ics.uci.edu>
8. Kohonen, T.: Self-organization and associative memory. Springer, New York (1989)
9. Butz, M., Wilson, S.: An algorithmic descriptionof XCS. In: Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds.) IWLCS 2000. LNCS, vol. 1996, pp. 253–272. Springer, Heidelberg (2001)

New Particle Swarm Optimization Algorithm for Solving Degree Constrained Minimum Spanning Tree Problem

Huynh Thi Thanh Binh and Truong Binh Nguyen

Hanoi University of Technology
Ha Noi, Viet Nam
binhht@it-hut.edu.vn, raylight85@yahoo.com

Abstract. Given a connected, weighted, undirected graph $G=(V, E)$ and a bound d . The Degree-Constrained Minimum Spanning Tree problem (DCMST or d -MST) seeks the spanning tree with smallest weight in which no vertex have degree more than d . This problem is NP-hard with $d \geq 2$. This paper proposes a new Particle Swarm Optimization algorithm for solving the d -MST problem. The proposed algorithm uses some new methods for selecting vector of particles. Results of computational experiments are reported to show the efficiency of the algorithm.

Keywords: degree constrained minimum spanning tree, particle swarm optimization, swarm intelligent, genetic algorithm.

1 Introduction

The Degree-Constrained Minimum Spanning Tree (DCMST or d -MST) or Bounded Degree Minimum Spanning Tree problem is combinatorial optimization problem that appear in many application such as in the design of telecommunication networks and integrated circuits, switch in an actual communication network i.e.

Let $G = (V, E)$ be a connected undirected graph with positive edge weight $w(e)$. d -MST can be formulated as follows:

Minimize

$$W(T) = \sum_{e \in T} w(e)$$

subject to

T is spanning tree of G ,

$\text{degree}(T) \leq d$.

This problem is known to be NP-hard with $d \geq 2$ [6].

2-MST is the Hamilton path problem which is NP-complete. There are several techniques for solving d -MST problem, such as heuristic algorithms and genetic algorithm.

2 Previous Works for Solving the d-MST Problem

There are several techniques for solving *d-MST* problem, such as heuristic algorithms, genetic algorithm. In 1980, Narula and Ho use branch and bound based on Lagrange relaxtion [18] and edge exchange for solving problem. In 1996, Krishnamoorthy, Craig proposed heuristic algorithm based neural network, simulated annealing, greedy algorithms and greedy random algorithms for solving *d-MST* problem.

In the mid of the year 90, genetic algorithm is the main approach method for solving *d-MST* problem. In [2], Zhou and Gen proposed genetic algorithm used Prüfer number encoding. In [4], Knowles and Corne used another encoding by using array with size $n \times (d-1)$ (n : number of vertices, d : degree constrained). In 2000, Raild and Julstrom proposed weighted encoding and edge-set encoding. Experiment showed that edge-set encoding has better result than the other until this year. In 2001, Krishnamoorthy, Earnst and Sharaiha [7] proposed some new heuristics algorithm for solving *d-MST*, genetic algorithm use Prüfer encoding and some simulated annealing. They also proposed exact algorithm for solving *d-MST* with small data set. They use this algorithm for initializing population in large data set. In 2006, Thang and Catherine [9] proposed ant-based algorithm for solving *d-MST* problem.

Until now, particle swarm optimization algorithm for solving *d-MST* problem have not used yet. This paper will propose new particle swarm optimization algorithm for solving *d-MST*.

3 A New Particles Swarm Optimization Algorithm

We propose a new *PSO* algorithm for solving *d-MST* problem called *PSO-mst*. In this section, we focus on the design of *PSO* for solving the *d-MST* problem.

3.1 Initialization

Each particle of the swarm is a degree constrained spanning tree (*d-ST*). In order to create only feasible solutions for the initial swarm of the *PSO*, we altered some algorithms which solve the MST problem such as Kruskal's or Prim's called Kruskal'MST and Prim's MST. When creating a *d-ST*, an edge after the other is then checked in the predetermined order for an eventual inclusion in the spanning tree. An edge is included in the *d-ST* if it does not violate the degree constraint in its vertices and these vertices are not yet connected via other edges in *d-ST* (to ensure no cycles is contained).

Another method for initializing the swarm is using the depth first search algorithm. This method is called *InitSearch*. This algorithm also creates a *d-ST*. The candidate edge is based on two criterions: weight and randomize. S is the set of the vertexes which is included in the under-constructed *d-ST*. The algorithm will select a random vertex and find the minimum weighted edge which has one vertex is the selected vertex, the other one is in the rest of S . This edge is included in the *d-ST* if it

does not violate the degree constraint in its vertices (no cycles is introduced). The procedure terminates when all the vertexes have been selected (all vertexes of the graph have been in under-constructed d -ST). InitSearch algorithm will be hoped making a small weight tree and divert swarm.

3.2 Velocity

This paper proposes a new method for the movement of the particles. In the PSO model, each particle bases on three positions to select velocity to move from its current position to a new position. They are: its current position, its best position and best position of its swarm.

In the d -MST problem, each particle is a d -ST. The current position of a particle is its d -ST. The best position is the smallest weighted d -ST which the particle has been, and the best position of the swarm is the smallest weighted d -ST (the smallest weighted particle) of the swarm.

Each particle has three new positions:

- *Move to new position by itself: p_1*

We'll alter the current position (d -ST) of the particle to a new d -ST by using LocalSearch or GreedyExchange procedures.

T is a d -ST of G . $L = E \setminus T$ is the set of edges which is of G but not in T and sorted increasingly by weight. The LocalSearch procedure will insert an edge of L in T and delete the highest appropriate edge in the cycle which has just introduced.

The GreedyExchange procedure will exchange the greatest weighted edge in T by a random edge or smallest edge in L which not violates the constrained to get a new d -ST.

- *Move to new position by learning its current position and its best position: p_2*

We proposed a procedure (called Recombine) to recombine two d -ST. First d -ST is the current position of the particle; second d -ST is the best position which it explored. The result of recombining these positions is the new position of the particle. The output d -ST of the procedure has edges which both are in two d -ST above. We hope that may be these edges will be included in the minimum d -MST we've been finding.

- *Move to new position by learning its current position, its best position and the best position of the swarm : p_3*

We also use a recombine procedure to recombine three d -ST. First d -ST is the current position of the particle; second d -ST is the best position of this particle; and third is the best position of the swarm. The result of recombining these positions is the new position of the particle.

According to examine the candidate edges to exchange or rebuild the d -ST in the procedures above (LocalSearch, GreedyExchange and Recombine), two criterions

are proposed: examining edges by random or by their weight, we have more procedures to make the swarm more varied and not easy to fall into a local minimum's position.

3.3 PSO-mst Algorithm

PSO-mst algorithm for solving *d-MST* problem is proposed can be explained as below:

```

End case
End do

<Update  $p_i$ best>
...
<Update  $g$ best>
...
End While

End Procedure

```

4 Experimental Result

4.1 Experiment Description

The data set is used in this paper has been used in previous papers for solving d -MST problem. Two set of instances are Euclidean and Non-Euclidean graph are used in this paper. Data set can be downloaded from webpage: <http://cs.hbg.psu.edu/~bui/data/SHRD-Graphs.zip>

Parameters used in algorithms:

- Swarm size (10 – 50)
- Number of loop (10)
- Randomize parameters (*choice* ...)

We have experimented with more value of parameters choice to get better. In this paper, we choose parameter *choice* is random in [0...9]. The movement of the particle is chosen based on parameter *choice*. If *choice* < 7, the particle will move to p_1 ; if $6 < \text{choice} < 8$, the particle will move to p_2 , else ($7 < \text{choice} < 9$), the particle will move to p_3 .

4.2 Results of Computational Experiments

Tables 1–5 show some result of PSO-mst algorithm on CRD, SHRD, SYM, STR and RANDOM instances.

N: number of vertices; **d:** Degree-constraint; **Np:** number of particles; **Prev. Best:** the best weight of tree obtained by all the previous algorithms; **Best.** the best weight of tree obtained by the algorithm proposed in this paper; **Avg.** the average weight of tree obtained after 50 runs; **Std. Dev.** standard deviation.

(*) Note means the best result found by PSO-mst algorithm is better than the previous algorithms for solving d -MST problem.

(-) Note means that the test in this paper is the first time for solving these instances.

Table 1. Result of PSO-mst on CRD instances

	n	d	Prev Best	PSO			
				Best	Gain(%)	Avg	Std Dev
CRD 100	100	2	7524*	7204.34	4.25	7375.43	54.90
	100	3	6199	6199.05	0	6199.05	0.00
	100	4	6197	6197.57	0	6197.57	0.00
	100	5	6197	6197.57	0	6197.57	0.00
CRD 501	50	2	5625*	5605.6	0.34	5732	27.93
	50	3	5130	5130.3	0	5130.3	0.00
	50	4	5130	5130.3	0	5130.3	0.00
	50	5	5130	5130.3	0	5130.3	0.00
CRD 700	70	2	6544*	6448.72	1.46	6628	73.46
	70	3	5789	5789.55	0	5789.55	0.00
	70	4	5789	5789.55	0	5789.55	0.00
	70	5	5789	5789.55	0	5789.55	0.00

Table 2. Result of PSO-mst on SHRD instances

	n	d	Prev Best	PSO			
				Best	Gain(%)	Avg	Std Dev
SHRD 159	15	2	906	904	0.22	906.68	1.98
	15	3	597	597	0	597	0.00
	15	4	430	430	0	430	0.00
	15	5	332	332	0	332.02	0.15
SHRD200	20	2	1873	1679	10.36	1681.55	2.08
	20	3	1100	1088	1.09	1088	0.00
	20	4	829	802	3.26	802.33	0.15
	20	5	638	627	1.72	627.19	0.61
SHRD 259	25	2	2984	2714	9.05	2720.64	4.75
	25	3	1870	1756	6.1	1756.68	2.50
	25	4	1312	1292	1.52	1292.34	0.71
	25	5	1019	1016	0.29	1016	0.00

Table 2. (*continued*)

	n	d	Prev Best	PSO			
				Best	Gain(%)	Avg	Std Dev
SHRD 300	30	2	4560	3992	12.46	4005.43	16.97
	30	3	2738	2592	5.33	2595.56	12.42
	30	4	1965	1905	3.05	1905.80	2.60
	30	5	1526	1504	1.44	1504	0.00

Table 3. Result of PSO-mst on SYM instances

	n	d	Prev Best	PSO			
				Best	Gain(%)	Avg	Std Dev
SYM 500	50	2	2522*	1802	28.55	1903.86	68.79
	50	3	1156	1156	0	1157.75	3.35
	50	4	1105	1105	0	1105	0.00
	50	5	1098	1098	0	1098	0.00
SYM 701	70	2	2908*	2008	30.95	2198.24	84.08
	70	3	1270	1270	0	1270	0.00
	70	4	1198	1198	0	1198	0.00
	70	5	1186	1186	0	1186	0.00

Table 4. Result of PSO-mst on STR instances

	n	d	Prev Best	PSO			
				Best	Gain(%)	Avg	Std Dev
STR 100	100	2	5211*	5021	3.65	5059.13	19.71
	100	3	4702	4702	0	4702.1	0.30
	100	4	4546	4546	0	4546	0.00
	100	5	4403	4403	0	4403	0.00
STR 500	50	2	4471*	4420	1.14	4439.09	9.88
	50	3	4128	4118	0.24	4118	0.00
	50	4	3962	3956	0.15	3956	0.00
	50	5	3807	3807	0	3807	0.00

Table 4. (continued)

	n	d	Prev Best	PSO			
				Best	Gain(%)	Avg	Std Dev
STR 700	70	2	4727*	4734	-0.17	4766.85	6.53
	70	3	4397	4397	0	4397	0.00
	70	4	4249	4249	0	4249	0.00
	70	5	4100	4100	0	4100	0.00

Table 5. Result of PSO-mst on RANDOM instances

	n	d	Prev Best	PSO			
				Best	Gain(%)	Avg	Std Dev
RAND 200	200	2	-	2180.74	-	2218.74	17.17
	200	3	-	1909.66	-	1909.66	0
	200	4	-	1908.89	-	1908.89	0
	200	5	-	1908.89	-	1908.89	0

Experiment results show that:

- With a Euclidean graph (CRD, SYM, STR and RANDOM), with the bound $d = 3, 4, 5$, the PSO-mst's result always are the best solutions. With $d=2$, the result of PSO-mst are better than the best result obtained by previous algorithms.
- With non-Euclidean graph (SHRD), with $d=2, 3, 4, 5$, the result of PSO-mst are also better than the best result obtained by previous algorithms.

5 Conclusion

This paper proposes new particle swarm optimization for solving d -MST problem. The experimental result shows the efficiency of the algorithm on the instances which are used in previous algorithms for solving this problem.

Acknowledgment. We would like to thank Prof. Thang N.Bui, Pennsylvan State University and Prof. Gunther Raidl – Vienna University of Technology for providing us the d -MST problem instances which used in this paper, as well as sending us the materials related to their works on d -MST problems.

References

- Goldberg, E.F.G., de Souza, G.R., Goldberg, C.: Particle Swarm Optimization for the Bi-objective Degree constrained Minimum Spanning Tree. IEEE Congress on Evolutionary Computation, 1527–1534 (2006)

2. Zhou, G., Gen, M.: Approach to degree-constrained minimum spanning tree problem using genetic algorithm. *Engineering Design & Automation* 3(2), 157–165 (1997)
3. Gottlieb, J., Julstrom, B.A., Raidl, G.R., Rothlauf, F.: Prüfer numbers: A poor representation of spanning trees for evolutionary search (2000)
4. Knowles, J., Corne, D.: A New Evolutionary Approach to the Degree-Constrained Minimum Spanning Tree Problem. *IEEE Trans. on Evolutionary Computation* 4(2), 125–134 (2000)
5. Krishnamoorthy, M., Ernst, A.T., Sharaiha, Y.M.: Comparison of Algorithms for the Degree Constrained Minimum Spanning Tree. *Journal of Heuristics* 7, 587–611 (2001)
6. Hanr, L., Wang, Y.: A Novel Genetic Algorithm for Degree-Constrained Minimum Spanning Tree Problem. *IJCSNS International Journal of Computer Science and Network Security* 6(7A) (2006)
7. Krishnamoorthy, M., Craig, G., Palaniswami, M.: Comparison of heuristic algorithms for the degree constrained minimum spanning tree. In: Osman, I.H., Kelly, J.P. (eds.) *Metaheuristics: Theory and Applications*, pp. 83–96 (1996)
8. Hansen, P., Mladenović, N.: An Introduction to variable neighborhood search. In: Vos, S., Martello, S., Osman, I.H., Roucairol, C. (eds.) *Metaheuristic: Advances and trends in local Search Procedures for Optimization*. Kluwer, Dordrecht (1999)
9. Bui, T.N., Zrncic, C.M.: An Ant-Based Algorithm for Finding Degree-Constrained Minimum Spanning Tree. In: *GECCO 2006* (2006)

Continuous Pitch Contour as an Improvement Feature for Music Information Retrieval by Humming/Singing

Tri Nguyen Truong Duc, Minh Le Nhat, Ha Nguyen Duc Hoang, and Quan Vu Hai

Faculty of Information Technology

University of Natural Sciences, VNU-HCMC, Vietnam

{ntdtri, lnhatminh}@gmail.com, {ndhha, vhquan}@fit.hcmuns.edu.vn

Abstract. In this paper we present a method for smoothing the pitch sequence to remove the outliers caused by pitch tracking method and errors in sung/hummed voice. This approach is used for constructing a query by singing/humming system. After the pitch sequence is smoothed, the continuous pitch contour is calculated. Then, this feature can be used with Dynamic Time Warping (DTW) matching to compute the difference score between the each query and song. Experimental result of this method on TCS Corpus for query by Singing/Humming will be reported and discussed in detail in this paper.

Keywords: music information retrieval (MIR), music query, melodic matching, query by sample, query by humming/singing.

1 Introduction

Currently, query by singing/humming is becoming a very common and popular research. There have been many techniques proposed for constructing a query by singing/humming system. There are two types of QBSH methods:

- The note-based methods, which use note segmentation and string alignment algorithm [1, 2].
- The continuous-pitch-sequence-based approaches, which do not need to segment the pitch sequence into notes and Dynamic Time Warping, or Linear Scaling [3] methods for matching.

In this paper we present a method for smoothing the pitch sequence. After applying the smoothing method, the feature continuous pitch contour can be extracted. This feature then is used with DTW matching to calculate the difference score between 2 features.

Section 2 deals with the continuous pitch contour as melody feature, which is an improvement for the system performance. The experiment and results on this system are reported and discussed in details in section 3.

2 Continuous Pitch Contour

The user's query is recorded and the samples are divided into frames. The pitch sequence is extracted by using Autocorrelation method [1]. The raw pitch sequence is not very effectively used with DTW algorithm because there are a lot of outliers in the

data due to inevitable errors from users and the pitch tracking methods. Moreover, the pitch sequence cannot deal with key transposition issue which happens very usually when the user sing the melody in the key different from one in the database. Thus, we proposed smoothing the pitch sequence and using continuous pitch contour (CPC) as the melodic feature. The method to compute this feature is presented in this section.

2.1 Outlier Removing

After using the pitch tracking method, we have the pitch sequence of an audio query. This sequence always has some outliers due to the errors in the method and the user's query. The four-step approach described below can be used to remove these unavoidable outliers.

- i. The silent segment at the beginning (pitch = 0) is removed.
- ii. A new pitch change is established whenever the difference between current average pitch segment and the next pitch is greater than a threshold T (semitones). Then, all the pitch values in the segment are replaced by the average value inside that segment. (In the following experiment, we use $T = 0.7$ semitone)
- iii. If a music segment has duration less than 0.1s, it is merged to its neighboring segment which is closer in value.
- iv. If a silence segment is less than 0.3s, it will be replaced by the previous segment average value.

```
Program SmoothPitchSequence()
    PitSeg = Ø;
    For i = 1..PitArr.Length
        if (|Average(PitSeg, PitArr[i]) - PitArr[i]| > T)
            Note ← (Average(PitSeg), PitSeg.Length)
            PitSeg = Ø;
        endif;
        PitSeg ← PitArr[i];
    EndFor;

    {Combine notes if the contiguous difference < 0.5}
    For i = 1..Note.Length - 1
        if (|Note[i] - Note[i + 1] < 0.5)
            Combine(Note[i], Note[i + 1]);
    EndFor

    For i = 1..Note.Length - 1
        if (Note[i].Length < 0.1s)
            Combine Note[i] with the closest contiguous note
    EndFor

    For i = 1..Note.Length
        if (Note[i].Pitch = 0 & Note[i].Length < 0.3s)
            Combine Note[i] with Note[i-1]
    EndFor
End.
```

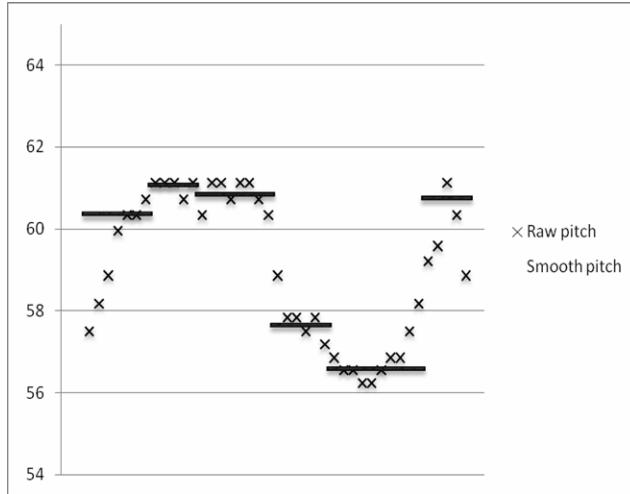


Fig. 1. The pitch sequence before (raw pitch) and after (smooth pitch) removing the outliers

Fig. 1 shows the raw pitch sequence of a sung query as the x-mark. We can see that there are many outliers in the sequence, which can affect the recognition rate dramatically. The pitch sequence is segmented and all the values in a segment will be replaced by the average value of that segment. The result is also shown in Fig. 1 with the pitch lines, each line represents a pitch segment and its average value.

2.2 Continuous Pitch Contour Feature

As presented in the previous section, the audio query is extracted into a sequence of pitches, called continuous pitch. The silent points are removed from the continuous pitch array. Then, the continuous pitch contour is defined as the array of differences between a pitch and its previous pitch (see the following formula):

$$CPC[i] = \text{pitch}[i] - \text{pitch}[i-1], i = 1..n-1 \quad (1)$$

In the previous formula, we assumed that the length of pitch sequence (numbered from 0) is n. Therefore, the CPC length is n-1.

The pitch contour is useful when dealing with key transposition in hummed/sung queries. User may not sing or hum the melody in its original key due to the different pitch range in each person's voice. Plus, according to [4], melodic contour, the difference between contiguous notes, is more easily remembered than an exact melody.

3 Experimental Results

We have conducted an experiment using this system. This section presents the data corpuses which are used in the experiment and discusses the result.

3.1 Corpus Data

We conducted an experiment on this system with the TCS Corpus for Query by Singing/Humming [5]. This corpus data include 200 midi files represents the melodies from 158 songs of Trinh Cong Son, Vietnamese composer. One song may be divided into many themes to deal with the fact that users may not sing from the beginning of the song, but from the beginning of a theme. These melodies here are a little more complicated than that in QBSH. These melodies, however, are still popular with many Vietnamese people. There are also 292 recorded queries (59 humming files and 233 singing files) which were collected from 5 people (3 males and 2 females), none of who have had prior official music training.

3.2 Evaluation

The evaluation method for the performance of a music retrieval system, in general, or a matching method, particularly is MRRR [2] (Mean Reciprocal Right Rank), which is calculated as in the following equation.

$$MRRR = \frac{\sum_{n=1}^{NumberOfQueries} \frac{1}{RightRank_n}}{NumberOfQueries} \quad (2)$$

3.3 Experimental Set-Up

The songs database was all MIDI-format files. Thus, the continuous pitch contours can be extracted easily and automatically by reading the MIDI file information.

All the queries were automatically converted to 12,000-samples-per-second and 8-bit-per-sample files. Then, the CPC features were extracted from the queries by using the method described above. The window size used for pitch extraction in midi files and audio queries was 32 milliseconds.

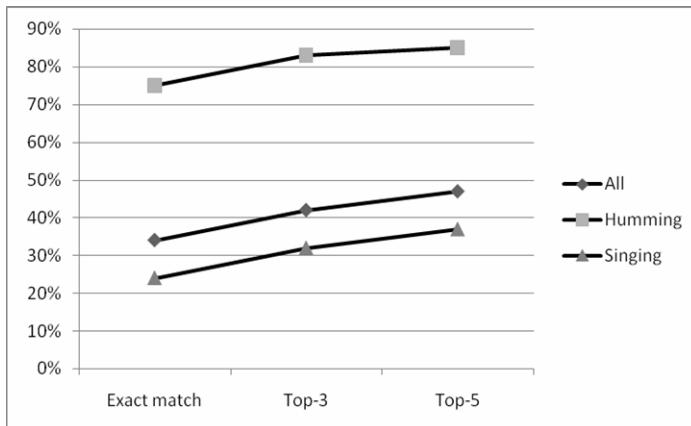
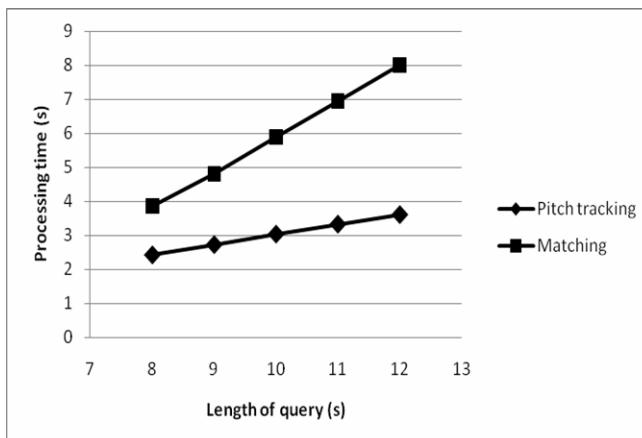
The extracted feature of each query, then, was used in DTW method to calculate the matching score between it and all songs in the database.

3.4 Results

First, we conducted the experiment without removing the outliers in the pitch sequence. The CFC is still calculated as in Equation 1. With this approach, we attained the MRRR result equal to 0.068. When the removing outlier method is used, the MRRR result on TCS Corpus increased to 0.408

We invested the result in more details. The humming MRRR was 0.794, while the singing MRRR was 0.310. Fig. 2 shows the percentage of right rank songs returned in the top-5, top-3, and the exact match in the ranked list. Accordingly, it can be assumed that this method works better on humming voice. That is understandable because frequency of humming voice is more stable than the singing voice.

The average responding time values of pitch tracking module and matching module are shown in Fig. 3. According to the chart, the complexity of the method is linear. The average elapsed time to retrieve 10-second query is 8.7s.

**Fig. 2.** Ranking result overview for TCS Corpus**Fig. 3.** Processing time of pitch tracking and matching module

4 Conclusion

Errors in singing/humming query from users and the pitch tracking methods are unavoidable and must be handled by the query system. In this paper, we presented a smoothing method which is used to remove the outliers in the query pitch sequence. This approach can reduce the errors made by users and the extracting feature module. The continuous pitch contour feature can deal well with the key transposition problem. The experimental result on this data set showed that the smoothing method works well with continuous pitch contour feature and DTW matching method.

References

1. Ghias, A., Logan, J., Chamberlin, D., Smith, B.C.: Query By Humming - Musical Information Retrieval in An Audio Database. In: Proceedings of ACM Multimedia 1995, pp. 231–236 (November 1995)
2. Pardo, B., Shifrin, J., Birming-ham, W.: Name that tunes: A Pilot Study in Finding a Melody from a Sung Query. Journal of the American Society for Information and Technology (2004)
3. Jang, R.: Audio signal processing and recognition,
<http://neural.cs.nthu.edu.tw/jang/books/audioSignalProcessing/>
4. Chai, W.: Melody as a Significant Musical Feature in Repertory Classification. In: Music Query: Methods, Models, and User Studies (Computing in Musicology 13), CCARH. MIT Press, Cambridge (2004)
5. SLP Group, HCM-US. Vietnam, Trinh Cong Son Corpus for Query by Singing/Humming, available at SLP Group homepage, <http://services.fit.hcmuns.edu.vn/slp/>

Classification Using Improved Hybrid Wavelet Neural Networks

Nhu Khue Vuong¹, Yi Zhi Zhao², and Xiang Li²

¹ School of Computer Engineering, Nanyang Technological University, Singapore 639798

² Singapore Institute of Manufacturing Technology, 71 Nanyang Drive, Singapore 638075

vuon0002@ntu.edu.sg, yzzhao@simtech.a-star.edu.sg,
xli@simtech.a-star.edu.sg

Abstract. In this study, we propose a novel neural net-based classifier called improved Hybrid Wavelet Neural Networks (iHWNN). iHWNN makes good use of the characteristics of Wavelet Neural Networks (WNN) and Back Propagation Neural Networks (BPN), so that it inherits WNN's capability in learning efficiency and BPN's applicability in handling problems of large dimensions. To show the advantages of the developed algorithm, we compare its performance with those from existing classifier systems on several applications. Comparable results are achieved over several datasets from the UCI Machine Learning, with an average increase in accuracy from 91.69% for classification-based objective functions training to 94.17% using optimized iHWNN networks.

Keywords: Wavelet Neural Networks, Back Propagation Neural Networks.

1 Introduction

Classification problems have a long history in the machine learning literature. Neural networks have great advantages of nonlinear input-output mapping capability; therefore, neural networks learning is a potential and promising technique to deal with classification problems. Several researches also showed that Wavelet Neural Networks possesses high forecasting accuracy than traditional predicting methods; however, it suffers from the “curse of dimensionality” and therefore its applications are limited to the case of small dimensions [1-3]. In spite of some deficiencies such as slow convergence speed and getting local minimum value, Back Propagation Neural Networks can be used for handling problems of large dimensions [4]. Therefore, combining WNN with BPN can hopefully remedy the weakness of each other, resulting in an efficient artificial neural networks algorithm having the capabilities of handling problems of moderate or even large dimension. Technically speaking, the idea of combining both WNN and BPN is feasible because of the similarity between WNN decomposition and one-hidden-layer BPN [4]. Inspired by that investigation, we propose a novel single neural net-based algorithm called improved Hybrid Wavelet Neural Networks. iHWNN makes good use of the highly efficient characteristics of WNN and the scalar properties of BPN; hence, it is quite capable of mapping non-linear input-output, and has a great potential and promise in dealing with classification as well as prediction problems. For details about WNN and BPN, please refer to [1-4].

2 Improved Hybrid Wavelet Neural Networks

2.1 Characteristics of Improved Hybrid Wavelet Neural Networks

An improved hybrid wavelet neural network is a sort of WNN with relatively simple modifications based on the Back-propagation algorithm. The first modification is the addition of biases in the input and hidden layers. And the second one is the usage of sigmoid function as the activation function at output neurons. The explanation for the two modifications was represented in below.

Firstly, it is common knowledge that adding the bias in BPN is to prevent the net from stopping learning in case all values of an input pattern are zero. If all values of an input pattern are zero, the weights in weight matrix would never change and the net could not continue to learn. Hence, a "pseudo input" or bias with a constant output value of one is created. By sending a constant output of one from input neurons to hidden neurons and from hidden neurons to output neurons, it is guaranteed that the input values of the hidden and output neurons are always different from zero.

Secondly, WNN uses linear activation function ($f(x) = x$) as the activation function of output neurons. Despite of being continuous, differentiable, and monotonically non-decreasing, it is not saturate, i.e., approach finite maximum and minimum values asymptotically. Therefore, it does not clamp the signals within a specified range.

2.2 Training Algorithm of iHWNN

2.2.1 Wavelet Basis Functions

We favored the Morlet wavelet as the basis function in our applications because of its high resolution in both time and frequency domains. Therefore, it is applicable for the developed algorithm to use the type of Morlet wavelet as follows:

$$\psi(t) = \cos(1.75t) \exp(-t^2/2) \quad (1)$$

2.2.2 Initialization of Improved Hybrid Wavelet Neural Networks

To improve the training efficiency, we adopted a simple initialization procedure presented in [5]. We denote $[x_{iMin}, x_{iMax}]$ to be the domain containing the input vector x_i . Let t^* and $\Delta\psi$ be the center and the radius of the Morlet mother wavelet. In order for the wavelet $\psi_{a_h b_h}$ to be able to cover the input space, the dilation and translation parameters can be initialized to:

$$a_h = \frac{1}{2\Delta\psi} \sum_{i=1}^I v_{ih} (x_{iMax} - x_{iMin}) \quad (2)$$

$$b_h = \frac{1}{2\Delta\psi} \left(\sum_{i=1}^I v_{ih} x_{iMax} (\Delta\psi - t^*) + \sum_{i=1}^I v_{ih} x_{iMin} (\Delta\psi + t^*) \right) \quad (3)$$

For more detailed discussion, please refer to [3].

2.2.3 Extra Momentum Coefficient

To solve BPN's problem of getting local minimum, iHWNN uses an extra momentum term in its learning algorithm so that it can stabilize the learning procedure and speed

up the convergence. In learning algorithms of WNN and BPN [1-4], the formulae of parameter adjustment can be simplified as:

$$P_r = P_{r-1} - \eta \frac{\partial MSE}{\partial P_{r-1}} + \alpha \Delta P_{r-1} \quad (4)$$

where P represents the vectors of connective weights, or the dilation matrix, or the translation matrix and ΔP is their increments. It is known that the momentum coefficient tries to keep the process of parameter adjustment moving and therefore not to get stuck in the local minimum. In fact, assume that a local minimum is met during the learning, it will lead to

$$\frac{\partial MSE}{\partial P_{r-1}} = -E_r = 0 \quad (5)$$

the update of P will be stopped at this point. Therefore, the adjustment process can escape from the local minimum if ΔP_{r-1} is not zero. Motivated by this idea, the adjustment process of network parameters in iHWNN is described by

$$P_r = P_{r-1} - \eta \frac{\partial MSE}{\partial P_{r-1}} + \alpha_1 \Delta P_{r-1} + \alpha_2 \Delta P_{r-2} \quad (6)$$

where α_1 and α_2 are the two momentum terms of the algorithm. This indicates that the variable change not only depends on the gradient, the variable change at the iteration $(r-1)^{th}$ but also the variable change at the iteration $(r-2)^{th}$. The proposed iHWNN approach, however, is more likely to escape from a local minimum than BPN and WNN. From (9), we see that the variable change is stuck only when E_r , ΔP_{r-1} , as well as ΔP_{r-2} are all equal to zero. Nevertheless, the probability of such a case is little in a local minimum. With this structure, the algorithm may avoid local minimum or jump out of it in case such a situation happens.

2.2.4 Training Algorithm

Since the iHWNN is derived from a feed-forward neural network, we use back-propagation method to train this network. The algorithm is presented Fig. 1. For more details, please refer to [3].

```

1 begin
2   initialize network parameters
3   do
4     supply training sample set
5     self-train the network
6     compute the mean square error
7     if the computed error is not less than ε
8       then
9         compute gradient vectors
10        modify network parameters
11 until the computed error is less than ε
12 end

```

Fig. 1. iHWNN Training Algorithm

3 Experimental Result and Discussion

3.1 Data Description

Four well-known and most common classification problems were selected from the UCI Machine Learning Data Repository [6]. The selected benchmark datasets include Iris (IR), Pima Indian Diabetes (PD), Breast Cancer (BC), and Wine (WI). Table 1 presents the abstract of those selected datasets.

Table 1. Brief Summary of Datasets Used for Experiments. **N** indicates the number of instances; **m** is the number of attributes; **C** is the number of classes.

Dataset	N	m	C
IR	150	4	3
PD	768	8	2
BC	683	9	2
WI	178	13	3

3.2 Results

3.2.1 Optimal Network Parameters

From the software, we found out the best structural configuration for each dataset. The results were tabulated in Table 2.

Table 2. Optimal Network Parameters with iHWNN Algorithm

Dataset	Network Topology	Learning Rate	Momentum 1 (α_1)	Momentum 2 (α_2)
IR	4-7-1	0.6	0.88	0.03
PD	8-12-1	0.5	0.7	0.01
BC	9-10-1	0.4	0.97	0.01
WI	13-16-1	0.8	0.97	0.01

3.2.2 Classification Accuracy or Confidence Level

In order to collect the results for research, each of the studied dataset was trained and tested 10 times (trials) of 10-fold cross validation because it was the standard procedure that other peer methods (SONG, HBN, and CB) [7-9] used. This time, the experiments were conducted on the optimized network structure and parameters that we obtained for each dataset. The termination conditions that we used for testing and training Iris, Breast Cancer, Diabetes, and Wine datasets, were (MSE = 0.0019), (MSE = 0.005), (MSE = 0.034), and (MSE = 0.0007) respectively. These MSE values were chosen from the empirical results of some preliminary tests.

Next, we present the classification accuracy that the proposed algorithms achieved for each dataset. In Table 3, we can see the comparisons of the proposed algorithm with other more sophisticated MCS. The last row represented the average classification accuracy and standard deviation of the different algorithms being compared over the four datasets used in this study.

Table 3. Comparison of iHWNN with Other Existing Classification Models

Dataset	iHWNN	SONG	HNB	CB
IR	98.67 \pm 2.68	96.20 ± 6.04	94.00 ± 2.0	95.37 ± 5.25
PD	80.72 \pm 4.37	74.90 ± 7.05	76.04 ± 1.5	76.82 ± 6.46
BC	98.02 \pm 1.32	97.00 ± 2.41	97.36 ± 0.6	97.36 ± 1.81
WI	99.27 \pm 1.90	97.60 ± 4.41	98.86 ± 0.8	97.19 ± 3.47
Average	94.17 \pm 2.57	91.43 ± 4.98	91.57 ± 1.23	91.69 ± 4.25

It is clear that over four datasets, iHWNN leads to more accurate or comparable classifiers than other MCS. From Table 3, we can see that the average increase in classification accuracy is from 91.69% for CB training (the most accurate method among 5 methods being compared) to 94.17% for iHWNN training, or a 2.48% decrease in error. An overall decrease in standard deviation also indicates that iHWNN training is more robust and consistent to initial parameter values and pattern variance than SONG and CB algorithms. Though iHWNN showed overall increases in standard deviation over HNB training, the average increases in classification accuracy almost doubled the overall increases in standard deviation. Still, iHWNN is much better than HNB.

Table 4. Unpaired T-test Results of iHWNN

Dataset	Compare iHWNN with	t-value	Degree of Freedom	p-value	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
					Lower	Upper		
IR	<i>SONG</i>	4.5781	298	<0.0001	2.47	0.54	1.4082	3.5318
PD	<i>CB</i>	13.8577	1534	<0.0001	3.9	0.281	3.3471	4.4529
BC	<i>HNB</i>	11.8959	1364	<0.0001	0.66	0.055	0.551	0.769
WI	<i>HNB</i>	2.6534	354	0.0083	0.41	0.155	0.1061	0.7139

For more in-depth analysis, we conducted a series of independence unpaired student T-test to examine whether the improved classification accuracy obtained by iHWNN is significantly better than the one from other three models (SONG, HNB, and CB). The detailed results of the student t-tests were tabulated in Table 4. iHWNN was respectively compared with the classifier which achieved the highest classification accuracy for each dataset among the three models (SONG, HNB, and CB). For example, iHWNN was compared with SONG on the Iris problem, but it was compared with CB on the Pima Diabetes problem, and so forth. The results from these statistical tests showed that at 95% confidence level, iHWNN is extremely significantly better than the other three classifiers for the Iris, Pima Diabetes, and Breast Cancer problems. For the Wine dataset, the difference between the classification accuracies of iHWNN and HNB algorithms is considered to be very statistically significant.

In brief, iHWNN is better than other MCS in terms of the classification accuracy and robustness.

4 Conclusion

In this study, we proposed a new neural net-based classifier based on iHWNN algorithm and evaluated its performance in terms of classification accuracy using benchmark data. Experimental results showed that iHWNN achieved higher classification accuracy than other existing classifier systems on several applications. On UCI MLDR problems, there was an average increase in accuracy from 91.69% for CB training to 94.17% for optimized iHWNN networks training performing 10-fold stratified cross-validation. The results from our study also demonstrated that iHWNN is more robust than CB and SONG. Hence, it is a useful and practical tool for data mining applications.

References

1. Zhang, X.: Prediction of Programmed-temperature Retention Values of Naphthas by Wavelet Neural Networks. *Journal of Computers and Chemistry* 25, 125–133 (2001)
2. Cui, W.Z., Zhu, C.C., Zhao, H.P.: Prediction of Thin Film Thickness of Field Emission Using Wavelet Neural Networks. *Thin Solid Films* 473, 224–229 (2005)
3. Zhao, Y.Z., Vuong, N.K., Li, X.: A Novel Hybrid Wavelet Neural Networks and Its Performance Evaluation. In: IEEE Conf. on Industrial Informatics, pp. 1097–1102 (2008)
4. Tan, Y., Dang, X., Liang, F., Su, C.Y.: Dynamic Wavelet Neural Network for Non-linear Dynamic System Identification. In: IEEE Conf. on Ctrl. Applications, pp. 214–219 (2000)
5. Zhou, B., Shi, A., Cai, F., Zhang, Y.: Wavelet Neural Networks for Non-linear Time Series Analysis. LNCS, vol. 3147, pp. 430–435. Springer, Heidelberg (2004)
6. Blake, C., Merz, C.: <http://www.ics.uci.edu/~mlearn/MLRepository.html>
7. Inoue, H., Nrihisa, H.: Self-organizing Neural Grove: Effective Multiple Classifier System with Pruned Self-generating Neural Trees. In: IEEE International Symposium on Circuits and Systems, pp. 2502–2505. IEEE Press, New York (2005)
8. Langseth, H., Nielsen, T.D.: Classification Using Hierarchical Naive Bayes Models. *Journal of Machine Learning* 63, 135–159 (2006)
9. Rimer, M., Martinez, T.: Classification-based Objective Functions. *Journal of Machine Learning* 63, 183–205 (2006)

Online Classifier Considering the Importance of Attributes

Hiroaki Ueda¹, Yo Nasu², Yuki Mikura¹, and Kenichi Takahashi¹

¹ Graduate School of Information Sciences, Hiroshima City University

² Hitachi Electronics Services Co., Ltd.

{ueda,takahasi}@hiroshima-cu.ac.jp,

{nasu,mikura}@rea.its.hiroshima-cu.ac.jp

Abstract. We propose a new classifier ARTMAP2-AW based on adaptive resonance theory. ARTMAP2-AW evaluates the degree of importance of each attribute, and on the basis of the importance, attributes irrelevant to classification are detected for efficient learning. Experimental results show that ARTMAP2-AW acquires better classification rules than well-known classifiers.

1 Introduction

Classification is a fundamental technique in machine learning and many classifiers have been proposed [1,2,3,4,5,6,7]. One of classifiers that learn classification rules incrementally is fuzzy ARTMAP [2,3] that is based on ART (Adaptive Resonance Theory) [8]. However, fuzzy ARTMAP has some drawbacks, e.g., it has no operations to generalize rules and does not take account of the importance of attributes characterizing data. In order to overcome the drawbacks, we propose a new classifier, ARTMAP2-AW.

In ARTMAP2-AW, a cluster is defined as a hyper-ellipse, so that it can take account of the importance of attributes. In addition, ARTMAP2-AW incorporates not only an operation to specialize classification rules but also an operation to generalize rules. By the operations, ARTMAP2-AW evaluates the degree of importance of each attribute and it detects attributes irrelevant to classification. ARTMAP2-AW has been implemented and experimental results for some data sets are shown.

This paper is organized as follows. In the next section, we present a new classifier ARTMAP2-AW. Some experimental results are shown in section 3. And finally, section 4 concludes the paper.

2 ARTMAP2 Considering Attribute Weights

2.1 The Learning Mechanism of ARTMAP2-AW

In this section, we propose a new classifier *ARTMAP2-AW* (ARTMAP2 considering Attribute Weights). An input of ARTMAP2-AW is a case $I=(X, y)$.

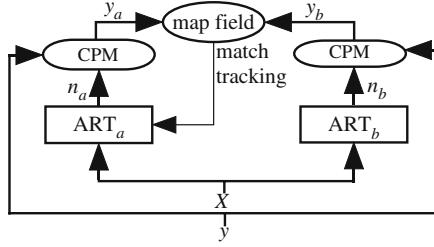


Fig. 1. Architecture of ARTMAP2-AW

$X = (x_1, x_2, \dots, x_M)$ ($x_m \in [0, 1]$) is an attribute vector and M is the number of attributes. y is a class. Possible values of y are defined as a set of symbolic values $S = \{s_1, s_2, \dots, s_L\}$. Figure 1 shows architecture of ARTMAP2-AW. It consists of two ART modules, ART_a and ART_b . ART_a categorizes X into a category n_a and by (1) a class prediction module (CPM) evaluates the most frequent class y_a in n_a . $freq(n_a, s_l)$ is the number of cases that are categorized into n_a and belong to s_l .

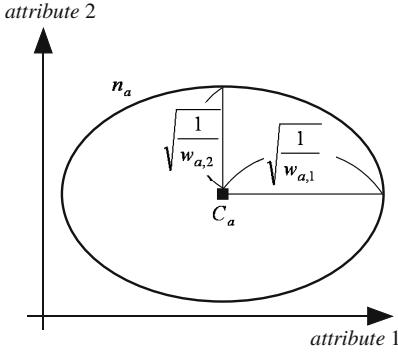
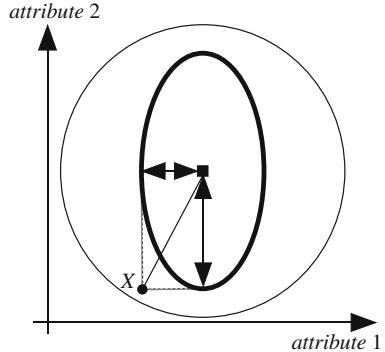
$$y_a = \operatorname{argmax}_{s_l \in S} freq(n_a, s_l). \quad (1)$$

ART_b also categorizes X into n_b and CPM evaluates the most frequent class y_b in n_b . When $y_a = y_b$, ARTMAP2-AW predicts that the class of I is y_a . Then, ART_a and ART_b are updated. $freq(n_a, y_a)$ and $freq(n_b, y_b)$ are also updated. When y_a and y_b do not match, match tracking is performed.

ART_a is based on ART2 [6]. In ART_a , the distance between X and C_a is defined by (2). $C_a = (c_{a,1}, c_{a,2}, \dots, c_{a,M})$ ($c_{a,m} \in [0, 1]$) is a center vector of n_a . $W_a = (w_{a,1}, w_{a,2}, \dots, w_{a,M})$ is a weight vector that indicates the degrees of importance of attributes. Figure 2 shows an example of a category. In this example, the attribute 2 is more important than the attribute 1 since $w_{a,2}$ is greater than $w_{a,1}$. The optimal value of $w_{a,m}$ is evaluated by match tracking and category merging. The details of them are discussed at section 2.2.

$$D(C_a, X) = \sqrt{\sum_{m=1}^M w_{a,m}(c_{a,m} - x_m)^2}. \quad (2)$$

When X is given, ART_a finds n_a minimizing $D(C_a, X)$. When $D(C_a, X) < 1$, n_a is adopted as the category for X and C_a is modified as $\frac{f_a-1}{f_a}C_a + \frac{1}{f_a}X$. f_a is the number of attribute vectors (including X) categorized into n_a . Otherwise, a new category n_{new} is created. C_{new} is initialized to X and $w_{new,m}$ is initialized to w_{init} . w_{init} is a parameter to control the size of a new category. In ART_b , n_b is selected on the basis of the distance defined by (2). However, any $w_{b,m}$ has a constant value w_b , since the output of ART_b is used as teacher signal. The parameter w_b controls the accuracy of classification rules.

**Fig. 2.** An example of a category**Fig. 3.** An example of match tracking

2.2 Match Tracking and Category Merging

Similar to fuzzy ARTMAP, ART_a learns classification rules and the output of ART_b is used as teacher signal. When y_a and y_b do not match, match tracking specializes the category n_a . In this case, the difference vector $V = (v_1, v_2, \dots, v_M) = (|c_{a,1} - x_1|, |c_{a,2} - x_2|, \dots, |c_{a,M} - x_M|)$ is evaluated at first. Then, $w_{a,m}$ is set at v_m^{-2} . Figure 3 shows an example of match tracking. The circle is the category n_a before match tracking. The ellipse is the category after match tracking.

Category merging is a generalization operation and is applied to categories of ART_a. The conditions to merge categories are shown in the following. When all of them are satisfied, two categories, n_j and n_k , are merged into n_{new} .

C1: Equation (3) is satisfied.

C2: Equation (4) is satisfied.

C3: Both the most and second frequent classes in n_j are equal to those in n_k .

$$G(C_j, C_k) = C'_j \cdot C'_k < G_{min}. \quad (3)$$

$$H(n_j, n_k) = \sum_{m=1}^M \left| \left(c_{j,m} + \sqrt{\frac{1}{w_{j,m}}} \right) - \left(c_{k,m} + \sqrt{\frac{1}{w_{k,m}}} \right) \right| < H_{min}. \quad (4)$$

In (3), C'_j (C'_k) is the normalized vector of C_j (C_k), i.e., $C'_j = \frac{C_j}{\|C_j\|_2}$. $C'_j \cdot C'_k$ is the inner product between C'_j and C'_k . G_{min} is a parameter controlling the ease of merging. $H(n_j, n_k)$ is an approximation of the size of the areas that either n_j or n_k covers. H_{min} also controls the ease of merging. Figure 4(a) shows the meanings of terms in $H(n_j, n_k)$. The third condition C3 is a constraint for not deteriorating rules by merging. When n_j and n_k are merged into n_{new} , the center vector $C_{new} = (c_{new,1}, c_{new,2}, \dots, c_{new,M})$ is evaluated by (5). The weight vector $W_{new} = (w_{new,1}, w_{new,2}, \dots, w_{new,M})$ is evaluated by using two difference vectors. One is the difference vector between C_j and C_{new} , i.e., $U_j = (u_{j,1}, u_{j,2}, \dots, u_{j,M}) = (|c_{j,1} - c_{new,1}|, |c_{j,2} - c_{new,2}|, \dots, |c_{j,M} - c_{new,j}|)$. The other is

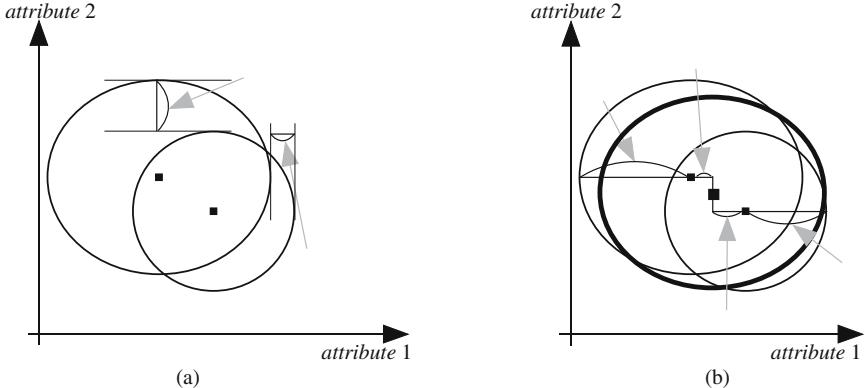


Fig. 4. An example of category merging. (a) Evaluation of the size of areas that either n_j or n_k covers. (b) Creation of a new category n_{new} from n_j and n_k .

U_k . Not to make n_{new} be too large, the greater of two candidates, $(u_{j,m} + w_{j,m}^{-\frac{1}{2}})^{-2}$ and $(u_{k,m} + w_{k,m}^{-\frac{1}{2}})^{-2}$, is selected as $w_{new,m}$. Figure 4(b) shows an example of category merging. In this example, $w_{new,1}$ is evaluated as $(u_{k,1} + w_{k,1}^{-\frac{1}{2}})^{-2}$.

$$c_{new,m} = \frac{\sqrt{\frac{1}{w_{j,m}}}}{\sqrt{\frac{1}{w_{j,m}}} + \sqrt{\frac{1}{w_{k,m}}}} c_{j,m} + \frac{\sqrt{\frac{1}{w_{k,m}}}}{\sqrt{\frac{1}{w_{j,m}}} + \sqrt{\frac{1}{w_{k,m}}}} c_{k,m}. \quad (5)$$

When we ignore attributes irrelevant to classification, computational costs of ARTMAP2-AW can be reduced. In ARTMAP2-AW, the degree of importance of the attribute m , \bar{w}_m , is defined as $\frac{1}{N_a} \sum_{j=1}^{N_a} w_{j,m}$. N_a is the number of categories in ART $_a$. Whenever a predefined number of cases are instantiated to ARTMAP2-AW, \bar{w}_m is evaluated. Then, ARTMAP2-AW marks the attribute minimizing \bar{w}_m as ignorable. In order not to deteriorate the accuracy of rules, the number of ignorable attributes is restricted to $\frac{M}{2}$.

2.3 Treatment of Missing Values, Symbolic Attributes and Unknown Cases

An attribute vector X might have some missing attribute values. $c_{j,m}$ might be unknown, too. For both cases, $(c_{j,m} - x_m)$ is evaluated as 0.

Possible values of some attributes might be defined as sets of symbolic values. In this paper, we call the attributes *symbolic attributes*. When some attributes are symbolic, each of them is translated to numeric attributes. When possible values of a symbolic attribute are defined as $\{s_1, s_2, \dots, s_L\}$, a symbolic value s_l is translated to the L -dimensional binary vector $T_l = (t_1, \dots, t_L)$ of which l -th component is one and of which others are zero.

When an unknown case whose class is unknown is given, ART_a categorizes its attribute vector into a category n_a . Then, the most frequent class in n_a becomes a predicted class of the case.

3 Experimental Results

ARTMAP2-AW is implemented in C language and it has been run on a 2.4GHz Intel Pentium-4 processor with 512MB of memory. The method has been applied to data sets from UCI Machine Learning Repository [9] and has been compared with four classifiers, C4.5, RBFN, SVM, and Decision Table, which are available in the collection of machine learning algorithms “Weka” (version 3.4) [7].

Table 1. Characteristics of data sets

Data set	Attr	CLS	Case		Data set	Attr	CLS	Case
<i>abalone</i>	10	29	4177		<i>tic-tac-toe</i>	9(27)	2	958
<i>cmc</i>	8	3	1472		<i>letter-recognition</i>	16	27	20000
<i>dermatology</i>	34	6	366		<i>magic04</i>	10	2	19020
<i>pima</i>	8	2	742		<i>abalone2</i>	12	29	4177
<i>credit-screening</i>	13(48)	2	690		<i>yeast</i>	9	10	1484

Table 2. Results

	C4.5		RBFN		SVM		Decision Table		ARTMAP2-AW	
	Acc	CPU	Acc	CPU	Acc	CPU	Acc	CPU	Acc	CPU
<i>abalone</i>	77.7	730	48.1	930	49.3	9980	42.3	1240	80.6	1650
<i>cmc</i>	95.9	120	66.6	270	94.1	750	82.6	190	98.2	190
<i>dermatology</i>	94.7	70	91.5	750	89.1	1130	92.1	125	98.5	152
<i>pima</i>	72.6	40	72.2	140	78.4	540	76.6	80	85.6	327
<i>credit-screening</i>	86.5	150	79.5	450	74.5	4650	84.2	350	89.2	425
<i>tic-tac-toe</i>	92.8	110	65.5	350	98.1	320	76.7	140	98.9	301
<i>letter-recognition</i>	87.6	5210	N/A	N/A	N/A	N/A	62.1	25470	90.3	2650
<i>magic04</i>	86.3	4810	N/A	N/A	N/A	N/A	82.4	3520	95.2	1240
<i>abalone2</i>	76.8	850	50.2	1200	48.9	5230	45.6	1690	81.2	2625
<i>yeast</i>	52.2	520	49.1	780	56.2	1980	45.3	350	90.0	880

Table 1 shows the characteristics of the data sets. *Attr* indicates the number of attributes. *CLS* and *Case* are the numbers of classes and cases, respectively. Since some attributes in *credit-screening* and *tic-tac-toe* are symbolic, each of them is translated to a binary vector. Values in parentheses indicate the numbers of attributes after the translation. *abalone2* and *yeast* have some attributes irrelevant to classification. *abalone2* was created by adding two random attributes to the data set *abalone*. In order to evaluate the degrees of importance of attributes, each element x_m in any attribute vector X was normalized.

w_{init} for ART_a is 0.01 and w_b for ART_b is 0.1. G_{min} and H_{min} are 0.5 and 2.0, respectively. These parameter values are decided by performing preliminary experiments.

Table 2 shows the results. *Acc* indicates the accuracy [%] of the classification rules and *CPU* is the computational time [msec]. For the data sets, ARTMAP2-AW acquires rules with the highest accuracy. When the number of cases is small, ARTMAP2-AW takes more computational time than other methods. For data sets with a large number of cases such as *letter-recognition*, however, ARTMAP2-AW does not need more computational time than other methods. Finally, we analyze results for the data sets *abalone2* and *yeast* that have some irrelevant attributes to classification. In Ref. [6], it is reported that irrelevant attributes deteriorate the performance of fuzzy ARTMAP. ARTMAP2-AW, however, does not take computational time so much and can acquire classification rules with high accuracy.

4 Conclusion

In this paper, we have proposed a new classifier ARTMAP2-AW. It is the modification of ARTMAP to consider the importance of each attribute. Experimental results show that ARTMAP2-AW acquires better classification rules than well-known classifiers.

Acknowledgments. The authors would like to thank the anonymous reviewers for their valuable comments. This work is partly supported by Grant for Special Academic Research (Grant No. 7115) from Hiroshima City University.

References

1. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)
2. Anagnostopoulos, G.C., Georgopoulos, M.: New geometrical concepts in fuzzy-ART and fuzzy-ARTMAP: Category regions. In: Proc. IJCNN 2001, vol. 1, pp. 32–37 (2001)
3. Carpenter, G.A., Grossberg, S., Markuzon, M., Reynolds, J.H., Rosen, D.B.: Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans. Neural Networks 3(5), 698–713 (1992)
4. Er, M.J., Wu, S., Lu, J.W., Toh, H.L.: Face recognition using radial basis function (RBD) neural networks. IEEE Trans. Neural Networks 13(3), 697–710 (2002)
5. Zaho, W.B., Huang, D.S., Du, J.Y., Wang, L.M.: Genetic optimization of radial basis probabilistic neural networks. Int. J. Pattern Recognition and Artificial Intelligence 18(8), 1473–1499 (2004)
6. Ueda, H., Nasu, Y., Yamada, T., Takahashi, K., Miyahara, T.: Acquiring classification rules by using adaptive resonance theory. In: Proc. SMC 2007, pp. 1693–1698 (2007)
7. Frank, E., et al.: Weka Machine Learning Project, Univ. Waikato, <http://www.cs.waikato.ac.nz/ml/weka>
8. Carpenter, G.A., Grossberg, S., Rosen, D.B.: Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks 4(6), 759–771 (1991)
9. Blake, C., Merz, C.: UCI repository of machine learning databases. Dept. Inform. and Comput. Sci., Univ. California, Irvine, CA, <http://www.ics.uci.edu/~mlearn/MLRepository.html>

An Improved Tabu Search Algorithm for 3D Protein Folding Problem

Xiaolong Zhang and Wen Cheng

School of Computer Science and Technology
Wuhan University of Science and Technology, Wuhan 430081 P.R. China
xiaolong.zhang@wust.edu.cn, cw1022@163.com

Abstract. An improved Tabu Search (TS) algorithm is presented for three-dimensional (3D) protein folding structure prediction in off-lattice protein AB model. Tabu Search algorithm is one of global optimization algorithms which has strong local search and has been applied for many combination optimization problems. The experimental results show that the ground-state energies obtained by our algorithm are better than those by previous methods. Moreover, the improved Tabu search algorithm has higher searching performance and can be effectively used to predict 3D protein folding structure.

Keywords: protein folding prediction, off-lattice model, Tabu search.

1 Introduction

Since protein sequence of amino acid and its environment determine their three-dimensional conformation, predicting the native structure of a protein from its sequence is one of the most important problems in biophysics. Based on the minimum energy hypothesis [1] that protein native structures are conformation at the global minima of their accessible free energies, some theoretical methods have been applied for protein structure prediction, such as genetic-annealing algorithm (GAA) [2], conformational space annealing (CSA) [3], and simulated annealing [4].

Considering the complexity of realistic protein models, much work has been devoted to study simple models. One prominent model is off-lattice AB model [5], where the hydrophobic monomers are labeled by A and the hydrophilic ones by B. In this model, the interaction between nonadjacent monomers and the interaction between successive bonds are both considered.

Tabu search (TS) algorithm proposed by Glover [6,7,8] is one of heuristic iterative optimization algorithms and it has strong local search. Tabu search algorithm has two important features. To avoid trapping in local optima, the approach records recent certain moves in one tabu list. If a move is in the tabu list, the move leads to is undesirable. Another important feature is the aspiration criterion which helps the algorithm realizes global optimization. If a move is in the tabu list but it leads to a new solution which is better than the current optimal solution, its related solution is desirable.

The rest of this paper is organized as follows: off-lattice AB model is introduced in Section 2. Improved strategies and implementation of the improved algorithm are presented in Section 3. Experimental results and their discussion are in Section 4. Conclusions are given in Section 5.

2 Off-Lattice AB Model

AB model [5] consists of hydrophobic (A) monomers and hydrophilic (B) monomers. In three-dimensional space, the shape of a n -mer is specified by $n-2$ bond angles $\theta_2, \dots, \theta_{n-1}$ and $n-3$ torsional angles $\beta_3, \dots, \beta_{n-1}$. θ_i is angle between adjacent bond vectors and β_i is plane angle between three successive bond vectors. We adhere to the condition $-\pi \leq \theta_i < \pi$, $-\pi \leq \beta_i < \pi$. In AB model [5], the energy function for any N monomer chain is given by:

$$E = \sum_{i=2}^{n-1} \frac{1}{4} (1 - \cos \theta_i) + \sum_{i=1}^{n-2} \sum_{j=i+2}^n 4[r_{ij}^{-12} - C(\xi_i, \xi_j)r_{ij}^{-6}] \quad (1)$$

Here r_{ij} denotes the distance between monomers i and j . ξ_i denotes the kind of residues ($\xi_i = 1$ for hydrophobic and $\xi_i = -1$ for hydrophilic monomers). $C(\xi_i, \xi_j)$ is $+1, +1/2$ and $-1/2$, respectively, for AA, BB and AB pairs.

In off-lattice AB model, predicting 3D folding structure of n monomer is to find suitable $n-2$ bond angles and $n-3$ torsional angles which make energy function minimum. Therefore, the prediction problem becomes a global optimization problem:

$$\min_{\theta_i, \beta_i \in (-\pi, \pi)} E(\theta_2, \dots, \theta_{n-1}; \beta_3, \dots, \beta_{n-1})$$

3 Strategy and Improved Tabu Search Algorithm

3.1 Improved Strategies

Tabu search algorithm [6,7] starts with an initial solution which is generated randomly. New solutions are generated in a candidate set. The candidate set is a subset of neighborhood of the current solution. The performance of TS algorithm largely depends on the initial solution, the proper choice of the neighborhood, tabu list length, the aspiration criteria. A bad initial solution may cause lower convergence speed. When the given tabu list length is too small, it may lead to cycling searching. When the tabu list length is too large, it may cause total tabu. When candidate set size is too small, premature convergence easily occurs. One has to propose some strategies to adapt it for the above considered problem. Improved strategies are as follows.

1) Method of generating an initial solution

Since the native TS algorithm generates the initial solution randomly. Considering randomness and searching time, an effective heuristic strategy, which was also used for other algorithms, is proposed. On the basis of observation that the hydrophobic amino acid residues are always flanked by hydrophilic amino acid residues and form a single core in real proteins, the main idea is to locate hydrophobic residues at the center of three-dimension space and locate hydrophilic residues surrounding hydrophobic ones. In addition, every solution x is defined as a vector $(\theta_1, \dots, \theta_{n-2}, \beta_1, \dots, \beta_{n-3})$. The energy value (computed by Eq.(1)) of x is described as $E(x)$. Global

optimal solution is defined as x_{\min} and corresponding global optimal energy is described as E_{\min} .

2) Approach of generating neighborhood

Neighborhood $N(x)$ is a set of neighbor solution of current solution x . Disturbance mutation strategy in Genetic Algorithm is applied to generating neighbor solution. In order to assure searching diversity, two-point disturbance mutation is used in the early stage of searching. To ensure that the algorithm converges to global optimal solution, single-point mutation is used in the later stage. Detailed mutation implementation is presented as follows. For current solution $x=(\theta_1, \dots, \theta_{n-2}, \beta_1, \dots, \beta_{n-3})$, randomly select one or two elements of the solution vector to generate mutation. The selected j th element is described as x^j and then new element x_{new}^j is given by:

$$x_{new}^j = x^j + f(r) \times 2\pi \times \text{random}(0 \dots 1) \times \text{rate}^i \quad (2)$$

Where

$$f(r) = \begin{cases} -1, & r < 0.5 \\ 1, & r \geq 0.5 \end{cases} \quad (3)$$

Here r is generated randomly number between 0 and 1. rate^i is to ensure the diversity of neighbor solution. rate is a scale factor and i denotes the iteration number of generating neighbor solution and changes from 0 to $NL - 1$ (NL is the size of the neighborhood). In this paper, we set $\text{rate}=0.95$.

3) Candidate selection method

A candidate set $C(x)$ is a subset of the neighborhood $N(x)$. When building the candidate set, the algorithm computes energy value of each solution of neighborhood $N(x)$ and then sorts the solutions by their energy values in the descending, finally selects the top CL solutions with lowest energies as candidate set.

4) Tabu list, Tabu conditions, total tabu

Tabu list is a set of solutions from the last TL (Tabu list length) iterations of our algorithm. To avoid trapping in local optima, TL is defined as adaptive dynamical variable and varies between 8 to 11.

A solution vector of a candidate set is described as $z(\theta_1, \dots, \theta_{n-2}, \beta_1, \dots, \beta_{n-3})$ and its energy value as $E(z)$. Any solution vector of the tabu list is described as $y(\theta_1, \dots, \theta_{n-2}, \beta_1, \dots, \beta_{n-3})$ and its energy value is $E(y)$. The tabu conditions are described as follows:

- (1) $|E(y) - E(z)| \leq \varphi$, $|E(y) - E(z)|$ is a change value between two energy values.
- (2) $\|y - z\| \leq \eta$, $\|y - z\|$ is the distance of two solution vectors.

If the first condition and the second condition above are both satisfied, the new solution is considered as tabu, that is, we should select other solutions from the candidate set. In this paper, we set $\varphi = 0.08$, $\eta = 0.004$.

When all solutions in the candidate set are forbidden and no solution is better than the current best solution, this condition is considered as total tabu. Our algorithm randomly selects one element of current best solution to generate mutation and the new generated solution is considered as the next current solution.

3.2 Improved Tabu Search Algorithm

The improved TS algorithm is described as follows:

TS($x_0, x, E(x), E_{\min}, N(x), C(x), k, L_{\max}$)

x_0 : initial solution in the first iteration.

x : current solution in all iterations. x_{\min} : global optimal solution.

$E(x)$: energy value of current solution x . E_{\min} : global optimal energy.

$N(x)$: neighborhood of current solution x .

$C(x)$: a candidate set.

k : current iteration number.

L_{\max} : max iteration number.

Begin

Initialize(generate $x_0(\theta_1, \dots, \theta_{n-2}, \beta_1, \dots, \beta_{n-3})$, compute its energy $E(x_0)$, $x:=x_0$, $x_{\min}:=x_0$, $E(x):=E(x_0)$, $E_{\min}:=E(x_0)$, neighborhood size NL , candidate set size CL , tabu list length TL , tabu list $T1 = \emptyset$, $T2 = \emptyset$, L_{\max}).

Set $k:=1$.

While($k < L_{\max}$), do:

1. If($k < L_{\max} \times \delta$) then generate the neighborhood $N(x)$ using two-point mutation; else generate $N(x)$ using single-point. Here $\delta=0.85$.
2. build candidate set $C(x)$ using Candidate selection method in Section 3.1 and choose the optimal solution x_1 of the candidate set and its energy value is $E(x_1)$.
3. If ($E(x_1) < E_{\min}$), then $x_{\min}:=x_1$, $E_{\min}:=E(x_1)$.
4. If ($E(x_1) < E(x)$), then update current solution $x:=x_1$, set $E(x):=E(x_1)$ and go to 9.
5. Set $l := 1$. (l denotes the subscript of the solution in candidate set).
6. If ($l \leq CL$), then go to 7; else go to 8.
7. If tabu condition are not satisfied for x_l , then update $x:=x_l$, set $E(x):=E(x_l)$ and go to 9; else set $l := l + 1$ and return to 6.
8. generate new solution x_{new} according to total tabu method and compute its energy value $E(x_{\text{new}})$, update current solution $x:=x_{\text{new}}$ and set $E(x):=E(x_{\text{new}})$. If ($E(x_{\text{new}}) < E_{\min}$), then update the best solution $x_{\min}:=x_{\text{new}}$ and set $E_{\min}:=E(x_{\text{new}})$.
9. Update tabu list: if tabu list is full then remove the first element from tabu list T_1 and the first element (corresponding energy value) from tabu list T_2 . put the current solution x at the end of T_1 and put energy value $E(x)$ at the end of T_2 .
10. Set $k:=k+1$.

End

4 Experimental Results and Discussion

4.1 3D Structure Prediction for Fibonacci Sequences

The Fibonacci sequence are also studied in [3,4,9,10]. Table 1 shows the optimized energies in three-dimensional AB model. The result comparison of several algorithms

is also given. The results are obtained with our improved Tabu search algorithm (TS), those by improved pruned enriched Rosenbluth method with importance sampling (nPERM) [9], simulated annealing (SA) [4], energy landscape paving minimizer (ELP) [10], and conformational space annealing (CSA) [3].

Table 1 shows that our results (E_{TS}) are much lower than those by nPERM, SA for all the three Fibonacci sequences, and are also better than those by ELP, CSA. For all these sequences, our algorithm appears to have the best optimized results among the above methods.

Table 1. Lowest energies obtained by nPERM, SA, ELP, CSA, TS

N	SEQUENCE	E_{nPERM}	E_{SA}	E_{ELP}	E_{CSA}	E_{TS}
13	ABBBABBABBBAB	-4.9616	-4.9746	-4.967	-4.9746	-6.5687
21	BABBBBABBBABB	-11.5238	-12.0617	-12.316	-12.3266	-13.4151
	ABABBAB					
	ABBABBABBBABB					
34	ABABBABABBABAB	-21.5678	-23.0441	-25.476	-25.5113	-27.9903
	ABBAB					

4.2 3D Structure Prediction for Real Protein

As the previous methods such as ELP [10] and CSA [3] are just studied for Fibonacci sequences and have not been used to predict real protein sequences, we also predict some real protein sequences with our improved TS algorithm. The real protein sequences are obtained from <http://pdbebeta.rcsb.org/pdb/Welcome.do>. Following K-D method [11] that is used to distinguish hydrophobic (A) monomers and hydrophilic (B) monomers, I, V, L, P, C, M, A, G monomers are considered as hydrophobic (A) monomers and D, E, F, H, K, N, Q, R, S, T, W, Y monomers are considered as hydrophilic (B) monomers.

Table 2. Real protein sequences and their lowest energies

No	PDB ID	SEQUENCE	E_{TS}
1	1BXL	GQVGRQLAIIGDDINR	- 15.7164
2	1EDP	CSCSSLMDKECVYFCHL	- 12.8392
3	1AGT	GVPINVSCTGSPQCICPKD QGMRGKCMNRKCHCTPK	- 44.2656

In Table 2, “PDB ID” is the only identifier of real protein sequence in PDB Protein Data Bank and the “SEQUENCE” column displays sequence of real protein. E_{TS} is the lowest energy of real protein sequence obtained by improved TS algorithm.

5 Conclusions

An improved Tabu search algorithm for protein three-dimensional structure and some improved strategies are proposed in this paper. The generation of the initial solution

depends on an effective heuristic strategy. The experimental results show our algorithm acquires the better lowest-energy conformation compared to the previous algorithms. As one of the future work, we try to make use of parallel feature of Genetic algorithm, and combine Tabu search algorithm and Genetic algorithm to improve the efficiency of the search.

Acknowledgment

This work was supported in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, and National Natural Science Foundation of P. R. China (60674115), as well as Open Foundation of State Key Laboratory of Bioelectronics, P.R. China.

References

1. Anfinsen, C.B.: Principles that govern the folding of protein chains. *Science* 181, 223–227 (1973)
2. Zhang, X., Lin, X., Wan, C., Li, T.: Genetic-annealing algorithm for 3D off-lattice protein folding model. In: PAKDD Workshops 2007, pp. 186–193 (2007)
3. Kim, S.-Y., Lee, S.B., Lee, J.: Structure optimization by conformational space annealing in an off-lattice protein model. *Phys. Rev. E* 72, 011916 (2005)
4. Chen, M., Huang, W.: Simulated annealing algorithm for protein folding problem. *Mini-Micro Systems* 28(1), 75–78 (2007) (in Chinese)
5. Stillinger, F.H., Head-Gordon, T., Hirshfeld, C.L.: Toy model for protein folding. *Phys. Rev. E* 48, 1469–1477 (1993)
6. Glover, F.: Tabu search: Part I. ORSA. *Journal on Computing* 1, 190–206 (1989)
7. Glover, F.: Tabu search: part II. ORSA. *Journal on Computing* 2, 4–329 (1990)
8. Glover, F.: Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* 13, 533–549 (1986)
9. Hsu, H.-P., Mehra, V., Grassberger, P.: Structure optimization in an off-lattice protein model. *Phys. Rev. E* 68, 037703 (2003)
10. Bachmann, M., Arkin, H., Janke, W.: Multi-canonical study of coarse grained off-lattice models for folding heteropolymers. *Phys. Rev. E* 71, 031906 (2005)
11. Mount, D.W.: *Bioinformatics: sequence and genome analysis*. Cold Spring Harbor Laboratory Press (2001)

Transferring Knowledge from Another Domain for Learning Action Models

Hankui Zhuo¹, Qiang Yang², Derek Hao Hu², and Lei Li¹

¹ Software Research Institute, Sun Yat-sen University
zhuohank@gmail.com

² Hong Kong University of Science and Technology
{qyang, derekhh}@cse.ust.hk*

Abstract. Learning action models is an important and difficult task for AI planning, since it is both time-consuming and tedious for a human to encode the action models by hand using a formal language such as PDDL. In this paper, we present a new algorithm to learn action models from plan traces by transferring useful knowledge from another domain whose action models are already known. We call this algorithm *t-LAMP*, (transfer Learning Action Models from Plan traces) which can learn action models in PDDL language with quantifiers from plan traces where the intermediate states can contain noise and partial information. We apply Markov Logic Network to enable knowledge transfer, and show that using the transfer learning framework, the quality of the learned action models are generally better than the case when not using an existing domain for transfer.

1 Introduction

Planning systems require action models as input. A typical way to describe action models is to use action languages such as the planning domain description language (PDDL) [6]. A traditional way of building action models is to ask domain experts to analyze a planning domain and write a complete action model representation. However, it is very difficult and time-consuming to build action models in complex real world scenarios in such a way, even for experts. Thus, researchers have explored ways to reduce the human efforts of building action models by learning from observed examples or plan traces. Some researchers have developed methods to learn action models from complete state information before and after an action [1]. Other researchers, such as Yang, Wu and Jiang [2,3] have developed an approach known as *Action Relation Modeling System* (ARMS) to learn action models in a STRIPS [5] representation using a weighted maximal satisfiability based approach.

In this paper, we present a novel action-model learning algorithm called *t-LAMP*, which stands for *transfer Learning Action Models from Plan traces*. In this algorithm, we use the shared common knowledge of one domain to help learn another domain. For instance, we may have a domain *elevator*¹ where action models are already encoded. One example action is *the action ‘up(?)f1,?f2’* which means the elevator can go up from

* We thank the support of CERG grant HKUST 621307.

¹ <http://www.cs.toronto.edu/aips2000/>

a floor ‘f1’ to a floor ‘f2’. This action has a precondition ‘lift-at(?f1)’ and an effect ‘lift-at(?f2)’. Now suppose we wish to learn the logical model of an action ‘move(?l1,?l2)’ in the *briefcase*¹ domain, which means that the case is moved from location ‘l1’ to location ‘l2’. In this new domain we have a precondition ‘is-at(?l1)’ and an effect ‘is-at(?l2)’. Because of the similarity between the two domains, these two actions share the common knowledge on the actions that can cause changes in locations. Thus, to learn one action model, transferring the knowledge from the other action model is likely to be helpful.

2 Problem Definition and Our Algorithm

A classical planning problem can be represented as $\mathcal{P} = (\Sigma, s_0, g)$, where $\Sigma = (S, A, \gamma)$ is the planning domain, s_0 is the initial state, and g is the goal state. In Σ , S is the set of states, A is the set of actions, γ is the deterministic transition function. A solution to a planning problem is called a plan, an action sequence (a_0, a_1, \dots, a_n) . Each a_i is an action schema in the form of ‘action name(parameters)’ such as ‘move(?m - location ?l - location)’. Furthermore, a plan trace is defined as $T = (s_0, a_0, \dots, s_n, a_n, g)$, where s_1, \dots, s_n are intermediate state observations allowed to be partial or empty. Our learning problem can be stated as follows. We are given: (1) a set of plan traces \mathcal{T} in a *target domain* (that is, the domain from which we wish to learn the action models), (2) the description of predicates and action schemas in the target domain, and (3) the completely available action models in a similar *source domain*. As output, *t-LAMP* will output the preconditions and effects of each action schema in our target domain \mathcal{T} .

The *t-LAMP* algorithm can be described shortly in four steps. In our first step, we will encode the plan traces into propositional formulae, which is rather standard and interested readers could refer to [9] for technical details. Then in Step 2, we will generate formulae according to some specific correctness constraints and provide the generated formulae as the input of the MLN (denoted as M). In Step 3, we will encode the action model from our *source domain* into another MLN, denoted as M^* , and then transfer from knowledge from M^* to M . After that we can learn the most likely subset of candidate formulae in M . In the last step, we will convert the formulae we learn to the final action models. We will describe Step 2, 3, 4 in detail, where Step 1 is omitted since readers can check it in [9] as we mentioned.

2.1 [Step 2] Generating Candidate Formulae

In Step 1, plan traces have been encoded as a set of propositional formulae, each of which is a conjunction of propositional variables. Thus plan traces can be represented by a set of propositional variables, whose elements are conjunctions. This set is recorded in a database called DB.

Next, we will generate candidate formulae for individual actions in the following steps from **F1** to **F4**. These candidate formulae attempt to ensure the correctness of action models, that the action models generated are sound. Due to space constraints, we omit the detailed formulae we will add as constraints into MLN in this paper. Nevertheless, we will describe what characteristics our constraints must satisfy so that the correctness and soundness of the action models are ensured.

- F1:** (*The effect of an action must hold after the execution of this action.*) If a literal p is an effect of some action a , then an instance of p must hold after a is executed.
- F2:** (*The negative effect of an action must have its corresponding positive counterpart hold before the execution of this action.*) Similar to F1, a literal p 's negation is an effect of some action a , which means an instance of p is deleted after a is executed, but it exists before the execution of a .
- F3:** (*The precondition of an action will be the subset of the state before the execution of this action.*) A formula f (can be a single literal, or with quantifiers) is a precondition of a , which means the instance of f holds before a is executed.
- F4:** (*A conditional effects holds only when its condition holds before the action.*) A conditional effect, in PDDL form, like “*forall \bar{x} (when $f(\bar{x}) q(\bar{x})$)*”, is a conditional effect of some action a , which means for any \bar{x} , if $f(\bar{x})$ is satisfied then $q(\bar{x})$ will be added after a is executed.

By **F1-F4**, we can generate possible candidate formulae which are used to describe combinations that are possible for describing preconditions and effects based on the soundness requirement of individual actions.

2.2 [Step 3] Transfer Learning Weights of a MLN

Encoding source-domain action models as a MLN M^* : In this step, we convert all source domain action models into formulae F1 to F4 in order to transfer the source domain knowledge to target domain knowledge. To do this, we convert each action model to the formulae, and then give them the maximum weights. When these formulas are put together with the target domain formulae in the next step, they will influence the learning of action models in the target domain through the mapping function between the two domains. Note that the mapping function can be learned as well.

Transfer learning M from M^* : We find the best way to map M^* into M based on the quality of the mapping. The quality of a mapping is measured by the performance of M on DB, estimated by a weighted pseudo log-likelihood measure (WPLL) score. It sums over the log-likelihood of each node given its Markov blanket, weighting it appropriately to ensure that predicates with many literals do not dominate the result. We do a global mapping to establish a mapping from each predicate in M^* to a predicate in M and then use it to translate the entire M^* to a new MLN \bar{M} . The algorithm is shown below. Note that we do not need to require each mapping to be complete.

Transfer Learning M from M^* :

Input: M and M^*

Output: M , whose weights of formulae are initiated

1. Find a mapping from each predicate in M^* to a predicate in M .
2. By the mapping, translate the entire M^* to a new MLN \bar{M} .
3. Learning weights of formulae in \bar{M} with DB.
4. Compute the WPLL in this iteration with a previous one. If the new WPLL is better, then $\bar{M}_{best} = \bar{M}$.
5. If all the possible mappings are done, continue; else do a new mapping and goto 2.
6. Assign the weight of each formula in \bar{M}_{best} to the same formula in M , leaving the weights of other formulae as zero, and output M .

In the first step of the algorithm, a mapping is found by the following process: firstly, for a predicate p^* in M^* and a predicate p in M , we build a *unifier* by mapping their corresponding names and arguments (we require that the number of arguments are the same in p^* and p , otherwise, we find next p to be mapped with p^*); and then substitute all the predicates in M by this unifier; for every p^* and p , we repeat the process of unifier-building and substitution. Next, by the mapping built in the first step, we translate the formulae of M^* to M 's formulae, whose predicates belong to M . In the third step, The learning process can refer to [4]. The other steps are straightforward. After M is outputted, we can finally learn weights of M as presented in [4,8].

2.3 [Step 4] Generating Action Models

The optimization of WPLL indicates that when the number of true grounding of f_i is larger, the corresponding weight of f_i will be higher. Thus, the final weight of a formula in the MLN is a confidence measure of that formula. Intuitively speaking, the larger the weight of a formula is, the more probable that formula will be. However, when generating the final action models from these formulae, we need to determine a threshold, based on the validation set of plan traces and our evaluation criteria (definition of error rate) to choose a set of formulae from MLN.

3 Experiments

3.1 Data Set and Evaluation Criteria

We collect plan traces from *briefcase* and *elevator* domains . Traces are generated by generating plans from the given initial and goal states in these planning domains using the human encoded action models and a planning algorithm, FF planner². These domains have the characteristics we need to evaluate our *t-LAMP* algorithm: they have enough similarities and hence we have the intuition that one can borrow knowledge from the other while learning the action models (as shown in the examples of earlier sections). The initial and goal states we use in our experiments are from planning competition (IPC-2), which we use for generating plan traces as input.

We define error rates of our learning algorithm as the differences between our learned action models and the hand-written action models that are considered as the “ground truth” from IPC-2. If a precondition appears in our learned action models’ preconditions but not in hand-written action models’ preconditions, the error count of preconditions, denoted by $E(\text{pre})$, increases by one. Similarly, if a precondition appears in hand-written action models’ preconditions but not in our learned action models’ preconditions, $E(\text{pre})$ increases by one. Likewise, error count of effects are denoted by $E(\text{eff})$. Furthermore, we denote the total number of all the possible preconditions and effects of action models as $T(\text{pre})$ and $T(\text{eff})$, respectively. In our experiments, the error rate of an action model is defined as $R(a) = \frac{1}{2}(E(\text{pre})/T(\text{pre}) + E(\text{eff})/T(\text{eff}))$, where we assume the error rates of preconditions and effects are equally important, and the range of error rate $R(a)$ should be within $[0,1]$. Furthermore, the error rate of all the action models A is defined as $R(A) = \frac{1}{|A|} \sum_{a \in A} R(a)$, where $|A|$ is the number of A ’s elements.

² <http://members.deri.at/joergh/ff.html>

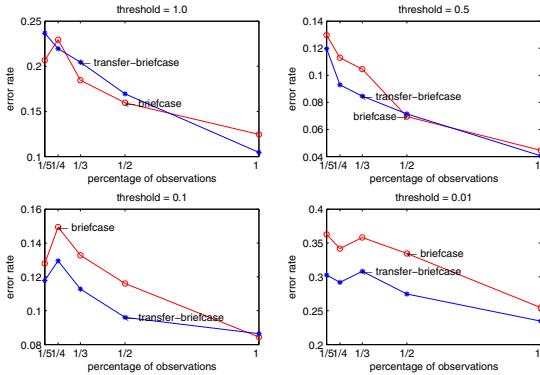


Fig. 1. Learning action models in *briefcase* with or without transferring knowledge from *elevator*. The blue (red) curve shows the result with (without) transfer learning.

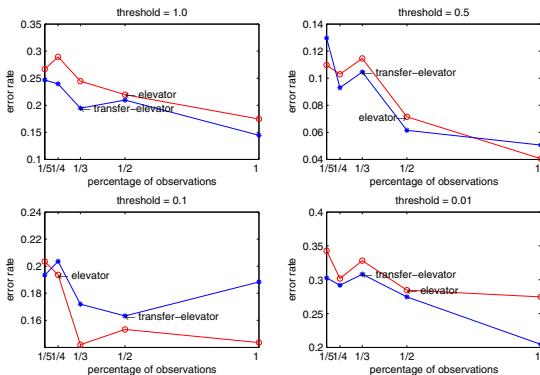


Fig. 2. Learning action models in *elevator* by transferring knowledge from *briefcase*. The blue (red) curve shows the result with (without) transfer learning.

3.2 Experimental Results

The evaluation results of *t*-LAMP in two domains are shown in Figure 1 and 2. Figure 1 shows the result of learning the action models in *briefcase* by transferring the knowledge from *elevator*, while Figure 2 shows the result of learning the action models in *elevator* by transferring the knowledge from *briefcase*. We have chosen different thresholds with weights 1.0, 0.5, 0.1 and 0.01 to test the effect of the threshold on the performance of learning. The results show that generally the threshold can be neither too large nor too small, but the performance is not very sensitive to the value.

Since our *t*-LAMP algorithm does not require all intermediate states, we can still learn useful information from a partial set of intermediate states. In our experiment, we have chosen the observable percentage of 1/5, 1/4, 1/3, 1/2, 1 where 1/5 of observable intermediate states means we will keep only one intermediate state to be observable in every five successive actions. The other percentages 1/4, 1/3, 1/2 and 1 have similar

meanings. Our experiment shows that in most cases, the more states that are observable, the lower the error rate will be, which is consistent with our intuition. However, there are some other cases, e.g. when threshold is set to 1.0 and there are only 1/3 of states that are observable, the error rate is higher than the case when 1/2 of states are observable.

From experiments, we can see that transferring useful knowledge from another domain will help improve our action model learning result. On the other hand, determining the similarity of two domains is important, which will be given in our future work.

4 Conclusion

In this paper, we have presented a novel approach to learn action models through transfer learning and a set of observed plan traces. Our *t*-LAMP learning algorithm makes use of *Markov Logic Networks* to learn action models by transferring knowledge from another domain. Our empirical tests in two domains showed that the method is both accurate and effective in learning the action models via knowledge transfer. In the future, we wish to understand better the conditions under which transfer learning is effective in learning the action models, and to extend the learning algorithm to more elaborate action representation languages including resources and functions. We also wish to explore how to make use of other inductive learning algorithms to help us learn better.

References

1. Blythe, J., Kim, J., Ramachandran, S., Gil, Y.: An integrated environment for knowledge acquisition. *Intelligent User Interfaces*, 13–20 (2001)
2. Yang, Q., Wu, K., Jiang, Y.: Learning Actions Models from Plan Examples with Incomplete Knowledge. In: ICAPS, 241–250 (2005)
3. Yang, Q., Wu, K., Jiang, Y.: Learning action models from plan examples using weighted MAX-SAT. *Artif. Intell.* 171(2-3), 107–143 (2007)
4. Richardson, M., Domingos, P.: Markov Logic Networks. *Machine Learning* 62(1-2), 107–136 (2006)
5. Fikes, R., Nilsson, N.J.: STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artif. Intell.* 2(3/4), 189–208 (1971)
6. Fox, M., Long, D.: PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *J. Artif. Intell. Res. (JAIR)* 20, 61–124 (2003)
7. Mihalkova, L., Huynh, T., Mooney, R.J.: Mapping and Revising Markov Logic Networks for Transfer Learning. In: AAAI (2007)
8. Kok, S., Singla, P., Richardson, M., Domingos, P.: The Alchemy system for statistical relational AI. University of Washington, Seattle (2005)
9. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory and Practice. Morgan Kaufmann, San Francisco (2004)

Texture and Target Orientation Estimation from Phase Congruency

Qingbo Yin^{1,2}, Liran Shen³, and Jong Nam Kim¹

¹ Division of Electronic Computer and Telecommunication Engineering, Pukyong National University, 599-1 Daeyeon-dong Nam-gu, Busan, 608-737, Korea

² College of Computer Science and Technology, Harbin Engineering University, P.R. China

³ College of Information and Communication Engineering, Harbin Engineering University, P.R. China

yinqingbo@hrbeu.edu.cn

Abstract. The problem of orientation estimation is basic to many tasks in machine vision and image processing. A new approach for orientation estimation is proposed based on the phase congruity in radon domain. Here, the image principal direction is defined as the orientation of the image, which has the maximum of phase congruity of variance. The performance of this technique is determined by conducting simulation experiments on two sets of images, containing military targets and textures, respectively.

Keywords: texture and target, orientation estimation, phase congruency.

1 Introduction

The problem of orientation estimation is basic to many tasks in machine vision and image processing. In automatic target recognition problem it is necessary to identify the desired target in a scene and to determine its exact location and orientation [1]. It was also proved to be one of the three fundamental properties influencing texture recognition along with complexity and periodicity.

There are techniques in the literature to estimate the orientation of the image, including methods based on image gradients [2], angular distribution of signal power in the Fourier domain [3], [1], and signal autocorrelation structure [2]. Here, a method based on phase congruity in Radon domain is proposed to estimate the texture and target orientation.

2 Proposed Method for Orientation Estimation

Due to the inherent properties of the Radon transform, it is a useful tool to capture the directional information of the images. The Radon transform of a 2D function $f(x,y)$ is defined as:

$$R(r,\theta)[f(x,y)] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x,y) \delta(r - x \cos \theta - y \sin \theta) dx dy \quad (1)$$

where r is the perpendicular distance of a line from the origin and θ is the angle between the line and the y-axis [4].

Morrone and Owens [5] define the phase congruency function in terms of the Fourier series expansion of a signal at some location x as

$$PC(x) = \max_{\bar{\phi}(x) \in [0, 2\pi]} \frac{|E(x)|}{\sum_n A_n(x)}, \quad (2)$$

$$|E(x)| = \sum_n A_n(\cos(\phi(x) - \bar{\phi}(x)))$$

where A_n represents the amplitude of the n th Fourier component, and $\phi_n(x)$ represents the local phase of the Fourier component at position x . The value of $\bar{\phi}(x)$ that maximizes this equation is the amplitude weighted mean local phase angle of all the Fourier terms at the point being considered. Taking the cosine of the difference between the actual phase angle of a frequency component and this weighted mean, $\bar{\phi}(x)$, generates a quantity approximately equal to one minus half this difference squared (the Taylor expansion of $\cos(x) \approx 1 - x^2/2$ for small x). Thus, finding where phase congruency is a maximum is approximately equivalent to finding where the weighted variance of local phase angles, relative to the weighted average local phase, is a minimum.

Under this definition, if all the Fourier components are in phase all the complex vectors would be aligned and the ratio of $|E(x)|/\sum_n A_n(x)$ would be 1. If there is no coherence of phase the ratio falls to a minimum of 0. Phase congruency provides a measure that is independent of the overall magnitude of the signal making it invariant to variations in image illumination and/or contrast.

As it stands, phase congruency is a rather awkward quantity to calculate. As an alternative, Venkatesh and Owens [6] show that points of maximum phase congruency can be calculated equivalently by searching for peaks in the local energy function. The local energy function is defined for a one-dimensional luminance profile, $I(x)$, as

$$E(x) = \sqrt{F^2(x) + H^2(x)} \quad (3)$$

where $F(x)$ is the signal $I(x)$ with its DC component removed, and $H(x)$ is the Hilbert transform of $F(x)$ (a 90 deg. phase shift of $F(x)$). Venkatesh and Owens show that energy is equal to phase congruency scaled by the sum of the Fourier amplitudes; that is,

$$E(x) = PC(x) \sum_n A_n \quad (4)$$

Thus, the local energy function is directly proportional to the phase congruency function, so peaks in local energy will correspond to peaks in phase congruency.

The Radon transform can be used to detect linear trends in images. The Radon transform, along a direction which there are more straight lines, usually has larger variations. Therefore, the variance of the projection at this direction is locally maximum. Here, we define the image principle direction as the orientation of the image, which has the maximum of phase congruency of variance.

3 Experimental Results

The proposed method is tested by two databases. Data set 1 consists of 25 texture images of size 512×512 from Brodatz album [7], as shown in Fig.1. We divide each texture into four 256×256 nonoverlapping regions, and each 256×256 region was rotated at angle 0 degrees to 180 degrees with 5 degrees increments and, from each rotated image, one 128×128 subimage was selected. Therefore, there are 3600 ($25 \times 4 \times 36$) testing samples.

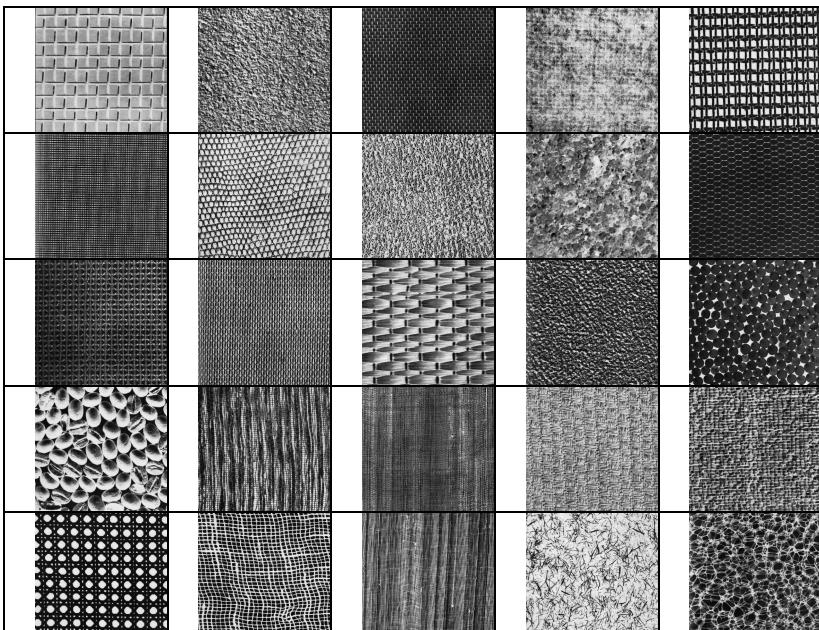


Fig. 1. Twenty-five classes of textures from the Brodatz album. Row1: D1, D4, D6, D19, D20. Row 2: D21, D22, D24, D28, D34. Row 3: D52, D53, D56, D57, D66. Row 4: D74, D76, D78, D82, D84. Row 5: D102, D103, D105, D110, D111.

As a criterion of evaluating performance, we used the error mean and square root of mean square(SRMS) error and compared the performance of the proposed method with the second derivative[8]. Table1 summarizes the error mean and square

root of mean square (SRMS) error. The error is defined as the difference between the estimated orientations of each two successive rotated textures (which are supposed to be 5 degrees) minus 5.

Table 1. The performance of the proposed method and the second derivative

Second Derivative[8]		Proposed	
Error Mean	SRMS	Error Mean	SRMS
5.56	12.61	1.64	5.71

As shown in table1, it is known that the error mean of the proposed method is within $\pm 1.7^\circ$ and smaller than the second derivative, $\pm 5.6^\circ$.

For Data set 2 with joint rotation and scale changes, it consists of 10 images of size 128×128 , containing military tank targets, and is shown in the fig2. Each was rotated at angle 0 degrees to 180 degrees with 5 degrees increments, and then was scaled using four parameters (0.25, 0.5, 2, 4). Table2 summarizes the error mean and square root of mean square error.



Fig. 2. Ten image of tanks for Data set 2

Table 2. The Error Mean and SRMS for data set 2

tank	Error Mean	SRMS
1	0.33	0.67
2	0.11	0.33
3	0.11	0.33
4	0.77	1.33
5	3.89	7.82
6	0.44	0.75
7	0.78	1.33
8	3.33	6.54
9	0.44	1.05
10	0.22	0.47

The results in table2 indicate that worst are tank 5 and 8, error mean 3.89 and 3.33, respectively; other error mean are smaller than $\pm 1^\circ$. So, it indicates that this approach is fairly robust.

4 Conclusions

A new simple approach for orientation estimation is proposed based on the phase congruency in radon domain. The performance of this technique is determined by conducting simulation experiments on two sets of images, containing military targets and textures, respective. The experiments confirm that the method can be used successfully in determining the orientation of the texture and military targets to a reasonable degree of accuracy. Therefore, the location of the target in the scene, target size, small variations in the target aspect angle, and certain amount of occlusion of the target does not cause significant problems in the determination of orientation.

References

- Chandra, D.V.S.: Target Orientation Estimation Using Fourier Energy Spectrum. *IEEE Trans. Aerospace Electronic Systems* 34(3), 1009–1012 (1998)
- Mester, R.: Orientation Estimation: Conventional Techniques and a New Non-Differential Approach. In: Proc. 10th European Signal Processing Conf., vol. 2, pp. 921–924 (2000)
- Bigun, J., Granlund, G.H., Wiklund, J.: Multidimensional Orientation Estimation with Applications to Texture Analysis and Optical Flow. *IEEE Trans. Pattern Analysis and Machine Intelligence* 13(8), 775–790 (1991)
- Bracewell, R.N.: Two-Dimensional Imaging. Prentice Hall, Englewood Cliffs (1995)
- Morrone, M.C., Owens, R.A.: Feature detection from local energy. *Pattern Recognition Letters* 6, 303–313 (1987)
- Venkatesh, S., Owens, R.A.: An energy feature detection scheme. In: The International Conference on Image Processing, Singapore, pp. 553–557 (1989)
- Brodatz, P.: Texture: A Photographic Album for Artists and Designers. Dover, New York (1966)
- Jafari-Khouzani, K., Soltanian-Zadeh, H.: Radon Transform Orientation Estimation for Rotation Invariant Texture Analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence* 27(6), 1004–1008 (2005)

Query Classification and Expansion for Translation Mining Via Search Engines

Jian-Min Yao, Jun Sun, Lei Guo, and Qiao-Ming Zhu

Provincial Key Laboratory of Computer Information Processing Technology
Soochow University, Suzhou, China, 215006

{jyao, qmzhu}@suda.edu.cn, guolei0304@126.com, sundaey@163.com

Abstract. Out-of-vocabulary lexicons, including new words, collocations, as well as phrases, are the key flesh of a human language while an obstacle to machine translation. But the translation of OOV is quite difficult to obtain. A web-mining solution to the OOV translation is adopted in our research. The basic assumption lies in that most of the OOV's translations exist on the web, and search engines can provide many web pages containing the OOV and corresponding translations. We mine the translation from returned snippets of the search engine with expanded OOV as the query term. The difference of our method from other methods lies in that a query classification is made before submitting to the search engine. Experiment shows our solution can discover the translation to many of the OOVs with quite high precision.

Keywords: web mining, OOV, transliteration, free translation, literal translation, query classification, query expansion.

1 Introduction

The web is becoming more and more important in knowledge acquisition, as is the same for searching for the translation of unknown words or phrases. The web has abundant resources for translation knowledge. The first kind of web resources is parallel data on the web [1]. Gale and Church, Kupiec, Melamed, Smadja et al have used sentence-aligned parallel corpora to extract translations [2-5]. Another kind of web resources is the comparable corpus on the web by Fung [6]. This task is more difficult due to lack of parallel correlation between document or sentence pairs.

Lu et al. extracted translation pairs from anchor texts pointing to the same web page [7]. Anchor text sets, which are composed of a number of anchor texts linking to the same pages, may contain similar description texts in multiple languages, thus it is more likely that words and their corresponding translations frequently appear together in the same anchor text sets.

A more common web resource for term translations is the web page of mixed languages. Web-based approach to exploring abundant language-mixed texts on the Web like anchor texts and search-result pages for alleviating the difficulty of unknown query term translation is a hot topic [8-10].

Research on digital libraries has proposed similar approaches. Larson, Gey and Chen introduced a method for translingual vocabulary mapping using multilingual subject headings of book titles in online library catalogs, which is similar to the parallel corpora [11].

This paper introduces our work on web-based translation mining. The query classification technique is first introduced in the paper and shows good performance in various domains.

2 Query Classification Techniques

The query can be any Chinese phrases, idioms, words, or even segments of sentences. To improve the web mining performance, we classify the query terms into literal translation query and free translation query, so that target-oriented mining algorithms can be designed for different kinds of queries.

For literal translation, we refer to the kind of query that the translation of the whole query is a simple combination of the corresponding components. For example, the query “机器翻译” is translated as “machine translation”, which is a literal translation because it’s a direct combination of the translation of “机器” (machine) and “翻译” (translation).

For free translation, we refer to those queries that their translations are not simple combinations of the translations of their components, that is, the meaning of the whole query is different from the literal meaning of the components. For example, the translation of “炒鱿鱼” (get fired) is not related to “鱿鱼” (which means squid).

Our solution to query classification into free translation and literal translation are composed of three modules: 1) to identify whether the query is a transliteration; 2) Free translation identification; and 3) Literal translation identification. The modules are described in the following sub-sections.

2.1 Transliteration Identification

Transliteration is the practice of transcribing a word or text written in one writing system into another writing system or system of rules for such practice. From a linguistic point of view, transliteration is a mapping from one system of writing into another, word by word, such as “克林顿” for Clinton, “列支敦士登” for Liechtenstein etc. We put transliteration as a special kind of free translation. The process we designed to identify transliterations is based on two Chinese-character-based models. The first model is shown in the equation 1.

$$P_t(w) = \frac{C_t(w)}{C(w)} \quad (1)$$

This is a ratio of transliteration characters over ordinary characters. The transliteration characters refer to those Chinese characters that are in the transliteration character list published by the Xinhua New Agency.

The second language model is as shown in equation 2.

$$P_2(w) = \frac{\sum p(w_i)}{C(w)} \quad (2)$$

In which, $p_2(w)$ is an average probability of all the characters in the word w to be in the transliteration character list. $p(w_i)$ refers to the probability of i th character into serve as a transliteration character. $C(w)$ is the total number of characters in the query w .

To take advantage of the two language models, we also have designed some heuristic rules, which complement the statistical methods. Two main rules are as follows:

- 1) If the query contains 1~2 characters, use the first language model;
- 2) If the query contains more than 2 characters, use the first model; if true, return YES; Else use the second model, if true, return YES; Else return NO.

2.2 Free Translation Identification

If the query is not a transliteration, whether a query is to be literally translated or freely translated can be judged according to the following two aspects of knowledge: 1) To judge from the pragmatic environment of the query; 2) To judge from the linguistic features of the query.

The pragmatic environments of the query can be decided according to the following clues: 1) look up in the returned snippets of a search engines; 2) Definition of the query in the dictionaries.

According to linguistic assumption, if the query is a free translation, then the literal translation of the query cannot co-occur frequently with the word, described as in the equation below,

$$\theta(w) = \frac{\sum_i count(w, e_i) - count(w, e_1, e_2, \dots, e_n)}{count(w)} \quad (3)$$

In which, w is the query to be classified, e_i are the two or more component words of the query; $count(\bullet)$ is the number of the argument in the return snippets of a search engine. The threshold of the classification is decided experimentally.

A threshold is experimentally decided to identify whether the query is a literal translation or a free translation, which will help in the translation mining step.

3 Query Expansion and Translation Mining Algorithm

Query expansion before submitting to the search engine (in our case, it's www.baidu.com, which is the largest search engine for Chinese web pages) is the first key step in our algorithm. The query involves two parts: 1) the Chinese query (word, phrase or segment to be translated.) and 2) translation of its longest-possible subsequence. When a query mixing Chinese and English is submitted to the search

engine, many of the returned snippets contain the query, so some may contain the whole translation of the Chinese query. Many of the returned snippets contain the translation of the Chinese query, but most of them are buried among many other English words. Many returned snippets for a query of mixed languages contain the translation to the Chinese query.

If a Chinese query (a word, a phrase, or a segment) is fed into the system, a lookup of the local dictionary is first carried out to see whether the query is already in the vocabulary. If the query is in the vocabulary, the translation is returned; else the query is expanded by the following algorithm [12], which resembles the longest match method for Chinese word segmentation.

```
ALGORITHM: Query expansion
INPUT: the Chinese query C_Query
OUTPUT: Expanded query composed of the Chinese query
plus English query Exp_Query
Sub_Seq = C_Query
LOOP UNTIL Sub_Seq is NULL
{
    Sub_Seq = C_Query - first character of the C_Query
    IF (Sub_Seq is in the dictionary)
        Exp_Query = C_Query + translation of the Sub_Seq
    RETURN Exp_Query
    ENDIF
}
}
```

Co-occurrence frequency is utilized for the translation mining from the returned snippets. The process is to submit the Chinese query plus each of the translation of the longest sub-sequence to the search engine, and get all the returned snippets together. After filtering out the stop words such as function words or some impossible strings, count the frequency of all the English strings, and just return the top N most frequent strings as the translation candidates. For an algorithmic description, see the algorithm below.

```
ALGORITHM: Translation mining
INPUT: the expanded query Exp_Query
OUTPUT: Top N most probable translation candidates for
the Chinese query C_Query
BEGIN PROCEDURE
FOR EACH element Eng_Trans in the translations of the
longest subsequence
{
    Return_Snip = Web_Search(C_Query + Eng_Trans)
    Del_Stop_Words_from(Return_Snip)
    Tran_Candidates += all strings in Return_Snip
}
Sort_by_Freq(Trans_Candidates)
Return(Top_K Trans_Candidates)
END PROCEDURE
```

4 Experiment and Analysis

In the experiment, we just take the top 10 returned snippets, which will lead to higher time and space efficiency while not much decrease in precision, because most reliable information will be ranked higher. 6 Evaluation of the translation mining system

The translation mining system is taken as a huge dictionary for translation. So we randomly choose some phrases and technical terms from technical books to see the performance of the system. We take the test set from the China Translation Seminar web site (<http://www.chinatranslation.org>), which includes the first 200 automobile terms and 200 computer terms. In addition, we take the first 20 key university names from the China Education Ministry web site to test the performance on named entities. The translation mining result is given in the table 1.

Table 1. Proportion of the TOP N mining results containing the correct translation in various domains

Domain	TOP N	Proportion
Automobile	1	80%
	2	80%
	3	86%
	1	50%
Computer	2	50%
	3	60%
	1	75%
University names	2	90%
	3	95%

In our daily works, many people are using the search engines to assist in translation, e.g. to search for the translation candidates, to confirm the correctness of a possible translation, etc. From this point of view, the Internet resources are playing a key role in translation assistance in our daily lives. This paper is an effort towards automating the translation mining process, which shows satisfactory performance in translation mining accuracy. The work needs to be improved in the following aspects: 1) The literalness of the query will be calculated in more accuracy based on better algorithm; 2) Alignment and other methods will be tried to improve the translation mining process from the returned snippets.

Acknowledgements. The research project is supported by the Natural Science Foundation of Jiangsu Province (Contract No. BK2006539), the Natural Science Foundation for Higher Education in Jiangsu Province (Contract No. 06KJB520095).

References

1. Resnik, P., Smith, N.A.: The Web as a Parallel Corpus. Computational Linguistics 29(3), 349–380 (2003)
2. Gale, W.A., Church, K.W.: Identifying Word Correspondances in Parallel Texts. In: Proc. Of DARPA Speech and Natural Language Workshop, pp. 152–157 (1991)

3. Kupiec, J.M.: An Algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora. In: Proc. of ACL, pp. 17–22 (1993)
4. Melamed, I.D.: Models of Translational Equivalence Among Words. Computational Linguistics 26(2), 221–249 (2000)
5. Smadja, F., McKeown, K., Hatzivassiloglou, V.: Translating Collocations for Bilingual Lexicons: A Statistical Approach. Computational Linguistics 22(1), 1–38 (1996)
6. Fung, P., Yee, L.-Y.: An IR Approach for Translating New Words From Nonparallel, Comparable Texts. In: Proc. of ACL, pp. 414–420 (1998)
7. Lu, W.H., Chien, L.F., Lee, H.J.: Translation of Web Queries Using Anchor Text Mining. ACM Trans. Asian Language Information Processing (TALIP) 1(2), 159–172 (2002)
8. Shia, M.-S.: Improving Translation of Unknown Proper Names Using a Hybrid Web-based Translation Extraction Method. In: Proceedings of ROCLING (2005)
9. Cheng, P.J., Pan, Y.C., Lu, W.H., Chien, L.F.: Creating Multilingual Translation Lexicons with Regional Variations Using Web Corpora. In: Proc. of ACL, pp. 535–542 (2004)
10. Huang, F., Zhang, Y., Vogel, S.: Mining Key Phrase Translations from Web Corpora. In: The Proceedings of the Human Language Technologies Conference (HLT-EMNLP), pp. 483–490 (2005)
11. Larson, R.R., Gey, F., Chen, A.: Harvesting Translingual Vocabulary Mappings for Multilingual Digital Libraries. In: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries, pp. 185–190 (2002)
12. Sun, J., Yao, J.M., Zhang, J., Zhu, Q.M.: Web Mining of OOV Translations. Journal of Information & Computational Science 5(5), 2057–2064 (2008)

Author Index

- Ahmed, Chowdhury Farhan 933
Ahmed, K. 593
Analyti, Anastasia 5
Anh, Duong Tuan 698
Anthony, Patricia 223
Antoniou, Grigoris 5
Arkoudas, Konstantine 17
- Bagnoli, Franco 829
Bai, Xiang 30
Baral, Chitta 345
Baxter, Rohan 556
Beal, Carole R. 66
Binh, Huynh Thi Thanh 1077
Blockeel, Hendrik 2
Borujerdi, Mohammadreza Matash 646
Bringsjord, Selmer 17
Britz, Katarina 370
Bui, Hung H. 903
- Cao, Longbing 849, 1010
Cao, Peng 778
Cao, Tru H. 603
Castellanos-Nieves, Dagoberto 42
Chang, Liang 778
Chen, Changyou 533
Chen, Jyun-Sing 971
Cheng, Wen 1104
Cheng, Yuan 533
Cheung, Yiu-ming 913
Cios, Krzysztof J. 788
Clark, John A. 1053
Cleaver, Timothy William 54
Cohen, Paul R. 1, 66
Cruz, Carlos 42
Cuzzolin, Fabio 78, 91
- Damásio, Carlos Viegas 5
Dinh, Dien 809
Doan, An-Hai 3
Doukari, Omar 939
Dung, Le Xuan 945
Duong, Thanh D.X. 103
Duong, Vu N. 103
- Egawa, Seiji 115
Endo, Satoshi 614
- Fan, Fan 998
Feng, Xiang-hu 880
Fernandez-Breis, Jesualdo Tomas 42
Foitong, Sombut 1028
Fortuna, Blaž 626
Fu, Yuxiao 998
Fujii, Kunikazu 923
Fujita, Hiroshi 839
Fürnkranz, Johannes 636
- Gamberger, Dragan 636
Gardiner, Katheleen J. 788
George, Sarah 581
Gholami, Ehsanollah 646
Ghose, Aditya 991
Golińska-Pilarek, Joanna 128
Guo, Lei 1121
- Harnsamut, Nattapon 273
Hasegawa, Ryuzo 839
Hashimoto, Kazuo 799
Hashimoto, Takashi 152
Hashizume, Hideki 658
Hassan, Mohd Fadzil 668
Havens, William S. 768
Hayama, Tessai 678
He, Yanxiang 740
Hirashima, Tsukasa 951
Ho, Chong Mun 223
Hoang, Linh 688
Holmes, Geoffrey 485
Hoque, M.A. 593
Horiguchi, Tomoya 951
Hörne, Tertia 370
Hu, Derek Hao 1110
Hu, Yi 175
Huang, Xiaojiang 187
Hui, Siu Cheung 309
Hung, Nguyen Quoc Viet 698
Hušek, Petr 708
- Igarashi, Harukazu 164
Ishihara, Seiji 164

- Islam, K.K. 593
 Itoh, Chika 959
 Itoh, Hidenori 658, 959, 1016
 Iwata, Tomoharu 965
 Jauregui, Victor 718
 Jeansoulin, Robert 939
 Jeong, Byeong-Soo 933
 Jia, Yunde 466
 Jin, Zhi 544
 Kambara, Yoshiaki 839
 Kanazaki, Hirofumi 442
 Kanoh, Masayoshi 1016
 Katayama, Susumu 199
 Kato, Shohei 658, 959, 1016
 Kato, Yoshihide 115
 Khalid, Marzuki 568
 Kim, Jong Nam 1116
 Kim, Kye-Sung 211
 Kimura, Masahiro 977
 Kojiri, Tomoko 965
 Komatani, Kazunori 890
 Kordic, Savo 985
 Koshimura, Miyuki 839
 Koukolíková-Nicola, Zdena 829
 Kozaki, Kouji 614
 Kunifugi, Susumu 678
 Lam, Peng 985
 Lan, Zhen-zhong 880
 Latecki, Longin Jan 30
 Lavrač, Nada 626, 636
 Le, Hoai-Bac 819
 Le, Huong Thanh 728
 Le, Khanh C. 603
 Le Nhat, Minh 1086
 Lee, Jung-Tae 688
 Lee, Sang-Jo 211
 Lee, Young-Koo 933
 Leenen, Louise 991
 Li, Gang 333
 Li, Huaizhong 985
 Li, Lei 1110
 Li, Wenjie 175, 740
 Li, Xiang 1092
 Li, Xue 140
 Liang, Jung-Chin 971
 Lim, Deborah 223
 Lin, Weiqiang 556
 Lió, Pietro 829
 Liu, Fang 998
 Liu, Huawen 235
 Liu, Lei 235
 Liu, Li 1010
 Liu, Rey-Long 1004
 Liu, Wenyu 30
 Lu, Qin 175, 740
 Lu, Songfeng 998
 Luo, Chao 849, 1010
 Ma, Feifei 247
 Machida, Kazuo 442
 Makalic, Enes 581, 750
 Martinez-Bejar, Rodrigo 42
 Matsubara, Shigeki 115
 Matsuhsa, Takashi 760
 Matsui, Yuki 1016
 Matsumoto, Kazunori 799
 Matsumoto, Yuji 4
 Mikura, Yuki 1098
 Mine, Tsunenori 839
 Mirajkar, Pranav Prabhakar 1065
 Missine, Andrei 768
 Miyabe, Mai 1022
 Mizoguchi, Riichiro 614
 Mohd Zin, Zalhan 568
 Mora, Angel 128
 Motoda, Hiroshi 977
 Munir, M.S. 593
 Muñoz-Velasco, Emilio 128
 Murayama, Norifumi 260
 Mutoh, Atsuko 658
 Nanba, Hidetsugu 678
 Nasu, Yo 1098
 Natwichai, Juggapong 140, 273
 Ngo, Vuong M. 603
 Nguyen, Anh Kim 728
 Nguyen, Cao D. 788
 Nguyen, DucDung 799
 Nguyen, Duong 788
 Nguyen, Ha 284
 Nguyen, Phi-Vu 819
 Nguyen, Thach Huy 1028
 Nguyen, Truong Binh 1077
 Nguyen, Viet-Anh 829
 Nguyen Duc Hoang, Ha 1086
 Nguyen Thi, Hong-Nhung 809
 Nguyen Truong Duc, Tri 1086

- Niemann, Michael 581, 750
 Niu, Wenjia 778
 Noda, Itsuki 296
 Ogata, Tetsuya 890
 Ohta, Masayuki 296
 Oishi, Tetsuya 839
 Okumura, Manabu 260
 Okuno, Hiroshi G. 890
 Orgun, Mehmet A. 556
 Ou, Yuming 849, 1010
 Park, Se-Young 211
 Park, Seong-Bae 211
 Park, Sun 1034
 Peng, Hui 778
 Pfahringer, Bernhard 485
 Pham, Duc Nghia 405
 Pham, Son Bao 718
 Phung, Dinh Q. 430, 903
 Phuong, Nguyen Duy 859
 Phuong, Tu Minh 859
 Pinngern, Ouen 1028, 1071
 Pontelli, Enrico 358
 Prendes-Espinosa, Maria Paz 42
 Quan, Tho Thanh 309
 Ren, Fenghui 321
 Ren, Yongli 333
 Richards, Debbie 1039
 Rim, Hae-Chang 688
 Robertson, Dave 668
 Rojanavasu, Pornthep 1071
 Saito, Kazumi 977
 Sattar, Abdul 54
 Scherl, Richard 345
 Schmidt, Daniel 750
 Schönwälder, Jürgen 417
 Schwitter, Rolf 1046
 Şen, Sevil 1053
 Sheehan, Matthew 1059
 Shen, Liran 1116
 Shen, Yidong 382
 Shen, Zhiyong 382
 Shi, Yuan 870, 880
 Shi, Zhongzhi 778
 Shigenobu, Tomohiro 923, 1022
 Shihavuddin, A.S.M. 593
 Shiramatsu, Shun 890
 Song, Hyun-Je 211
 Song, Young-In 688
 Srinil, Phaitoon 1028
 Stevenson, Lynn 370
 Sun, Jun 382, 1121
 Sun, Xingzhi 140
 Sundar, Anoj Ramasamy 394
 Sureka, Ashish 1065
 Takahashi, Kenichi 1098
 Takatori, Daichi 799
 Takishima, Yasuhiro 799
 Tamee, Kreangsak 1071
 Tan, Colin Keng-Yan 394
 Tanbeer, Syed Khairuzzaman 933
 Terabe, Masahiro 799
 Thang, Le Quang 859
 Thornton, John 405
 Tran, Cao Son 345, 358
 Tran, Ha Manh 417
 Truyen, Tran The 430, 903
 Udomthanapong, Sonchai 1071
 Ueda, Hiroaki 1098
 Ueta, Atsushi 442
 Valencia-Garcia, Rafael 42
 Vanschoren, Joaquin 485
 Velardi, Paola 626
 Venkatesh, Svetha 430, 903
 Vo, Quoc Bao 497
 Vu Hai, Quan 1086
 Vuong, Nhu Khue 1092
 Wagner, Gerd 5
 Wan, Xiaojun 187, 454
 Wang, Kun-Chieh 971
 Wang, Tianjiang 998
 Wang, Yuanquan 466
 Warashina, Katsuhide 152
 Watanabe, Toyohide 965
 Watson, Ian 1059
 Wei, Furu 740
 Weruaga, Luis 284
 Williams, Mary-Anne 473
 Würbel, Eric 939
 Xiao, Jianguo 187
 Xiao, Jitian 985
 Xu, Yanbo 30

- Xue, Gui-Rong 509
Xue, Jiangwei 382
Yairi, Takehisa 442
Yamada, Takeshi 965
Yang, Jianwu 187
Yang, Qiang 1110
Yang, Xingwei 30
Yao, Jian-Min 1121
Ye, Yangdong 333
Yin, Qingbo 1116
Yoshino, Takashi 923, 1022
You, Ouyang 740
Yu, Yong 509, 521
Yusof, Rubiyah 568
Zeng, Hong 913
Zhang, Chengqi 849
Zhang, Congle 509, 521
Zhang, Huijie 235
Zhang, Jian 247
Zhang, Junping 533
Zhang, Minjie 321
Zhang, Shichao 544
Zhang, Xiaolong 1104
Zhang, Yihao 556
Zhao, Wenbo 382
Zhao, Yanchang 1010
Zhao, Yi Zhi 1092
Zhong, Xing 870
Zhu, Qiao-Ming 1121
Zhu, Shaohua 998
Zhu, Xiaofeng 544
Zhuo, Hankui 1110
Zukerman, Ingrid 581, 750